GI-Edition

Lecture Notes in Informatics

Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.)

Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband

11.–12. März 2013
Magdeburg

Proceedings

216

G. Saake et al. (Hrsg.): BTW 2013 – Workshopband

Gunter Saake, Andreas Henrich, Wolfgang Lehner,
Thomas Neumann, Veit Köppen (Hrsg.)

Datenbanksysteme für Business, Technologie und Web (BTW) 2013 - Workshopband

11. – 12.03.2013
in Magdeburg, Germany

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-216

ISBN 978-3-88579-610-7
ISSN 1617-5468

Volume Editors

Gunter Saake
Otto-von-Guericke-Universität Magdeburg
Institut für Technische und Betriebliche Informationssysteme
Universitätsplatz 2
39106 Magdeburg, Germany
E-Mail: Gunter.Saake@ovgu.de

Andreas Henrich
Otto-Friedrich-Universität Bamberg
Fakultät für Wirtschaftsinformatik und Angewandte Informatik
Lehrstuhl für Medieninformatik
96047 Bamberg, Germany
E-Mail: andreas.henrich@uni-bamberg.de

Wolfgang Lehner
Technische Universität Dresden
Fakultät Informatik
Institut für Systemarchitektur
01062 Dresden, Germany
Email: wolfgang.lehner@tu-dresden.de

Thomas Neumann
Technische Universität München
Institut für Informatik - Lehrstuhl III (I3)
Boltzmannstr. 3
85748 Garching bei München, Germany
E-Mail: thomas.neumann@in.tum.de

Veit Köppen
Otto-von-Guericke-Universität Magdeburg
Institut für Technische und Betriebliche Informationssysteme
Universitätsplatz 2
39106 Magdeburg, Germany
E-Mail: Veit.Koeppen@ovgu.de

printed by Köllen Druck+Verlag GmbH, Bonn

Vorwort

In den letzten Jahren hat es auf dem Gebiet des Datenmanagements große Veränderungen gegeben. Dabei muss sich die Datenbankforschungsgemeinschaft insbesondere den Herausforderungen von „Big Data“ stellen, welches die Analyse von riesigen Datenmengen unterschiedlicher Struktur mit kurzen Antwortzeiten erfordert. Neben klassisch strukturierten Daten müssen moderne Datenbanksysteme und Anwendungen ebenfalls semistrukturierte, textuelle und andere multi-modale Daten sowie Datenströme in völlig neuen Größenordnungen verwalten. Gleichzeitig müssen die Verarbeitungssysteme die Korrektheit und Konsistenz der Daten sicherstellen.

Die jüngsten Fortschritte bei Hardware und Rechnerarchitektur ermöglichen neuartige Datenmanagementtechniken, die von neuen Index- und Anfrageverarbeitungsparadigmen (In-Memory, SIMD, Multicore) bis zu neuartigen Speichertechniken (Flash, Remote Memory) reichen. Diese Entwicklungen spiegeln sich in aktuell relevanten Themen wie Informationsextraktion, Informationsintegration, Data Analytics, Web Data Management, Service-Oriented Architectures, Cloud Computing oder Virtualisierung wider.

Wie auf jeder BTW-Konferenz, gruppieren sich um die Tagung eine Reihe von Workshops, die spezielle Themen in kleinen Gruppen aufgreifen und diskutieren. Im Rahmen der BTW2013 finden insgesamt folgende sechs Workshops statt:

- Information Systems in Digital Engineering (ISDE): Fokus des Workshops ist die Schnittstelle von Datenbank/-Informationssystemen mit der Realisierung physischer mobiler und eingebetteter Geräte. Spezielle Herausforderungen für massiv-verteilte und eingebettete Systeme werden genauso diskutiert wie Aspekte von Safety, Security und Privacy.
- Data Management in the Cloud (DMC): Der DMC-Workshop spiegelt gleichzeitig die Gründungsveranstaltung der gleichnamigen Arbeitsgruppe im GI Fachbereich DBIS wider. Gegenstand des Workshops sind spezielle Themen, wie sich Cloud-Technologien auf die Datenverarbeitung auswirken bzw. welche neuen Applikationen durch cloud-basiertes Data Management ermöglicht werden.
- Data Streams and Event Processing (DSEP): Den speziellen Anforderungen aus dem Bereich der Event- und Datenstromverarbeitung widmet sich der Workshop DSEP. Wie bereits in den vergangenen Jahren, werden vielfältige Themenbereiche - angefangen von grundlegenden Modellierungsfragen bis hin zu speziellen Techniken der Datenstromverarbeitung - adressiert.

- Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS): Die umfassende Unterstützung von Datensicherheits- und Datenschutzanforderungen adressiert dieser Workshop. Ziel ist es, in kleiner Runde speziell Themen der Biometrie, der Verifikation der Unversehrtheit digitaler Dokumente etc. zu diskutieren.
- Mapping Research in Mobility and Mobile Information Systems (MMSmap): Dieser Arbeitskreis wird durch die Fachgruppe „Mobilität und Mobile Informationssystem“ im Fachbereich DEBIS geprägt; entsprechend werden Themen rund um mobiles Datenmanagement, speziell mit Blick auf die aktuellen Entwicklungen im Bereich mobiler Endgeräte, im Mittelpunkt stehen.
- Crowd-enabled Data and Information Management (CDIM): Crowd-Technologie wird mehr und mehr dazu eingesetzt, „schwierige“ Integrations- und Analyseaufgaben als Teil einer Data-Management-Lösung zu realisieren. Die Herausforderungen aber auch Möglichkeiten versucht dieser Workshop im Detail zu eruieren.

Zusammenfassend bleibt festzustellen, dass das Workshop-Angebot extrem reichhaltig und breit gefächert ist. Für jeden Geschmack sollte sich somit ein Betätigungsfeld finden lassen!

Die Materialien zur BTW 2013 werden auch über die Tagung hinaus unter http://www.btw-2013.de zur Verfügung stehen.

Die Organisation einer so großen Tagung wie der BTW mit ihren angeschlossenen Veranstaltungen ist nicht ohne zahlreiche Partner und Unterstützer möglich. Sie sind auf den folgenden Seiten aufgeführt. Ihnen gilt unser besonderer Dank ebenso wie den Sponsoren der Tagung und der GI-Geschäftsstelle.

Magdeburg, Bamberg, Dresden, München, im Februar 2013

Gunter Saake, Tagungsleitung und Vorsitzender des Organisationskomitees
Andreas Henrich und Wolfgang Lehner, Leitung Workshopkomitee
Thomas Neumann, Leitung Studierendenprogramm
Veit Köppen, Tagungsband und Organisationskomitee

Tagungsleitung

Gunter Saake, Otto-von-Guericke-Universität Magdeburg

Organisationskomitee

Gunter Saake
Veit Köppen
Stefan Barthel
Sebastian Breß
Alexander Grebhahn
Thomas Leich
Siba Mohammad
Viktor Sayenko
Ivonne Schröter
Anja Strube
Eike Schallehn
David Broneske
Ziqiang Diao
Katja Gündel
Andreas Lübcke
Maik Mory
Matin Schäler

Studierendenprogramm

Thomas Neumann, TU München

Koordination Workshops

Andreas Henrich, Universität Bamberg
Wolfgang Lehner, TU Dresden

Workshopkomitees

Information Systems in Digital Engineering (ISDE)

Vorsitz:Maik Mory, OVGU Magdeburg; Veit Köppen, OVGU Magdeburg; Gunter Saake, OVGU Magdeburg

Abdel-Badeeh M. Salem, Egypt
Matthias Güdemann, France
Syed Saif ur Rahman, Pakistan
Andreas Brenke, Germany
Norbert Elkmann, Germany
Sandor Vajna, Germany
Frank Ortmeier, Germany

Crowd-enabled Data and Information Management (CDIM)

Vorsitz Andreas Nürnberger, OVGU Magdeburg, Wolf-Tilo Balke, TU Braunschweig

Andreas Hotho, Universität Würzburg

Data Management in the Cloud (DMC)

Vorsitz: Andreas Thor Universität Passau; Kostas Tzoumas Technische Universität Berlin; Florian Stegmaier Universität Passau

Werner Bailer, Joanneum Research
Joos-Hendrik Boese, SAP
Rainer Gemulla, MPI für Informatik
Michael Granitzer, Universität Passau
Lars Kolb, Universität Leipzig
Ralf Klamma, RWTH Aachen
Harald Kosch, Universität Passau
Jorge Quiane, Saarland University
Tilmann Rabl, University of Toronto
Kai-Uwe Sattler, TU Ilmenau
Sebastian Schaffert, Salzburg Research
Holger Schwarz, University of Stuttgart
Sebastian Maneth, University of New South Wales
Stefanie Scherzinger, Regensburg University of Applied Sciences

Data Streams and Event Processing (DSEP)

Vorsitz: Peter Fischer, Universität Freiburg; Marco Grawunder Universität Oldenburg; Daniela Nicklas, Universität Oldenburg; Bernhard Seeger, Philipps-Universität Marburg

Andreas Behrend, Universität Bonn
Klemens Boehm, KIT
Ludger Fiege, Siemens Research
Dieter Gawlick, Oracle
Jörg Hähner, Universität Augsburg
Boris Koldehofe, Universität Stuttgart
Wolfgang Lehner, TU Dresden
Pedro Marrón, Universität Duisburg
Gero Mühl, Universität Rostock
Kai Sachs, SAP
Thorsten Schöler, FH Augsburg
Francois Bry, LMU München
Kai-Uwe Sattler, TU Illmenau
Klaus Meyer-Wegener, Universität Erlangen

Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS)

Vorsitz: Jana Dittmann, OVGU Magdeburg; Arno Fischer, FH-Brandenburg; Gunter Saake, OVGU Magdeburg; Claus Vielhauer, FH-Brandenburg

Rüdiger Grimm, University of Koblenz
Dominic Heutelbeck FTK,
Claus-Peter Klas, FTK
Sviatoslav Voloshynovskiy, unige
Stefan Katzenbeisser, Technical University Darmstadt
Edgar R. Weippl, sba-research

Inhaltsverzeichnis

Data Streams and Event Processing (DSEP)

Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS)

Workshop on Information Systems in Digital Engineering (ISDE)

Crowd-enabled Data and Information Management (CDIM)

BTW-Studierendenprogramm 2013

Data Streams and Event Processing (DSEP)

The processing of continuous data sources has become an important paradigm of modern data processing and management, covering many applications and domains such as monitoring and controlling networks or complex production system as well complex event processing in medicine, finance or compliance.

The goal of the workshop is to attract both academic and industrial contributions to foster the exchange of ideas and a discussion about the state of the art and future directions. In particular, the following topics are of interest to this workshop:

- Data streams
- Event processing
- Case Studies and Real-Life Usage
- Foundations
  - Semantics of Stream Models and Languages
  - Maintenance and Life Cycle
  - Metadata
  - Optimization
- Applications and Models
  - Statistical and Probabilistic Approaches
  - Quality of Service
  - Stream Mining
  - Provenance
- Platforms for event and stream processing, in particular
  - CEP Engines
  - DSMS
  - ”Conventional” DBMS
  - Main memory databases
  - Sensor Networks
- Scalability
  - Hardware acceleration (GPU, FPGA, ...)
  - Cloud Computing
- Standardisation

Program Chair

Peter M. Fischer (Universität Freiburg)
Marco Grawunder (Universität Oldenburg)
Daniela Nicklas (Universität Oldenburg)
Bernhard Seeger (Universität Marburg)

Program committee

Andreas Behrend (Universität Bonn)
Klemens Boehm (KIT)
Ludger Fiege (Siemens Research)
Dieter Gawlick (Oracle)
Jörg Hähner (Universität Augsburg)
Boris Koldehofe (Universität Stuttgart)
Wolfgang Lehner (TU Dresden)
Pedro Marrón (Universität Duisburg)
Klaus Meyer-Wegener (Universität Erlangen)
Gero Mühl (Universität Rostock)
Kai Sachs (SAP)
Thorsten Schöler (FH Augsburg)
Francois Bry (LMU München)
Kai-Uwe Sattler (TU Illmenau)

Stream Join Processing on Heterogeneous Processors

Tomas Karnagel, Benjamin Schlegel, Dirk Habich, and Wolfgang Lehner

University of Technology Dresden
Department of Computer Science
Database Technology Group
01062 Dresden
{tomas.karnagel; benjamin.schlegel; dirk.habich; wolfgang.lehner}@tu-dresden.de

Abstract:
The window-based stream join is an important operator in all data streaming systems. It has often high resource requirements so that many efficient sequential as well as parallel versions of it were proposed in the literature. The parallel stream join operators recently gain increasing interest because hardware is getting more and more parallel. Most of these operators, however, are only optimized for processors with homogeneous execution units (e.g., multi-core processors). Newly available processors with heterogeneous execution units cannot be exploited whereas such processors provide typically a very high peak performance. In this paper, we propose an initial variant of a window-based stream join operator that is optimized for processors with heterogeneous execution units. We provide an efficient load balancing approach to utilize all available execution units of a processor and further provide highly-optimized kernels that run on them. On our test machine with a 4-core CPU and an integrated graphics processor, our operator achieves a speedup of 69.2x compared to our single-threaded implementation.

1 Introduction

The window-based stream join is a fundamental operator in all data streaming systems. It joins tuples that fulfill certain predicates from two or more windows, which move continuously over input data streams. Depending on the size of the windows, the operator has high performance requirements so that many optimized versions have been proposed for it. This includes sequential stream join operators [VNB03, GO03], which exploit various index data structures (e.g., hash tables) for a faster processing as well as parallel stream join operators [GYB07, TM11], taking advantage of different types of parallelism. The parallel operators are gaining increasing interest because hardware nowadays is getting more and more parallel.

Besides increasing the number of parallel execution units (e.g., cores, vector instruction width) in modern multi-core processors, one interesting trend is the development of heterogeneous processors where circuits for different purposes are placed on one chip. On the one hand, each new processor generation provides new instruction sets like Streaming SIMD Extensions (SSE) or Advanced Vector Extensions (AVX) that provide specialized

instructions for video encoding, string processing, or data encryption. On the other hand, also different types of execution units are placed on a single chip. Examples for such processors with heterogeneous execution units are the IBM Cell processor, AMD processors of the Fusion brand, and Intel's Sandy Bridge or Ivy Bridge processors. The AMD and Intel processors, for example, host CPU and GPU cores on a single chip. Although heterogeneous processors typically provide a better peak performance than traditional multi-core processors, it is more demanding to develop algorithms that exploit this performance.

In this paper, we propose a window-based stream join operator that is tailor-made for processors with heterogeneous execution units. Our stream join processes tuples from windows of two input streams using a band predicate [DNS91]; the operator creates join tasks (i.e., batches of tuples from both streams) of which each is processed by one of the available execution units. A dynamic load balancer is used to evenly distribute the load. We implement the operator using OpenCL and provide three optimized OpenCL-kernels. On our test machine, which comprises CPU and GPU cores, we achieve high speedups of up to 70x compared to a single-threaded implementation and 20x compared to a multi-threaded implementation that both do not exploit the heterogeneous cores of the system.

2 Preliminaries

Our aim is to speed up window-based stream joins by partitioning the work and running the partitions on multiple devices within a heterogeneous environment. In the following, we present the window-based stream join and a way to run the join in a parallel fashion. We also give an introduction to OpenCL as a framework to program device code (called kernel) that is able to run on different hardware devices without further modifications.

2.1 Stream Joins with Band Conditions

The stream join is a important operation to concatenate and filter data streams. Stream joins are performed on two or more streams in which tuples are continuously arriving. Instead of holding all the data, the continuous stream is partitioned in stream windows, for which the join is executed. There exists many variants and optimizations for this operator. Aggarwal [Agg06] provides an excellent overview about these variants and respective sequential optimizations.

Recently, parallel stream join operators gained much attention. This includes stream join implementations for FPGAs and NUMA systems [TM11] as well as for the cell processor [GYB07]. In the latter, chunks of tuples are transferred to the 8 co-processors of the cell processor and these cores perform a nested-loop like operation. The authors chose the nested-loop join because it has no intermediate state that needs to be updated with every new tuple. It is also the best choice for implementing band conditions. A band join between relation R and S is a join, where the values of the join attribute on R need to fall in a predefined band of the join attribute values in S to be joined together. An example could

be found in position tracking. Two position sensors stream data and the join on a time window is done to find cases, where both sensors are close to each other. The definition of *close* is done by the band conditions, e.g., a radius. There are approaches to do a efficient sort-merge join [LT95] or hash join [Sol93] with band conditions but the nested-loop join is a straightforward approach, which is also easy to parallelize.

In this paper, we focus on a stream join with band condition as an important derivation of the stream join. Based on the stream join running on the cell processor, we port the algorithm to work efficiently on the GPU (Graphics Processing Unit) as well as on the CPU. Furthermore, we exploit both computing devices for parallel execution using OpenCL. Previous work has been done for database join algorithms on multi-core CPUs [SD89] as well as join algorithms on the GPUs [HYF+08, KLMV12]. In the latter, the join data is copied to and from the GPU device through the PCI express interface and the join execution was done completely on the GPU.

2.2 Open Computing Language (OpenCL)

We chose the *Open Computing Language* (OpenCL)[Gro] to implement our join for heterogeneous hardware environments. OpenCL is standardized by the Khronos Group and implemented by hardware vendors for their specific hardware. It allows to run the same code on different hardware platforms including x86-CPUs and most of the modern GPUs. With a pre-installed driver, OpenCL programs can use all supported computing devices available in a system. These could be multiple CPUs, GPUs, FPGAs, or different kinds of accelerators. To use the OpenCL framework, a programmer has to write host code and device code. The host code is written in C or C++ (or other languages using wrappers). It is used to initialize the different hardware devices, to manage data transfers to and from the devices, and to enqueue device code (kernel) executions. For the execution of a kernel, the host has to specify the arguments for the kernel and the dimensions the kernel is executed in. The arguments could be single values or pointer to arrays of data, which need to be transfered before kernel execution. The dimensions define the logical positioning of work-items, e.g., in a two dimensional array. Work-items can query their position (index in each dimension) during execution. It is possible to group work-items into work-groups. Synchronization between work-items is only possible within work-groups.

OpenCL devices are physically partitioned in compute units, which again are partitioned in processing elements. Each processing element can execute one or more work-items. Kernels are written in a subset of C, which is extended by mathematic, relational, and vector functions. All started work-items execute the same kernel. Such kernels read data, do computation, and store a result. While all work-items execute the same kernel, the difference between them is that they load data and put the results to different positions in memory. If used on a GPU, the kernels should be highly parallel.

In OpenCL, the main memory of the device is called *global memory*. For most compute devices, global memory is the largest memory but also the memory, which takes the most time to access. Some devices also have *local memory* (shared memory), which is faster

but smaller. Local memory can be shared between work-items in one work-group. When sharing memory, the work-group synchronization features are needed to avoid inconsistent data. There is also *private memory*, which can only be accessed by a corresponding work-item. When supported in hardware, private memory is implemented with fast registers.

3 Parallel Stream Join Execution

For a fast stream join implementation on multiple devices, efficient load balancing is needed as well as optimized device friendly code. In the following, we present our approach to achieve dynamical load balancing and we further discuss the implementation details of our OpenCL kernels as well as two CPU variants as our basis of comparison.

3.1 Dynamic Load balancing

In OpenCL it is possible to run kernels on specific devices. If the system has multiple devices of the same or a different kind, then kernel and load distribution is needed. One way would be to distribute individual kernels, which are adapted to the device properties together with dedicated data chunks for this kernel. Another way is a unified kernel that is deployed to all devices and load balancing is done through partitioning the data.

In this paper, we build an unified kernel that works with partitioned data. Such a kernel does not need information about the devices at compile time and all OpenCL devices are supported. Hence, our focus is on a good data partitioning and load balancing. For efficient load balancing, we apply a job queue approach. We implemented a task queue where each task holds the attributes of a number of join tuples from two stream windows. There is also a placeholder for the results of the join. The data for filling the tasks could come from hypothetical endless streams, which keep filling the queue with tasks. Join windows are taken from the streams to create a task for the queue. The windows of one stream could be overlapping or distinct. We choose the distinct variant for this work. This is equal to a batch-wise stream join operation. A batch of new values for one stream is joined with a join window of another stream and vice versa. Only the data stored in a task is joined together. The OpenCL devices pick the tasks from the front of the queue and mark it as 'in-work'. Only tasks not 'in-work' and not 'done' are picked. A task is executed on a device and the result is written back to the result placeholder. The task is then marked as 'done' and can be further processed (e.g., streamed out to the next node). The device then selects the next free task to execute.

The queuing setup is illustrated in Figure 1. With the job queue, it is possible to have a dynamic load balancing because every device executes the tasks as fast as possible and without being idle, selects a new task from the queue. This way it would be easy to add more devices to the environment even if they have different processing capabilities.

One single task execution involves copying the needed data to the device, execution of the kernel with the given data, and copying the data from the device back to the result

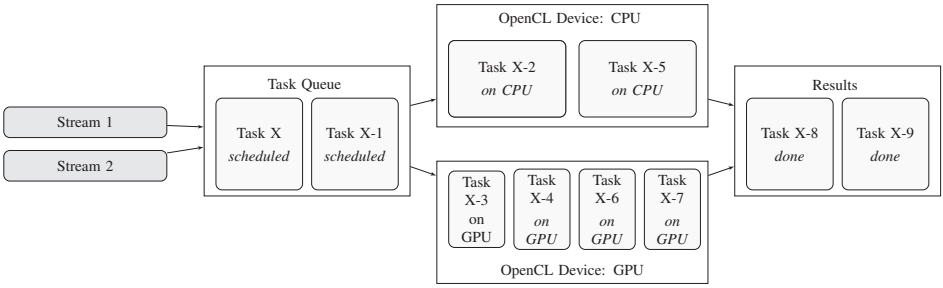


Figure 1: Queuing approach for a heterogeneous environment with a CPU and a GPU, both executing tasks from the same queue. The width of the task boxes within the devices symbolize their runtime.

placeholder. Taken a CPU with integrated Graphic Processor, for example, there are two devices, the CPU and GPU both on one processor chip. In OpenCL, copying the data to the CPU is nearly free because no data is copied physically. The data for the CPU is only referenced and the kernel is therefore working on the original data. Since the original data is not modified, there is no conflict with further processing steps. For the GPU, however, the data has to be physically copied. For the processors we aim for, the GPU shares the main memory with the CPU and thus the data copying takes a similar time to the memcpy instruction. If the GPU is connected through an external PCIe interface, then data copying is likely to be more time consuming.

3.2 Kernel Implementations

An OpenCL kernel is a code fragment, which can be run on all devices supporting OpenCL. To evaluate the performance of our stream join, we implemented 3 different kernels in OpenCL and 2 implementations not using OpenCL.

The first of our two non-OpenCL implementations is a **single-threaded** implementation, which processes the tasks sequentially and also performs the nested-loop comparisons within a task in sequential order. The single-threaded version runs only on the CPU. The second non-OpenCL implementation is a multi-threaded version for the CPU, implemented with **OpenMP**. This version also processes the tasks sequentially but parallelizes the nested-loop implementation within a task. We also tested running the tasks in parallel with OpenMP but we did not see a performance benefit.

The three OpenCL kernel work all in a similar way: each kernel is executed N times, where N is the number of rows in a join column. Each instance of the kernel has an unique ID. The ID is used by the kernel instance to determine the row in the first column it has to access. This row will then be compared to all entries in the second column. In this stage of our work, the kernel only counts the join partners found and does not return the join tuple. This way we know that the result is only one number per kernel instance and space for the result can be pre-allocated.

OpenCL device	AMD A10-5800K APU with Radeon HD Graphics CPU AMD A10-5800K	 GPU AMD Radeon HD 7660D
OpenCL compute units	4	6
Max parallel work-items	4 (cores)	384 (Shader)[1]
Global mem size	30,108 MB	1024 MB
Frequency	3.8 GHz	0.8 GHz

Table 1: Test platform with two compute devices and their properties.

The kernels differ in the way how they evaluate the join predicates and and increment the number of result tuples. **BranchedCL** works like explained above. During the kernel execution, the sum of join partners is incremented using a branch that evaluates whether the join predicate is true for two tuples.

The second kernel version is called **NoBranchCL**. In this kernel, we remove the branch using predication. The comparison result, which is either 0 or 1, is directly added to the sum with every comparison. This results in much more write operations but branching is avoided, which may beneficial for the CPU and GPU. The CPU could benefit from a better usage of the instruction pipeline. With branching the CPU has to guess which branch will be used next and wrong predictions could lead to performance loss. With a branch free algorithm this performance loss is avoided. The GPU can only execute in SIMD (Single Instruction Multiple Data) fashion. This means one instruction is run on all the data and then the next instruction is processed. When the execution is branched, the kernel instances not entering the branch are waiting idle for the kernel instances entering the branch. This idle time could be avoided with branch free algorithms.

Our final kernel is called **VectorizedCL**. Here, we use the branch free algorithm but instead of doing one compare operation at a time (in a kernel instance), we use OpenCL build-in SIMD instructions to compare 4 values at one time. This is done with vectors holding 4 values. Our first vector holds the search key on all positions. This vector is compared periodically with a second vector, which holds data from the second column. If the hardware supports SIMD instructions, the vector comparison can be done in the same amount of clock cycles as a normal comparison, resulting in an optimal speedup of 4x. Our test platform presented in Table 1 supports SIMD instructions on both devices.

4 Evaluation

We implemented the stream join algorithm in OpenCL with our dynamic load balancing approach. For a detailed evaluation, we use the 5 kernels as stated before. We are aware that in most scenarios with GPUs, the memory bandwidth of the PCIe bus is a limiting factor. But new hardware trends tend to include a graphic processor in the main processor. This could bring speedups in the data transfer rates because the GPU is accessing portions

[1]There are 6 SIMD units with each 16 thread processors with each 4 ALUs (Shader).

of the system main memory. Two current CPUs with integrated graphics are the Intel Ivy Bridge Processor and the AMD Trinity APU (Accelerated Processing Unit). We choose the AMD Trinity APU for our first tests, because the architecture seems to be more optimized for heterogeneous algorithms. The specifications of our test system are shown in Table 1. All tests presented were done on this system. We also tested our implementation on other systems with similar results.

In our implementation, we are able to select specific devices that access the job queue. We therefore tested our kernels on the CPU only, on the GPU only and both together. Having the tests limited to one specific device, it is possible to observe the performance differences of the kernels on these devices. For all our tests we used the same test parameters, which are shown in Table 2.

Property	Value
Initial No. of tasks	500
Task consists of	column1, column2, Spaceholder for result
Task size	300 KB (100 KB each)
No. of values per column	25,600
Comparisons per task	655.36 million
Started work-items in OpenCL	25,600 per task

Table 2: Overview on test parameters.

Our first test was done on the CPU and the results are illustrated in Figure 2. We have the single-threaded kernel (1 thread) and the OpenMP kernel on the left side of the figure. The OpenMP kernel achieves a speedup of 3.5x compared to the single-threaded approach, which is as expected for a 4-core CPU system. The right side of Figure 2 shows the OpenCL kernels run on the x86 CPU. The BranchedCL kernel runs similar to the OpenMP kernel. The overhead of OpenCL is responsible for the slightly worse performance. We can see, that the CPU benefits from avoiding branching and has a speedup of 1.9x compared to the kernel using branching. This is caused by a better instruction pipelining through avoided branch misses. Using SIMD with vectorized comparisons gives a further speedup of 3.6x, which is close to the expected optimal speedup for hardware supported SIMD instructions. The speedup is not 4x because of setup costs for the vectors. The vectorized OpenCL kernel on the CPU has a speedup of 21.3x compared to the single-threaded version.

Figure 3 shows the same test as before but here the OpenCL kernels were executed on the GPU. The time measurement for the GPU kernel includes all data transfers needed. We see a much higher speedup for the initial BranchedCL kernel. This is caused by the higher core count of the GPU. More surprising is the worse performance of the NoBranchCL kernel. We suppose that the additional write costs have a bigger influence on performance than branching. Having no branching, every comparison results in a write operation to the sum variable (increasing the value by 0 or 1). Branching on the GPU results in some parts of the work-item waiting for the work-items that are branching and staying idle. We suspect that incrementing the sum variable is a relatively fast operation, so that the idle time for the not branching work-items is small. Also without branching, all work-items have to write

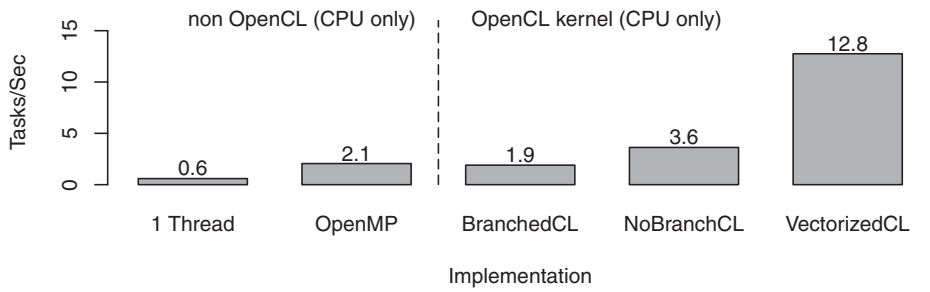


Figure 2: All implementations run on the CPU. The best performing OpenCL kernel has a speedup of 21.3x compared to single-threaded performance.

their sum variable while with branching some work-items would simply wait. Anyway the branch free algorithm is needed for the vectorization, which then brings a speedup of 2.4x compared to the not vectorized branch free algorithm and a final speedup of 47.5x compared to the single-threaded kernel.

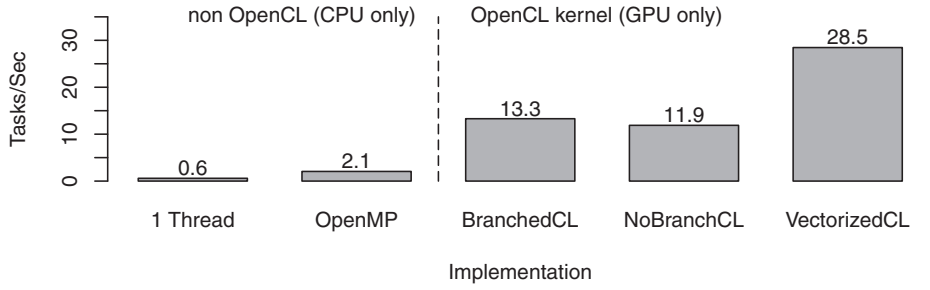


Figure 3: The first two implementations run on the CPU, the other three on the integrated GPU. The best performing OpenCL kernel has a speedup of 47.5x compared to single-threaded performance.

The final test was running our implementation using CPU and GPU for the OpenCL kernel. Here we only tested the vectorized version as the version with the most performance on both devices. Figure 4 illustrates the results. The first 4 bars are taken from previous tests and the last bar shows the performance of the VectorizedCL kernel running on both devices simultaneously. We see that the performance of both devices is added to a final processing throughput of 69.2 tasks per second (meaning 27 billion nested-loop comparisons per second, including transfer times). The final speedup to the single-threaded version is 69.2x and compared to the OpenMP variant the speedup is 19.8x.

5 Open Issues for Future Work

Having good performance results, still we left some issues open for discussion. We did not implement a join that returns result tuples but a join that returns the number of result tuples

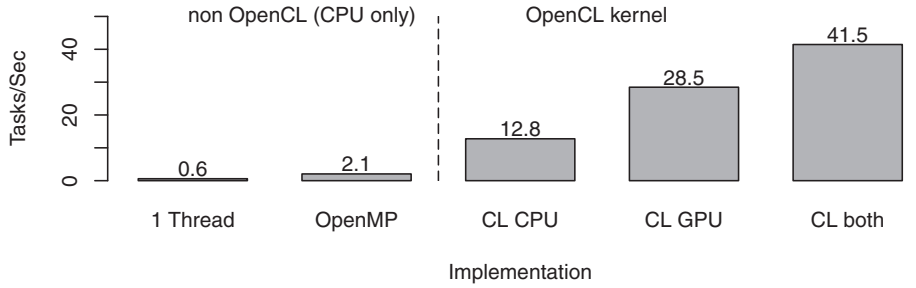


Figure 4: The first two implementations run on the CPU, while the OpenCL vectorized kernel runs either on the CPU, the GPU or on both devices. Running on both devices shows a speedup of 69.1x compared to single-threaded performance.

found. Returning a set of result tuples without prior knowledge of the result size, is not trivial with OpenCL. It is not possible to allocate memory dynamically within an OpenCL kernel. For our current implementation, the host code only allocates 4 bytes per started work-item to store the individual results. Besides the stream join, it would be possible to port other stream algorithms to our dynamic load balancing approach. We hope that our approach enables various algorithms to efficiently use heterogeneous hardware but we did not test that yet. Also it is worth thinking about dedicating specific tasks to the specific devices. While loosing the flexibility in load balancing, the performance could be better through more hardware specific optimizations. We also leave further testing for future work. We tested our approach on different AMD processors with integrated GPUs. We did not test Intel Core third generation chips (Ivy Bridge) or systems with distinct GPUs.

6 Conclusion

In this paper, we presented our novel approach to use a heterogeneous environment with all it's computing devices to speed up a stream join with band condition. The join is performed within windows of the stream and scheduled as a task within a queue. New values can be joined in batches with the window of the other stream. The compute devices select a task from the queue, execute the task, and return the results before selecting the next task. Executing a task involves copying the data if needed, the execution of the band join, and returning of the data. At the moment our implementation is limited to returning only the amount of join partners, not the joined tuples. With our heterogeneous approach we achieve a speedup of 69.2x compared to the single-threaded implementation on our test machine. In our final version, we use 3 levels of parallelization: parallel execution between devices, parallel execution on devices (between cores, respectively shaders) and using build-in SIMD instructions for execution of 4 comparisons at once. Also the heterogeneous execution is 3.2x faster than the OpenCL implementation only on the CPU (respectively 1.5x compared to GPU).

Seeing modern processors becoming more and more heterogeneous and proven by our

results, we believe using these heterogeneous environments effectively can have a big impact in computing and database acceleration.

7 Acknowledgments

This work was in part funded by the European Union and the state Saxony through the ESF young researcher group IMData 100098198.

References

[Agg06] Charu C. Aggarwal. *Data Streams: Models and Algorithms (Advances in Database Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[DNS91] David J. DeWitt, Jeffrey F. Naughton, and Donovan A. Schneider. An Evaluation of Non-Equijoin Algorithms. In *Proceedings of the 17th VLDB '91*, pages 443–452, San Francisco, CA, USA, 1991.

[GO03] Lukasz Golab and M Tamer Özsu. Processing sliding window multi-joins in continuous queries over data streams. In *Proceedings of the 29th VLDB '03*, pages 500–511, 2003.

[Gro] Khronos Group. http://www.khronos.org/opencl/.

[GYB07] Buğra Gedik, Philip S. Yu, and Rajesh R. Bordawekar. Executing stream joins on the cell processor. In *Proceedings of the 33rd VLDB '07*, pages 363–374. VLDB Endowment, 2007.

[HYF+08] Bingsheng He, Ke Yang, Rui Fang, Mian Lu, Naga Govindaraju, Qiong Luo, and Pedro Sander. Relational joins on graphics processors. In *Proceedings of the ACM SIGMOD '08*, pages 511–524, New York, NY, USA, 2008.

[KLMV12] Tim Kaldewey, Guy Lohman, Rene Mueller, and Peter Volk. GPU join processing revisited. In *Proceedings of the DaMoN '12*, pages 55–62, New York, NY, USA, 2012.

[LT95] Hongjun Lu and Kian-Lee Tan. On Sort-Merge Algorithm for Band Joins. *IEEE Trans. on Knowl. and Data Eng.*, 7(3):508–510, June 1995.

[SD89] Donovan A. Schneider and David J. DeWitt. A performance evaluation of four parallel join algorithms in a shared-nothing multiprocessor environment. In *Proceedings of the ACM SIGMOD '89*, pages 110–121, New York, NY, USA, 1989.

[Sol93] Valery Soloviev. A Truncating Hash Algorithm for Processing Band-Join Queries. In *ICDE*, pages 419–427. IEEE Computer Society, 1993.

[TM11] Jens Teubner and Rene Mueller. How soccer players would do stream joins. In *Proceedings of the ACM SIGMOD '11*, pages 625–636, New York, NY, USA, 2011.

[VNB03] Stratis D. Viglas, Jeffrey F. Naughton, and Josef Burger. Maximizing the output rate of multi-way join queries over streaming information sources. In *Proceedings of the 29th VLDB '03*, pages 285–296, 2003.

Bestimmung der semantischen Eigenschaften von Datenstromsystemen durch Black-Box-Tests

Frank Lauterwald, Niko Pollner, Klaus Meyer-Wegener
Lehrstuhl für Informatik 6 (Datenmanagement)
Friedrich-Alexander-Universität Erlangen-Nürnberg
{frank.lauterwald, niko.pollner, klaus.meyer-wegener}@fau.de

Abstract: Die Semantik von Datenstromsystemen (DSS) ist bislang nicht standardisiert. Für Anwendungsentwickler ist es jedoch wichtig zu wissen, wie sich ein bestimmtes System in einer bestimmten Situation verhält. Ebenso bedeutsam ist das Verhalten für föderierte Datenstromsysteme, die Anfragen automatisch auf verschiedene DSS verteilen. Als Hilfsmittel zur Beschreibung können semantische Modelle dienen. Diese werden parametrisiert und können durch verschiedene Parameterwerte das Verhalten verschiedener Systeme nachbilden. Da bisher auch kein allgemein anerkanntes Modell zur Beschreibung von DSS existiert, muss man sich möglicherweise mit verschiedenen Modellen auseinandersetzen. Daher wäre es hilfreich, die Bestimmung der jeweiligen Parameterwerte weitgehend zu automatisieren, wozu dieser Beitrag eine geeignete Evaluationsumgebung vorstellt. Diese vergleicht die Ausgaben eines DSS mit allen Vorhersagen, die ein Modell für verschiedene Parameter machen kann. Stimmen die Ergebnisse überein, sind die Parameter gefunden. Erfahrungen damit und Beschränkungen dieses Ansatzes werden diskutiert.

1 Einleitung und Motivation

Datenstromsysteme (DSS) finden zunehmenden Einsatz in realen Szenarien. Jedoch mangelt es bisher an ihrer Standardisierung, wodurch die Interoperabilität eingeschränkt und die Portierung von Anwendungen erschwert wird.

Die DSS unterscheiden sich insbesondere bezüglich der bereitgestellten Funktionalität, der Anfragesprachen und des Verhaltens.

Es ist zu erwarten, dass die fehlende Standardisierung aufgrund des zunehmenden Einsatzes von Datenstromsystemen in Zukunft ein immer größeres Problem darstellt. Allerdings zeigen sich in diesem Bereich auch einige Hoffnungsschimmer: Zum einen ist eine beginnende Konsolidierung im Markt der kommerziellen DSS zu sehen. Zum zweiten wurden in der Forschung bereits einige semantische Modelle entwickelt (z.B. [BDD+10]), die das Verhalten verschiedener DSS in einer gemeinsamen Terminologie beschreiben können. Zum dritten verspricht die Entwicklung föderierter DSS (z.B. [BCD+09]), die Unterschiede zwischen verschiedenen Systemen hinter einer gemeinsamen Schnittstelle zu verbergen.

Dieser Artikel vereint die letzten beiden Punkte, also semantische Modelle und föderierte DSS. Die behandelte Fragestellung ergibt sich aus den Erfahrungen der Autoren im von

ihnen durchgeführten DSAM-Projekt [DLF+10]. Der Data Stream Application Manager (DSAM) ist ein föderiertes DSS, das globale Anfragen auf ein Netzwerk heterogener DSS verteilt. Die Verteilungsentscheidung wird durch einen kostenbasierten Optimierer getroffen. Der Anwendungsentwickler wird somit von der Aufgabe befreit, selbst zu entscheiden, welche Anfrageteile auf welchem System ausgeführt werden, wodurch er aber auch keinen direkten Einfluss auf die Verteilung mehr hat. Dennoch muss sichergestellt werden, dass eine Anfrage die von ihm gewünschten Ergebnisse liefert. Eine formale Beschreibung des Verhaltens der verwendeten Systeme ist dazu unabdingbar. Der Beitrag dieses Artikels besteht aus einer Methode zur teilautomatischen Bestimmung des Verhaltens von DSS sowie einer Evaluation dieser Methodik am Beispiel des SECRET-Modells [BDD+10].

2 Methode

Black-Box-Tests haben sich bereits bei der Perfomance-Analyse von DSS bewährt [DLB+11]. Da für viele DSS kein Quellcode verfügbar ist, sind andere Methoden nicht anwendbar. Die Black-Box-Methode soll nun auch zur automatischen Bestimmung der semantischen Eigenschaften von DSS genutzt werden. Diese werden dabei durch Parameterwerte innerhalb eines semantischen Modells beschrieben. Als semantische Modelle werden hier solche Modelle bezeichnet, die das Verhalten von DSS zumindest in Teilaspekten beschreiben können und damit Vorhersagen über deren Anfrageergebnisse (für bekannte Eingabedaten und Anfragen) zulassen. Parametrisierbare Modelle können durch die Wahl geeigneter Parameterwerte die Anfrageergebnisse verschiedener DSS vorhersagen. Einzelne Aspekte des Systemverhaltens werden als Dimensionen bezeichnet, die idealerweise durch unabhängige Parameter beschrieben werden. Das hier beispielhaft verwendete SECRET-Modell ist ein solches Modell, das sich durch hohe Ausdrucksmächtigkeit auszeichnet. Es definiert Parameter in vier verschiedenen Dimensionen: *ScopE* beschreibt, über welche Zeitabschnitte Fensterinstanzen definiert sind; *Content* definiert, welche Tupel jeweils in diesen Fensterinstanzen enthalten sind; *Tick* gibt an, aufgrund welcher Ereignisse ein System aktiv wird (z.B. Eintreffen eines Tupels oder Fortschreiten der Zeit); *REport* schliesslich erlaubt die Angabe weiterer Kriterien, die erfüllt sein müssen, damit das System eine Ausgabe erzeugt.

Der Aufwand für die Parameterbestimmung steigt mit der Anzahl betrachteter DSS und Modelle. Daher sollte die Parameterbestimmung möglichst automatisch erfolgen.

Die dazu unternommenen Schritte werden im Folgenden beschrieben. Die erzielten Ergebnisse werden dann in Abschnitt 3 erläutert.

2.1 Simulator

Zunächst wurde ein Simulator in Python entwickelt. Dieser ist im Kern ein minimales DSS, das sich durch Konfiguration so verhalten kann wie jedes durch SECRET beschreibbare DSS. Ein- und Ausgabe erfolgt durch einfache CSV-Dateien. Da die nachfolgenden

Schritte von der Korrektheit des Simulators abhängen, muss er zunächst validiert werden. Dazu können die Testdaten aus [BDD+10] verwendet werden, sowie die Ausgaben des Simulators mit denen realer DSS verglichen werden.

2.2 Anfragen und Testdaten

Das SECRET-Modell befasst sich mit dem Verhalten zeitbasierter Fenster. Es kann erklären, wann Fenster ausgewertet werden und welche Tupel sie zu diesem Zeitpunkt enthalten. Daher wurde die einfachst mögliche Testanfrage verwendet, mittels derer diese Frage beantwortet werden kann:

```
SELECT min(id), max(id) FROM InputStream [keep 3 seconds]
```
[1]

Diese Anfrage gibt die niedrigste und höchste id der Tupel in einem Fenster aus. Verwendet man nun Testdaten mit aufsteigenden ids, so geht aus dem Anfrageergebnis eindeutig hervor, welche Tupel zum Auswertungszeitpunkt in einem Fenster waren.

Die Testdaten werden iterativ erzeugt: Der Simulator wird mit allen möglichen Parameterkombinationen konfiguriert und die Anfrageergebnisse werden aufgezeichnet. Führen zwei oder mehr Parameterkombinationen zu identischen Ergebnissen, so werden gezielt Testdaten hinzugefügt, die diese Kombinationen unterscheidbar machen sollen. Ist dies nicht möglich, wird der Grund dafür analysiert. Dieses Vorgehen wird wiederholt, bis alle unterscheidbaren Kombinationen auch unterscheidbare Ergebnisse liefern und für alle nicht-unterscheidbaren ein Grund angegeben werden kann. Ein Nachteil dieses Vorgehens ist, dass es möglicherweise eine große Anzahl an Testdatensätzen erfordert. Günstiger wäre es, die einzelnen Dimensionen des Modells auch einzeln testen zu können. Dies ist allerdings schwierig: Beispielsweise können bestimmte Parameterwerte in verschiedenen Dimensionen zur Unterdrückung von Ausgaben führen. Somit liefert ein Test in einer Dimension kein Ergebnis, wenn die Ergebnisse durch den Wert einer anderen Definition unterdrückt werden.

2.3 Parameterbestimmung von Datenstromsystemen

Um die semantischen Eigenschaften realer DSS zu testen, wird die Testanfrage auf diesen mit allen im vorherigen Schritt erzeugten Testdaten ausgeführt und die Ergebnisse werden aufgezeichnet. Diese Ergebnisse werden dann mit den Ausgaben des Simulators für alle möglichen Parameterkombinationen verglichen. Die Ausgabe des realen Systems sollte mit der Ausgabe des Simulators für genau eine Parameterkombination identisch sein. Die semantischen Parameter des realen Systems lassen sich unmittelbar daraus ablesen, mit *welchen* Parameterwerten der Simulator identische Ergebnisse erzeugt.

[1] Natürlich muss diese Anfrage in die jeweiligen Anfragesprachen der getesteten Systeme übersetzt werden.

3 Ergebnisse

Im diesem Abschnitt werden die Ergebnisse der Evaluation der vorgestellten Methoden auf Basis des SECRET-Modells beschrieben.

3.1 Modell und Simulator

Die Ergebnisse aus [BDD+10] konnten mit einigen Abweichungen manuell nachvollzogen werden.[2] Diese Ergebnisse dienen als Grundlage für die weiteren Versuche. Die automatische Parameterbestimmung soll also für jedes DSS (z.B. Coral8) die gleichen Ergebnisse liefern wie dieser manuelle Vorversuch.

Der Simulator wurde mittels der in [BDD+10] beschriebenen Testdaten validiert. Es wurde also getestet, ob der Simulator die in [BDD+10] beschriebenen Ergebnisse liefert. Als zweiten Validierungsschritt wurde der Simulator mit den bekannten Parametern für die in [BDD+10] beschriebenen realen DSS konfiguriert und es wurde getestet, ob er dieselben Ergebnisse erzeugt wie diese Systeme. Mit diesem Test wurde sichergestellt, dass der Simulator die Ergebnisse realer DSS korrekt simuliert.

3.2 Testdaten

Ein Ziel war es, Testdaten zu erzeugen, die für jede Ausprägung von SECRET-Parametern unterschiedliche Anfrageergebnisse liefern. SECRET unterscheidet drei Werte für *Tick* und 16 Kombinationen der *REport*-Flags. Für *ScopE* sind beliebige Formeln möglich [3], allerdings verwenden die in [BDD+10] beschriebenen Systemen nur 2 unterschiedliche Formeln. Daher wurden auch nur diese in die Untersuchung einbezogen. Durch Ausmultiplizieren ergibt sich eine Maximalzahl von 96 unterschiedlichen Parameterkombinationen.

Neben den SECRET-Parametern sind weitere Details der Anfrage zu spezifizieren (z.B. die Fenstergröße). Diese werden zu *Konfigurationen* zusammengefasst. Die Versuche wurden in zwei Konfigurationen, T_1 und T_2 durchgeführt, die sich im Wert des Slide-Parameters und der Häufigkeit von Ausgaben unterscheiden. [4] Die Werte in T_1 wurden für größtmögliche Unterscheidungskraft gewählt. So sind 84 der 96 möglichen Klassen anhand ihrer Ausgaben voneinander unterscheidbar. Allerdings sind bei vielen Systemen weder Slide noch Reporting Frequency frei wählbar. In T_2 sind diese Werte daher derart belegt, dass alle betrachteten Systeme mit ihr konfiguriert werden können. Hierdurch sinkt die Ausdrucksmächtigkeit deutlich und es können nur noch 15 Klassen unterschieden wer-

[2] Insbesondere verhielt sich in unseren Versuchen StreamBase zeitgetrieben und Coral8 konnte nicht batch-, sondern nur tupelgetrieben betrieben werden. Außerdem schien Coral8 nur für $\lambda = 1$ auf *non-empty content* zu reagieren. Solche Abweichungen stellen für die vorliegende Arbeit kein Problem dar.

[3] *ScopE* beschreibt u.a. den Startzeitpunkt des ersten Fensters

[4] T_1: Fenstergröße 5, $\lambda = 3$, $\beta = 2$; T_2: Fenstergröße 5, $\lambda = 1$, $\beta = 1$; die Fenstergröße wurde gewählt, um in T_1 teilerfremd zu λ und β zu sein, womit z.B. das Zusammenfallen der *Window-Close* und *Periodic*-Ereignisse verhindert wird.

den. Eine detaillierte Erklärung würde den Rahmen dieses Beitrags sprengen; festzuhalten ist aber, dass für alle ununterscheidbaren Klassen nachvollzogen werden konnte, warum sie nicht unterscheidbar sind. Dazu waren insgesamt etwa 10 verschiedene Tests mit unterschiedlichen Testdaten nötig. Die Testdaten zur Unterscheidung der Klassen sind dabei konfigurationsspezifisch. Um den Aufwand der Testdatenerstellung klein zu halten, sollte die Anzahl der Konfigurationen also klein gehalten werden.

4 Zusammenfassung und Ausblick

Föderierte Datenstromsysteme sind auf eine formale Beschreibung der angebundenen DSS angewiesen, um definierte Anfrageergebnisse erzielen zu können. Falls ein geeignetes semantisches Modell vorliegt, können diese durch *black-box*-Tests weitgehend automatisch ermittelt werden. Dies wurde am Beispiel eines komplexen Modells evaluiert.

Die Erweiterung um zusätzliche DSS und Modelle wird aktuell untersucht. Nach ersten Erkenntnissen ist dabei die Verwendung anderer DSS mit relativ geringem Aufwand möglich, während zur Verwendung eines anderen Modells zumindest Simulator und Testdaten weitgehend neuzuentwickeln sind.

Als Fernziel sollen die Portierung von Datenstromanwendungen und die Entwicklung föderierter DSS unterstützt werden. Daneben wird ein Nutzen für die Neu- oder Weiterentwicklung semantischer Modelle erwartet.

Literatur

[BCD+09] I. Botan, Y. Cho, R. Derakhshan, N. Dindar, L. Haas, K. Kim, C. Lee, G. Mundada, M. Shan, N. Tatbul, Y. Yan, B. Yun, and J. Zhang. Design and Implementation of the MaxStream Federated Stream Processing Architecture. Technical Report TR-632, ETH Zurich Department of Computer Science, June 2009.

[BDD+10] I. Botan, R. Derakhshan, N. Dindar, L. Haas, R. J. Miller, and N. Tatbul. SECRET: A Model for Analysis of the Execution Semantics of Stream Processing Systems. In *VLDB 2010*, pages 232–243, Singapore, September 2010.

[DLB+11] M. Daum, F. Lauterwald, P. Baumgärtel, N. Pollner, and K. Meyer-Wegener. Black-box Determination of Cost Models' Parameters for Federated Stream-Processing Systems. In *IDEAS 2011*, pages 226–232, Lisbon, Portugal, September 2011.

[DLF+10] M. Daum, F. Lauterwald, M. Fischer, M. Kiefer, and K. Meyer-Wegener. Integration of Heterogeneous Sensor Nodes by Data Stream Management. In Takahiro Hara, Vladimir I. Zadorozhny, and Erik Buchmann, editors, *Wireless Sensor Network Technologies for the Information Explosion Era*, volume 278 of *Studies in Computational Intelligence*, pages 139–172. Springer, 2010.

Event Processing on your own Database

Nikolaus Glombiewski, Bastian Hoßbach,
Andreas Morgen, Franz Ritter, Bernhard Seeger

Department of Mathematics and Computer Science
University of Marburg, Germany
{glombien,bhossbach,morgen,ritterf,seeger}@mathematik.uni-marburg.de

Abstract: Event processing (EP) is widely used for reacting on events in the very moment when they occur. Events that require immediate reactions can be found in various applications, e.g. algorithmic trading, business process monitoring and sensor-based human-computer interaction. For the support of EP in an application, a special-purpose EP system with its own API and query language has to be used. Typically, EP systems as well as their integration in applications are expensive. In this paper, we show how every standard database system can be used as an EP system via JDBC. An experimental evaluation proofs that databases behind JDBC are able to support small and medium-sized EP workloads.

1 Introduction

Today, many existing applications benefit from *event processing* (EP) or are enabled by using EP [Luc01]. Algorithmic trading, business process monitoring and sensor-based human-computer interaction are just a few examples of applications that require or take advantage of EP. In order to support EP in an application, a special-purpose EP system has to be used. After the integration, the application code and the EP system are intermeshed (vendor lock-in). Therefore, replacing the EP system for a different one becomes expensive.

In this paper, we motivate and present the implementation of an EP system purely on top of JDBC that enables every standard database system to provide EP functionality. In contrast to existing approaches (e.g. Oracle CEP [ora13] or StreamInsight [G$^+$09]), an EP system via JDBC is completely independent of specific database products. Therefore, different databases can be used and easily exchanged behind JDBC without affecting the EP implementation on top. Because databases are already integrated in most applications, our approach leads to low costs.

After an brief overview of EP in Section 2, we motivate our approach in Section 3 and present implementation details in Section 4. The proposed approach leads to two advantages. First, every standard database system can provide EP functionality. Second, the user can easily exchange different database products as EP back-end in Java applications. We show in Section 5 that database systems can support small and medium-sized EP workloads via JDBC. Section 6 concludes the main results of this paper.

2 Event Processing

The continuous analysis of streaming events is called *event processing* (EP). EP allows to react immediately when specific events occur. An event is a pair (p, t) consisting of a payload p that is some information about a real or virtual event and a timestamp t that specifies the instant of time when the event occurred. Typically, EP is used to combine events that occurred in a specific order or within a timeframe. For example, an application running on a touch screen device can wait for the user to put exactly three fingers into the upper right quarter of the screen within two seconds (correlation of events). Or a computer game can detect that a player has picked up some item at place A and brought it to place B before a predefined mission timer elapsed (pattern matching of events). Besides correlation and pattern matching, filtering and aggregating events are other important basic EP operators. These four basic EP operators can be combined arbitrarily to express more complex queries. Because every new input event can produce new results, all queries are performed continuously in an event-driven manner.

The implementation of EP in an application comprises three steps that are the core functionality of every EP provider. First, all event stream producing sources (e.g. sensors) are registered at the EP provider. In general, they have a fixed and structured schema that is send to the EP provider during registration. Second, the EP logic is created in the form of a set of continuous queries. Third, sinks (e.g. alarms) are registered at the output side to consume the results of the continuous queries.

3 Motivation

In this section, we give insights into our motivation that led to the development of an EP system on top of JDBC. Inspired by the database abstraction layers ODBC/JDBC, we developed a generic EP abstraction layer that allows the connectivity to different EP providers as well as the building of distributed and federated EP infrastructures. In contrast to databases, the design of such an interface is more complex for the following reasons. First, there is no standard language for EP, perhaps comparable to SQL for databases. The challenge is to design an interface that is powerful enough, but still allows a bridge to any kind of EP provider. Second, operators like pattern matching are not easy to adapt to the underlying EP providers. Our approach for an EP abstraction layer, entitled *Java Event Processing Connectivity*[1] (JEPC), is currently implemented on top of the EP systems *Esper* [esp13] (open source), *Odysseus* [A$^+$12] (academia) and *webMethods Business Events* [web13] (commercial) as shown in Figure 1. Here in this paper, we put our focus on the implementation of an JEPC-to-JDBC bridge. Such a bridge seems to be very appealing for reasonably small EP applications that do not require installing a special-purpose EP system. However, the JDBC bridge can also be used stand-alone to enable every standard database to provide EP functionality. In the following, the most important aspects of JEPC are introduced, because it is used as user interface for our EP system on top of JDBC.

[1] `http://dbs.mathematik.uni-marburg.de/research/projects/jepc/`

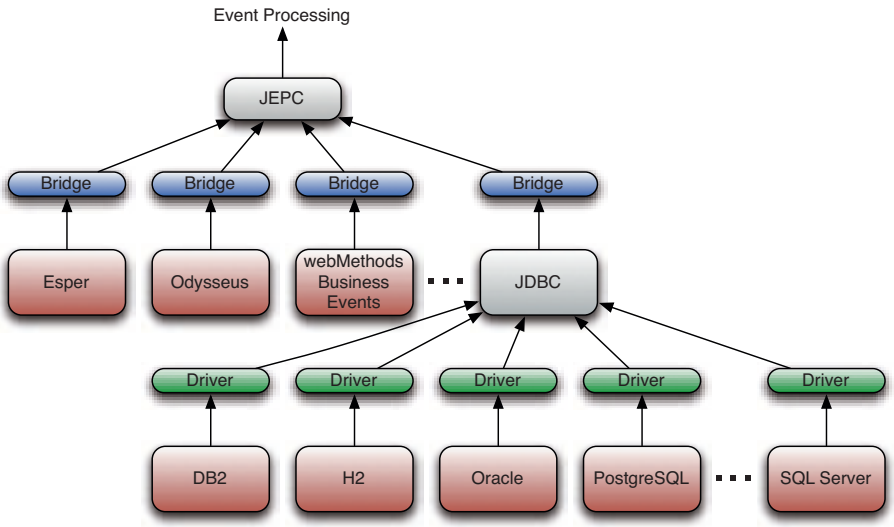


Figure 1: Java Event Processing Connectivity

To abstract EP functionality from specific EP providers, we have designed the JEPC interface shown in Table 1. However, only a unified API is not sufficient. We also have to abstract from the query language, the representation of streams and how query results are made available.

The method to register a new event stream requires a unique identifier e for later accesses and a schema definition s of the stream. Because we want the interface to be generally applicable but also simple, we decided to support the relational model. Thus, a schema is an array of attributes; each attribute consists of a name and a primitive data type.

In the following, we show how to express continuous queries in JEPC. A new query is created by giving it a unique identifier q and a definition d. The query definition makes use of a simple object representation of streams, four basic EP operators and their parameters. This representation has several advantages. First, queries can be composed arbitrarily to build more complex ones. Second, queries can be processed easily. This is very important when an implementation of the JEPC interface (like the bridges in Figure 1) has to translate the input query definition into the specific query language of the underlying EP provider with preservation of semantics. Third, a query definition of JEPC can also be generated easily. This allows the use of query languages, domain specific languages, integrated languages (e.g. LINQ [Mei11] in .NET) and graphical query composers on top of the JEPC interface.

In JEPC, query definitions follow the paradigm everything-is-an-object. A stream is represented by an object that contains all necessary information about the stream (e.g. its name). Each stream object must be related to a parameter object of the type time window [ABW06]. This object holds all information about the time window (e.g. its size). The next group of objects consists of the basic EP operators, namely the filter, the aggregator,

Method	Description
REGISTERSTREAM(e, s)	Registers a new event stream e with schema s
CREATEQUERY(q, d)	Creates a new continuous query q with definition d
ADDOUTPUTPROCESSOR(q, o)	Adds an output processor o to query q
PUSHEVENT(e, p, t)	Pushes the event (p,t) into the event stream e
REMOVESTREAM(e)	Removes the stream e
REMOVEQUERY(q)	Removes the query q
REMOVEOUTPUTPROCESSOR(o)	Removes the output processor o

Table 1: JEPC Interface

the correlator and the pattern matcher. Each of them is related to either one input object (filter, aggregator and pattern matcher) or multiple input objects (correlator). An input object can be a stream object as well as an operator object. On the output side, there is exactly one related output object. This can be another operator object or the final query output object that exists exactly once in every query definition. Each type of operator object needs its own specific parameter object. The filter and the correlator need a boolean expression that specifies all events (filter) respectively combinations of events (correlator), which should be forwarded to the output object. The aggregator needs an aggregate in order to compute its output events and the pattern matcher needs a pattern. Each element of a parameter object (e.g. literals and logical connectors in boolean expressions) is an object again. For the sake of simplicity, we do not mention them in detail here. Because everything is an object, entire parameters and query definitions are formed by composing objects. The final composition determines precisely all relations between the objects and makes it easy to traverse and translate the query definition as well as to create and modify it. In other words, queries are expressed in the form of a composition of primitive operators (directed EP operator graphs). Then, every bridge only needs a compiler that is able to translate the basic EP operators into the specific query language of the underlying EP provider with preservation of semantics. Following the well-known concept of subqueries, complex EP queries can be translated and executed as event processing networks [Luc01].

The semantics of JEPC queries is the same as the one presented in [KS09]. We have decided to adapt this one, because it is sufficiently expressive, deterministic, consistent and has a solid theoretical foundation (at each single point in time the queries produce the same results as standard SQL would do). Pattern matching is performed strictly sequential with time instant semantics [WDR06]. Each bridge guarantees to preserve the defined semantics.

To consume the results of a continuous query q, the user can add multiple output processors. Inside an output processor o, user-defined code is executed on each produced result. A new event (p, t) is pushed into a registered stream e by calling the corresponding method of the JEPC interface. Additionally, there are also methods needed to remove streams, queries and output processors.

4 JEPC-to-JDBC Bridge

In this section, we present our data structures and algorithms being used on top of JDBC to implement the JEPC interface. Time windows are the most important data structure of EP. They are used as building blocks to correlate and aggregate events. Pattern matching is the most challenging operator. For the sake of limited space, we assume that every input stream is ordered by time. But available techniques to handle out-of-order streams (e.g. on the basis of punctuations [TM03]) are fully compatible with our approach.

4.1 Query Translation

The JEPC-to-JDBC bridge translates each incoming JEPC query definition into standard SQL with additional clauses for time windows and pattern matching. See Listing 1 for examples of the syntax we use for the additional clauses. The main idea is to treat the SQL extensions accordingly so that the resulting statement will be pure standard SQL. Finally, the bridge can execute the resulting statements in the database via JDBC. Of course, it is also possible to express EP queries direct in the form of our extended SQL syntax without using JEPC (stand-alone mode).

4.2 Time Windows

Because event streams are potentially unbounded, they can not be queried as a whole. Instead, sliding time windows [ABW06] are used to keep a finite set of the freshest events and to perform the query execution. The sizes of time windows are specified on the source streams by the user. For all other streams obtained from an operator, the sizes of time windows are derived from the specification of the operator and its corresponding input streams. The JEPC-to-JDBC bridge creates for each time window a new database table. In the associated query statements, it replaces each source stream together with its time window specification by the corresponding database table. The resulting expression is already a valid SQL statement being ready for execution in a database system. Figure 2 visualizes this procedure. When a new event is added to an event stream, all dependent tables that represent a time window on it are updated [DR04]. Because the freshest event is the one that was recently added to the window, we can use its timestamp and the size of the corresponding time window to deduce an instant of time as a bound for purging. Events dating back before this bound are expired; therefore, they are deleted. In Figure 2 the update is shown for *table_a*. Afterwards, all tables only contain relevant events and it is possible to execute queries on them.

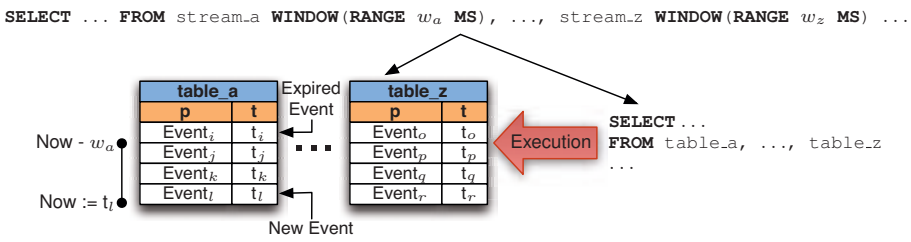


Figure 2: Time Windows

4.3 Filter, Correlation and Aggregation

Although we have stated in the last section that the rewritten queries can be executed on the simultaneously created tables such that all results are produced, we have to perform further modifications in order to achieve the behavior of an EP provider. The reason why we need to tweak the execution lies in the stream-based nature of EP. This especially means that the output of every operator must be an event stream again. So we are forced to report every result exactly once and to report all results ordered globally by time. An operator result is called new if the recently added event took part in its creation. Every EP operator has to report only new results.

To produce only new results, we created an environment capable of recognizing the recently added event. In our implementation, each single stream has an extra table that always contains the freshest event of the stream only. This allows to select it efficiently by querying this extra table instead of a time window. For the rest of the paper, we use the terms table and time window synonymously and call the extra table the *inbox* of a stream. Each stream has exactly one inbox and can have multiple time windows (the total count of time windows depends on the running EP operators).

After the common setup, the filter, correlation and aggregation operators are handled as follows. A filter basically checks whether an incoming event fulfills certain requirements or not. Thus, the filter is executed on each new event in the inbox and returns the result directly. An aggregation can not exploit the inbox, because it has to take every valid event inside a time window into consideration. So the time window on its input stream is updated and the aggregation is executed as described in the previous section. A correlation between event streams requires the inbox of the updated input stream in order to report only new results. The recently added event is inserted into its corresponding time window and expired events are removed in every time window on the input streams of the correlation with the timestamp of the recently added event as upper bound. The recently added event not only let the time progress in its corresponding time window but also in all others. Finally, the inbox that contains only the recently added event is correlated with all other time windows. This produces all results in that the recently added event takes part [BLT86].

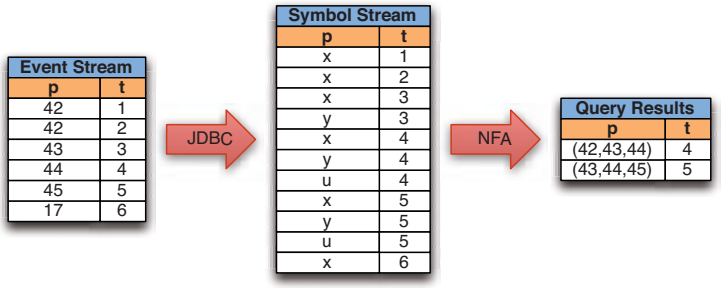


Figure 3: Pattern Matching Example

4.4 Sequential Pattern Matching

Creating a new pattern matching query via JEPC requires the following information in the query definition: a strictly sequential pattern that is described through a regular expression, boolean expressions that specify the mapping between events of the queried stream and the symbols used in the pattern, a time window within the pattern must occur completely, arbitrary global variables to hold values globally for the pattern and an output event consisting of a subset of the global variables. The test query ρ in Listing 1 shows the resulting SQL statement after the bridge has translated the incoming query definition. For each parameter there is an additional clause: the pattern-clause contains the regular expression, the within-clause contains the duration of the time window and the measures-clause declares all global variables. Inside the define-clause, each symbol is defined through a boolean expression (as-clause). This regular expressions can use constants, attributes of the event stream as well as previously set global variables as literals. The do-clause is used to set global variables when the corresponding symbol is emitted (that is the boolean expression was evaluated to true). Figure 3 shows the test query ρ in execution with i set to 1. In this case, the query detects all integer event sequences of size three. Furthermore, each integer event in a matching sequence must have a value that is decreased exactly by one in comparison to the value of its successor.

For each new event in the inbox of the stream, all symbols it emits are derived. Because symbols are defined through boolean expressions, this step is done completely inside the database. The result is a symbol stream that is maintained in the same way as every other event stream. Especially the specified time window is used to hold only relevant symbols. Also the values of global variables are derived and stored inside the database. Because in strictly sequential pattern matching all symbols with identical timestamps are interpreted as alternatives, we manage all alternative sequences contained in the symbol stream inside the database and give each of them a unique identifier. The same identifiers are used to tag the corresponding values of the global variables. All symbols and the identifiers are put into a *nondeterministic finite automaton* (NFA) that is built on the basis of the pattern. This NFA reports all identifiers of sequences that have a positive match with the pattern. Finally, the identifiers are used to load the corresponding values of the global variables in order to build and report the output event.

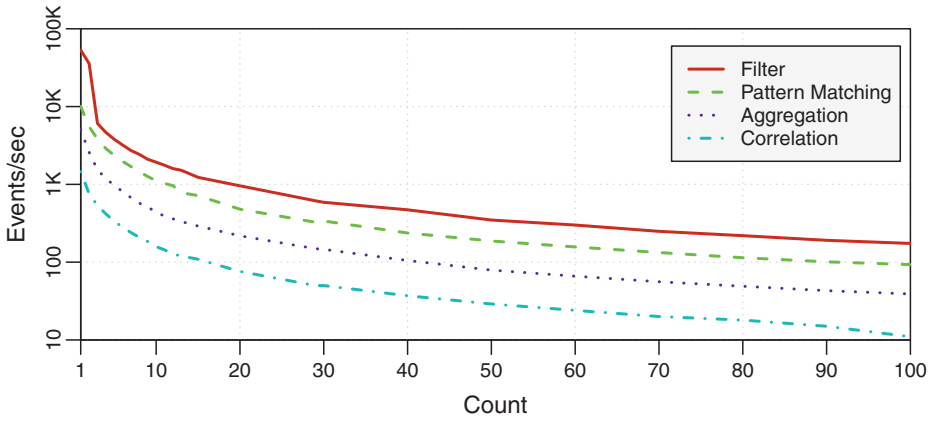


Figure 4: Workload Scalability

5 Evaluation

In order to obtain an impression of the performance of our EP system on top of JDBC, we conducted an experimental evaluation. Two things are remarkable about our implementation and test environment. First, our implementation is strictly single-threaded and consumes only few resources (in the worst case, one CPU core is fully utilized). We avoided to implement a multi-threaded EP server, because our target use-cases are applications that need embedded EP functionality on the same machine. Second, we used the database system *H2* [h2d13] in our experiments, because it is a simple Java library and in the presence of JEPC, users only have to add this library to their projects to enable EP. There is no complicated setup required.

5.1 Experiments

We have examined for each basic EP operator the number of input events being processed per second. We also have evaluated the scalability of all basic EP operators by running multiple of them at the same time. Our test machine had an Intel i7 CPU with 8 GB main memory. The database system *H2* used its standard configuration keeping its data on an ordinary hard disk. Figure 4 shows the results on a logarithmic y-axis.

```
σ(i,Count) = SELECT *
             FROM   stream[a:Integer, b:Integer]
             WHERE  a - b = i

⋈(i,Count) = SELECT *
              FROM   stream1[a:Integer] WINDOW(RANGE (500 + (Count/2) − i) MS),
                     stream2[b:Integer] WINDOW(RANGE (500 + (Count/2) − i) MS)
              WHERE  a - b = i + 1

α(i,Count) = SELECT COUNT(*)
             FROM   stream[a:Integer] WINDOW(RANGE (500 + (Count/2) − i) MS)

ρ(i,Count) = SELECT z1, z2, z3 FROM stream[a:Integer]
             MATCHING(PATTERN  xyu WITHIN (500 + (Count/2) − i) MS
                      MEASURES z1:Integer, z2:Integer, z3:Integer
                      DEFINE   x AS true       DO z1 = a
                               y AS a - z1 = i DO z2 = a
                               u AS a - z2 = i DO z3 = a)
```

Listing 1: Test Queries - Filter σ, Correlation $\bowtie$, Aggregation α and Pattern Matching ρ

In the experiments, exactly one event was pushed into all input event streams for every millisecond. So all adjacent events in an event stream had timestamps that differed exactly by one millisecond from each other. Therefore, all time windows of the parametrized test queries (see Listing 1) were filled with 500 events on average. Before each measurement, we performed a warm-up. During this phase, all time windows were filled up completely. We executed the following sets q_{Count} of test queries three times and have reported the average input event throughput:

$$\forall q \in \{\sigma, \bowtie, \alpha, \rho\} : \forall Count \in \{1, \ldots, 100\} : q_{Count} = \{q(i, Count) \mid 1 \leq i \leq Count\}$$

Independent of specific operator types, the throughput decreases rapidly at the beginning and becomes relative stable afterwards. An extract of the precise results is discussed in the following: 80 parallel running filter operators processed 219 events per second, 80 parallel running pattern matching operators processed 114 events per second, 80 parallel running aggregation operators processed 49 events per second and 80 parallel running correlation operators processed 18 events per second. Aggregations are expensive, because they are computed entirely from the scratch and not incremental like in real EP systems. Performance optimizations like incremental computation of aggregates are generally possible and will be addressed in our future work. We also tested a mixed workload consisting of 20 filter, 20 aggregation, 20 correlation and 20 pattern matching operators and measured an average event throughput of 100 events per second. This performance is sufficient for many EP tasks as well as for the development and testing of EP applications. We executed also all tests with the database on a solid state drive and in-memory, but the performance improvements were only in the range of 1 to 10 percent, so we do not report them in detail.

Our implementation behaves like most special-purpose EP systems; for each single input event all related queries are triggered (tuple-driven behavior). We would achieve better performance if queries are only triggered after some time has elapsed or a predefined number of input events have been pushed (batch-driven behavior).

6 Conclusion

We have motivated and presented the implementation of an *event processing* (EP) system purely on top of JDBC. This kind of implementation enables every standard database system to provide EP functionality in Java applications. To support small and medium-sized EP workloads, a database system that in most applications already exists can be (re-)used with only low costs. The use of a database system instead of a special-purpose EP system makes also sense for the development and testing of EP applications. Because JDBC abstracts from specific database products, they can be easily exchanged without affecting the implementation of the EP system on top of JDBC.

Acknowledgments This work has been supported by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) under grant no. 01BY1206A.

References

[A+12] H. Appelrath et al. Odysseus: a highly customizable framework for creating efficient event stream management systems. In *DEBS*, pages 367–368, 2012.

[ABW06] A. Arasu, S. Babu and J. Widom. The CQL containuous query language: semantic foundations and query execution. In *VLDB Journal*, 15(2), pages 121–142, 2006.

[BLT86] J. Blakeley, P. Larson and F. Tompa. Efficiently updating materialized views. In *SIGMOD*, pages 61–71, 1986.

[DR04] L. Ding and E. Rundensteiner. Evaluating window joins over punctuated streams. In *CIKM*, pages 98–107, 2004.

[esp13] Esper. `http://esper.codehaus.org/` (2013-01-17).

[G+09] T. Grabs et al. Introducing Microsoft StreamInsight. Technical report, 2009.

[h2d13] H2 Database. `http://www.h2database.com/` (2013-01-17).

[KS09] J. Krämer and B. Seeger. Semantics and implementation of continuous sliding window queries over data streams. In *TODS*, 34(1), pages 4:1–4:49, 2009.

[Luc01] D. Luckham. The power of events: an introduction to complex event processing in distributed enterprise systems. Addison-Wesley Longman Publishing, 2001.

[Mei11] E. Meijer. The world according to LINQ. In *Commun. of ACM*, 54(10), pages 45-51, 2011.

[ora13] Oracle CEP. `http://www.oracle.com/technetwork/middleware/complex-event-processing/overview/index.html` (2013-01-17).

[TM03] P. Tucker and D. Maier. Dealing with disorder. In *MPDS*, 2003.

[WDR06] E. Wu, Y. Diao and S. Rizvi. High-performance complex event processing over streams. In *SIGMOD*, pages 407–418, 2006.

[web13] webMethods Business Events. `http://www.softwareag.com/corporate/products/wm/events/` (2013-01-17).

Das dynamische Mindesthaltbarkeitsdatum – Auf dem Weg zu einer Echtzeitereignisverarbeitung in der Lebensmittellogistik

[1]Falk Alexander, [1]Thomas Hipp, [1]Tobias Scholze,
[1]Felix Wagner, [1]Thorsten Schöler, [2]Michael Krupp

[1]Fakultät für Informatik, [2]Fakultät für Wirtschaft
Hochschule für Angewandte Wissenschaften Augsburg
An der Hochschule 1
86161 Augsburg

{falk.alexander, thomas.hipp1, tobias.scholze,
felix.wagner, thorsten.schoeler, michael.krupp}@hs-augsburg.de

Abstract: Das dynamische Mindesthaltbarkeitsdatum stellt neue Herausforderungen an logistische Prozesse und deren IT-Unterstützung. Es wird gezeigt, dass die Ereignisverarbeitung in Echtzeit eine wertvolle Unterstützung in Logistik-IT-Systemen darstellt. Eine Softwarearchitektur zur Überwachung von Logistikprozessen wird vorgestellt und im Anwendungsfall Dynamisches Mindesthaltbarkeitsdatum angewandt.

1 Einleitung

Ein aktuell vieldiskutiertes Problem im Lebensmitteleinzelhandel, ist das massenhafte Wegwerfen von Nahrungsmitteln. Nur ein Bruchteil der erzeugten Lebensmittel landet wirklich auf dem Teller der Konsumenten. In Deutschland werden pro Jahr elf Millionen Tonnen Lebensmittel entlang der Versorgungskette weggeworfen [Ins12]. Dies beginnt bereits beim Erzeuger, mit Produkten, die der Norm und den Anforderungen der Verbraucher nicht standhalten, setzt sich fort in der logistischen Kette zum Handel, wo z. B. Druckstellen zu Aussortierung führen und endet im Kühlschrank der Verbraucher, wo Produkte oft grundlos rigoros entsorgt werden.

Bei Lebensmitteln und verderblichen Produkten, spielt in diesem Zusammenhang das Mindesthaltbarkeitsdatum (MHD) eine wesentliche Rolle. Jedes Lebensmittel trägt dieses Datum, das oftmals ein Richtwert für den Verbraucher ist, ab wann er Lebensmittel wegwirft [Bun12]. Für den Einzelhandel ist es wesentlich, nur Produkte im Regal anzubieten, die noch ausreichend Zeit bis zum Ereichen des MHD haben. Jeder kennt den Reflex, die „jüngere“ und vermeintlich bessere Milch von hinten aus dem Regal zu nehmen, um diese dann doch am gleichen Tag zu trinken.

Das Management der Güterbewegungen berücksichtigt heute bereits das MHD, so ist in

Großhandelslagern bekannt, welche Produkte welches MHD haben. Üblicherweise werden die „älteren“ Produkte zuerst ausgelagert und in den Verkauf gebracht. Allerdings ist nicht bei allen Produkten das MHD wirklich aussagekräftig für die Qualität und die verbleibende Zeit bis zum Verfall. So ist bei verderblichen Produkten, die oft nicht mit Datum gekennzeichnet sind, wie Obst und Gemüse, die Optik für den Verbraucher wichtiger als die angegebene Daten [Ins12]. Hinzu kommen objektive Qualitätskriterien wie bereits genannte Druckstellen oder Geruch. Je nach Reifegrad einer Frucht bleibt bis zum Verkauf einer Ware ein gewisses Zeitfenster. Ist dieses überschritten, kann die Ware nach heutigen Kundenanforderungen nicht mehr verkauft oder nur vergünstigt verkauft werden. Die Länge dieses Zeitfensters ist abhängig von der korrekten Behandlung der Früchte im logistischen Prozess. Das Schlüsselwort heißt hier Kühlung: Wird eine durchgängige Kühlung bei der richtigen Temperatur erreicht, dann spricht der Logistiker von der geschlossenen Kühlkette. Erschwert wird dies durch unterschiedliche Temperatur-Anforderungen. So benötigen Schokolade ca. 15 °C, Äpfel ca. 4 °C und Südfrüchte ca. 12 °C. In Lagerhallen werden meist nicht ausreichend unterschiedliche Temperaturzonen eingerichtet. Resultat ist, dass nicht alle Produkte bei optimaler Temperatur gelagert werden können. Da lautet die Devise: Geschwindigkeit, also Produkte möglichst schnell zu distributieren und möglichst kurz suboptimal zu kühlen.

Die Verfolgung der Kühlung mittels RFID wird seit langem diskutiert. RFID-Transponder werden hierzu mit Thermometern kombiniert. Jederzeit kann ausgelesen werden, welcher Temperatur das jeweilige Produkt entlang der logistischen Kette ausgesetzt war. Das Konzept des dynamischen MHD geht einen Schritt weiter: Ausgehend von der maximalen Haltbarkeit bei optimaler Kühlung, wird die Haltbarkeit reduziert, wenn in der logistischen Kette ungünstige Verhältnisse auf das Produkt eingewirkt haben. Das heißt, Temperaturüberschreitungen werden identifiziert, ausgewertet und in der Berechnung der restlichen Lebenszeit des Produktes berücksichtigt. Dies führt dazu, dass eventuell Palette A, die zwar später in ein Lager eingelagert wurde als Palette B, aber ungünstigere Rahmenbedingungen ausgesetzt war, als erste in die Verkaufsräume geht, oder gleich als B-Ware verkauft wird, um die Entsorgung zu vermeiden. Bei der aktuellen Entwicklung der Sensoren ist dieses Konzept auch in Kombination mit anderen Verfallsindikatoren, wie Ethylengasen (Reifegase bei Fruchtreife) oder auch Lichteinfall o. ä., denkbar [Fra12].

Die weiteren Abschnitte sind wie folgt gegliedert: Zunächst wird im Abschnitt 2 der Stand der Technik in der IT-Unterstützung von Lebensmittellogistikketten, speziell unter dem Gesichtspunkt RFID und Ereignisverarbeitung in Echtzeit, kurz dargestellt. Anschließend werden in Abschnitt 3 einige Anwendungsfälle präsentiert, welche in Abschnitt 4 im Rahmen eines Hardware-/Software-Co-Systems umgesetzt werden. Abschließend werden die mit der Umsetzung gesammelten Erfahrungen dargestellt (Abschnitt 5) und in Abschnitt 6 eine Zusammenfassung mit Ausblick auf nachfolgende Forschungsaktivitäten gegeben.

2 Stand der Technik aus IT-Sicht

Der Einsatz von Ereignisverarbeitungstechnologien, wie z. B. Complex Event Processing (CEP), in Logistikanwendungen und dem Internet der Dinge ist nicht neu, wie beispiels-

weise in [SSMW10] beschrieben. Ebenso wenig wie der Einsatz von RFID in der Lebensmittellogistik bereits gut untersucht ist (siehe auch [GSC+07] und [GSBB08]). Allerdings erscheint der in diesem Beitrag vorgestellte Ansatz, RFID mit CEP zu einem Echtzeitsystem für die Lebensmittellogistik zu kombinieren, erfolgversprechend, wie es die Studie ähnlich gelagerter Untersuchungen zeigt.

Einen Überblick über den Einsatz von RFID-Technologie in der Lebensmittel-Supply-Chain gibt [Jon06]. Am Beispiel der Fleischproduktion in Japan wird der Einsatz von RFID deutlich und der Informationsfluss über die einzelnen Teilnehmer der Logistikkette hinweg verdeutlicht. [GMRS07] beschreibt ebenfalls einen RFID-basierten Ansatz für die Lebensmittellogistik (in diesem Fall Früchte), allerdings mehr auf Funk- und Hardwareaspekte fokussierend.

In [HCC06] wird eine Softwarearchitektur beschrieben die dem hier vorgestellten Ansatz ähnlich ist. Es werden Informationen von RFID-Tags mittels Datenfusion ausgewertet. Im Gegensatz zum hier beschriebenen Ansatz wird ein regelbasiertes System mit Fuzzyfizierung verwendet, ein Ansatz der dem Fuzzy Classifier System unter [SMS05] nicht unähnlich ist.

Der Begriff eines „Real-Time Food Receiving operations management Systems (RFRS)“ aus [LCTP10] beschreibt den eingangs vorgestellten Anwendungsfall. Im Gegensatz zu den in [LCTP10] eingesetzten Case-Base-Reasoning- und Methoden der Fuzzy-Logik [HCC06], sowie dem Vorschlag aus [LCTP10], verwendet der hier im Anschluss vorgestellte Ansatz Standard-CEP-Komponenten zur Ereignisverarbeitung und steht somit den Untersuchungen in [HHX09] näher. Die beschreibungslogische Modellierung der Ereignisse in einer Ontologie, wie in [HHX09] vorgeschlagen, soll in Zukunft ebenfalls untersucht werden, wie es im Abschnitt 6 kurz beschrieben wird.

Um das Szenario Dynamisches MHD als Anwendungsfall in der IT-Infrastruktur eines Unternehmens umsetzen zu können, sind Softwarearchitekturen notwendig, die schnell und dynamisch – am besten in Echtzeit – auf aktuelle Gegebenheiten im Logistikprozess reagieren können. Bei der Lebensmittellogistik darf eben nicht erst in einem nachts laufenden Batch-Prozess festgestellt werden, dass auf Palette A schnell verderblicher Fisch gelagert ist, der am Vortag noch verkauft hätte werden können.

Um zu einer tragfähigen IT-Architektur zu gelangen, sollen nun einige exemplarische Anwendungsfälle analysiert werden.

3 Anwendungsfälle

Aus dem Logistik-Anwendungsfall Dynamisches MHD lassen sich Anwendungsfälle (Use cases) ableiten, aus denen wiederum die Anforderungen an eine Softwarearchitektur zur Logistikunterstützung abgeleitet werden können. Im Folgenden sollen exemplarisch Anwendungsfälle vorgestellt werden, die direkten Einfluss auf die in Abschnitt 4 vorgestellte Softwarearchitektur haben.

Warenfluss im Umschlagsplatz: Waren, bestehend aus beispielsweise verschiedenen Früchten und Gemüse, erreichen einen Umschlagplatz. Beim Einlagern der Waren werden diese auf Vollständigkeit geprüft. Dies geschieht beispielsweise durch einen Abgleich der Liste der ankommenden EPCs[1] und der vorher eingereichten Liste durch den Erntebetrieb. Zusätzlich werden die Messdaten über den bisherigen Kühlstatus ausgelesen. Ist die Kühlung nicht durchgehend optimal gewesen, so verringert sich automatisch das MHD. Die Waren müssen früher ausgelagert werden, damit sie beim Endverbraucher das gleiche MHD besitzen. Die Transportwege müssen kurzfristig adaptiv sein. Automatisierte Transportsysteme, z. B. Rollbänder, holen bevorzugt Waren mit geringer MHD. Personell besetzte Lieferanten wie Gabelstaplerfahrer nutzen mobile Endgeräte z. B. Tablets statt Klemmbretter. Deren Transportlisten werden durch ereignisorientierte Systeme aktualisiert.

Dynamische Warenlisten: Der Güterverkehr zwischen den Lagerplätzen ist vielfältig. Jedoch werden Waren immer in Sendungen gruppiert. Basierend auf der verbleibenden Haltbarkeit, können diese optimiert zusammengestellt werden. Die Geschwindigkeit der Transportwege unterscheidet sich. So können Waren mit einem geringen MHD einen schnelleren Transportweg bekommen. Sind Informationen über den Zustand einer Lieferung bekannt, während sie unterwegs ist, so können Verzögerungen kompensiert werden. Beispiel: Die Tomaten aus der Lieferung A erreichen nicht rechtzeitig eine Anschlusslieferung B. Durch dieses Ereignis lassen sich Transportlisten schnell anpassen um Lieferkapazitäten bestmöglich auszunutzen.

Historie: Im System wird eine Historie aller eingegangenen Waren protokolliert. Der Warenfluss ist sowohl für Unternehmer als auch Verbraucher transparent. Fehlerfälle können leichter identifiziert werden. Fehler können im Transportwesen entstehen: Eine Kiste Früchte wurde versehentlich zum Gemüse gestellt und bleibt vorerst unbemerkt, da die Kühlung annähernd ähnlich ist. Wird jedoch die vermeintliche Gemüsekiste zu einer Sendung Früchte gestellt, schlägt das System Alarm. Die Kombination von Früchten in diesem Gemüsetransport ist ein Fehlerfall. Die Früchte sind nicht auf der Sendungsliste eingetragen. Dieser Fehlerfall kann schon beim Verlassen des Kühlraums erkannt werden. Der Abholer hat keine Berechtigung die Früchte aus dem Kühlraum zu entnehmen. Basierend auf den Anforderungen aus den dargestellten Anwendungsfällen (Use Cases) wurde die folgende Hard-/Software-Architektur prototypisch umgesetzt.

4 Technischer Aufbau

Der Aufbau in Abbildung 1 lässt sich, analog der in [BD10] vorgestellten Systeme, in drei Schichten unterteilen, wie in den folgenden Abschnitten dargestellt.

[1] Elektronische Produktcode, international verwendete Identifikationsnummer für Produkte

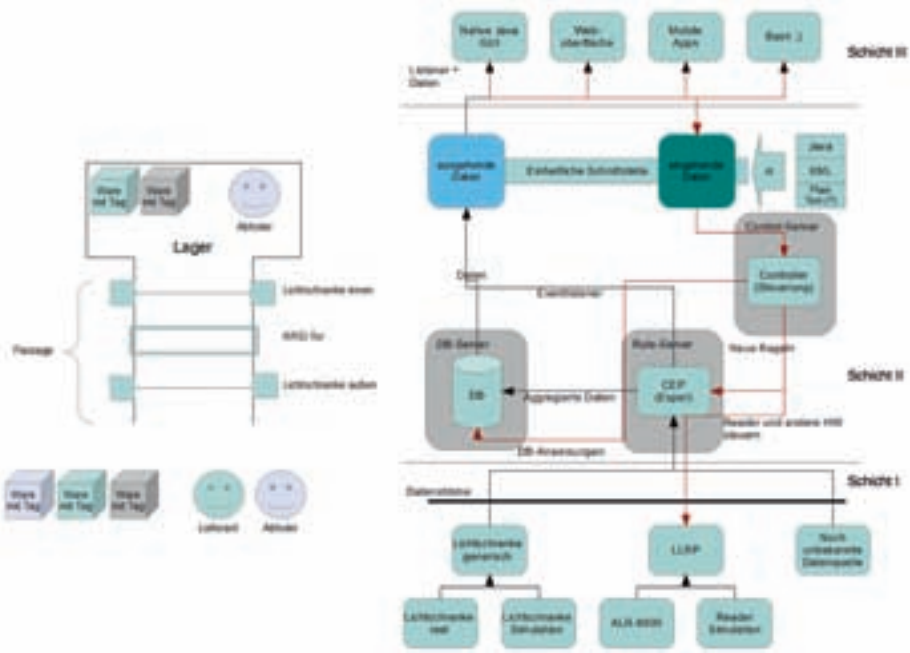


Abbildung 1: Softwarearchitektur der Ereignisverarbeitung

4.1 Hardwareschicht (Schicht I)

In dieser Schicht werden alle Sensoren verwaltet, die im Gesamtsystem Daten generieren. Jeder einzelne Sensor übermittelt seine Messungen, dies können von Temperaturmessungen bis hin zur Position eines Objekts alle denkbaren Daten sein, die für ein Objekt von Bedeutung sind. Ein Sensor ist nicht zwangsläufig ein klassischer Sensor für Temperatur, Luftdruck oder Feuchtigkeit, sondern kann auch ein RFID-Leser, Überwacher einer Datenbank, QR- oder Barcodescanner sein. Manuelle Datenerfassungen im logistischen Prozess werden ebenfalls als Sensoreingaben betrachtet. Die gesammelten Daten werden mit einem Messaging-Dienst an die Schicht II weitergegeben. Dieser Messaging-Dienst erlaubt es den Sensoren, vollständig verteilt zu sein. Weiterhin übernimmt er die Sicherung des Kommunikationsverkehrs und die Bewältigung der Datenmengen. Zum Einsatz kam das Messaging-System HornetQ[2] aus dem Open-Source-JBoss-Projekt. Dieses Messaging-System übernimmt die Nachrichtenzustellung zwischen allen Schichten und Komponenten der Softwarearchitektur. In der Hardwareschicht liegt das Hauptaugenmerk auf Verlässlichkeit und Durchsatz der Ereignisverarbeitung.

Konkret verwendet das System einen RFID-Multi-Antennen-Reader von Alien Technology (ALR-9900), welcher sogenanntes Bulk Reading, also das gleichzeitige Erfassen einer großen Menge RFID-Tags, wie sie üblicherweise bei der Erfassung logistischer Gebinde auf Paletten verwendet werden, unterstützt. Als weitere Informationsquelle im System (Datenfusion) wird eine Lichtschranke verwendet, durch welche die Bewegungsrichtung der Palette bei einer RFID-Tor-Durchfahrt erkannt werden kann (Ein- oder Auslagerung).

[2] http://hornetq.org/

Abbildung 2: Entwurf einer leichtverständlichen Ansicht und Rich-Client-Gui mit Java Swing

4.2 Verarbeitungsschicht (Schicht II)

Um aus der reinen Datenflut auf niedriger Ereignisebene Informationen zu extrahieren, werden eingehende Nachrichten der Sensoren an die integrierte CEP-Engine übertragen. Für die Verarbeitung der Ereignisse (z. B. Datenfusion und -aggregation) in der CEP-Engine wurden Regeln deklariert, die den verschiedenen logistischen Anwendungsfällen entsprechen. Die reinen Daten (Ereignisse auf niederster Ebene) sowie das Zutreffen/-Feuern von Regeln wird in der Datenbank persistiert. Über diesen Weg können – CEP-typisch – aus niederen einzelnen Ereignissen, höherwertige Informationen (komplexe Ereignisse nach Luckham [Luc07]) abgeleitet werden.

Zur Anbindung an die verwendeten In-Memory-Datenbank wurde der von JBoss entwickelte OR-Mapper Hibernate[3] verwendet. Ein OR-Mapper, kurz für Object-Relational-Mapper, ermöglicht die Abbildung von Datenstrukturen auf POJOs[4]. Somit ist eine Datenbankinteraktion aus Entwicklersicht ein Erstellen und Bearbeiten von Objekten. Lasttests der Persistenzarchitektur ergaben, dass die Persistierung von 10 000 Datensätzen zu je fünf atomaren Spalten mittels dieses Ansatzes auf einem Standard-PC weniger als zehn Sekunden dauerte.

4.3 Anzeigeschicht (Schicht III)

Die aus Schicht II abgeleiteten Informationen werden auf lesbare Art bereitgestellt (siehe Abbildung 2). Die einfachste Möglichkeit ist das Anzeigen des aktuellen Zustandes eines Logistikprozesses oder eines Objekts (beispielsweise einer Palette mit frischen Früchten) z. B. bezüglich Ort, Ziel und Prozessabschnitt. Dem Benutzer kann eine Meldung zugestellt werden, die sein Eingreifen erfordert oder ihn auf eine mögliche Fehlfunktion hinweist. Mögliche Meldungen sind u. a. Pop-Up-Fenster an seinem Arbeitsplatzrechner, Nachrichten auf sein Smartphone oder mobiles Gerät bzw. Warnsignale z. B. das Auslösen einer Warnleuchte mit Sirene.

Konkret wurden verschiedene gleichberechtigten Benutzeroberflächen umgesetzt und mit-

[3] http://www.hibernate.org
[4] Plain Old Java Objects

Abbildung 3: Ansicht auf mobilen Endgeräten

tels Messaging-System an die Verarbeitungsschicht gekoppelt: Eine native Rich-Client-GUI auf Basis von Java Swing (siehe Abbildung 2), sowie eine webbasierte Oberfläche welche mittels *Responsive Design* das enthaltene Layout je nach Größe des Bildschirms optimal anpasst[5]. Somit ist diese Web-Oberfläche von Desktop-Arbeitsplätzen sowie auch von mobilen Endgeräten wie Tablets oder Smartphones verwendbar, wie es Abbildung 3 zeigt. Beide Benutzeroberflächen ermöglichen die Ausgabe der aktuellen Ereignissituation sowie die Konfiguration und das Management des Ereignisverarbeitungssystems. Dies bedeutet, dass mehrere Clients gleichzeitig in der Lage sind, sich auf den Datenstrommanagementserver zu verbinden und dessen Informationen grafisch darzustellen. Als Anwendungsfall sind verteilte und einfach gehaltenen Kontrollbildschirmansichten für Lagerarbeiter und – parallel dazu – eine Anzeige einer komplexen Oberfläche für den Vorarbeiter denkbar.

5 Anwendung und Erfahrungen

Der skizzierte Hardware/Software-Aufbau ermöglicht die Umsetzung des in Abschnitt 1 beschriebenen Szenarios Dynamisches MHD. Waren können in das angeschlossene IT-System ein- und ausgelagert werden und die hierbei erzeugten Ereignisströme können in Echtzeit analysiert, überwacht und gemanagt werden. Die erstellte Infrastruktur ist flexibel und leicht an weitere logistische Anwendungen anpassbar.

Um nun sicherzustellen, dass sich im Lager keine Waren mit kritischem MHD befinden, müssen alle Prozesse, die den Warenfluss betreffen, überwacht werden. Durch diese Überwachung werden Prozesse transparenter und können analysiert werden. Jeder Schritt, den eine Ware im Prozess durchläuft, muss protokolliert werden. Dadurch werden die Daten für die Analyse gewonnen.

Durch die Aggregation der Ereignisdaten des Warenflusses mit weiteren Daten werden zusätzliche Informationen gewonnen. Lädt beispielsweise ein LKW Waren ab, so ist im

[5]Die Weboberfläche arbeitete im Hintergrund mit Tapestry (http://tapestry.apache.org/). Als Hilfe zur Erstellung der Oberfläche wurde das von Twitter stammende Bootstrap-CSS-Framework (http://twitter.github.com/bootstrap/) verwendet.

Prozess die Information bekannt, dass die Waren im System existieren. Der LKW identifiziert sich beim System und somit kann verifiziert werden, dass die Waren durch den korrekten Dienstleister, an der passenden Verladestelle zur gewünschten Zeit unter Berücksichtigung des MHDs angeliefert wurden. Das Datenaufkommen lässt sich auf Grund der gut skalierenden Architektur nahezu beliebig erhöhen.

Produzierende oder weiterverarbeitende Betriebe fügen den Waren Informationen über ihre weiteren Verarbeitungsprozesse hinzu wie es beispielsweise bei der Just-In-Sequence-Produktion der Fall ist. Komplexere Produkte besitzen eine Vielzahl von Daten, bestimmt durch die Komponenten, aus denen sie zusammengesetzt sind. Diese zusätzlichen Informationen können ebenfalls auf beschreibbaren RFID-Tags untergebracht werden. Die in Echtzeit erfassten Primärereignisse werden vorverarbeitet, aggregiert und mit weiteren Daten angereichert. Es entstehen, online und ebenfalls in Echtzeit, komplexe Ereignisse die tiefgreifendere Rückschlüsse auf den Warenfluss erst ermöglichen.

Ein Ereignis wird aus dem Zusammenspiel verschiedener Informationen erzeugt. An einem Warenumschlagplatz trifft eine Lieferung mit verschiedenen Früchten ein. Alle Früchte wurden auf ihrem Transportweg einheitlich gekühlt. Dadurch ist die optimale Lagertemperatur, von beispielsweise Südfrüchten, nicht eingehalten worden. Ihr MHD ist somit reduziert. Diese Information löst eine Benachrichtigung an das Smartphone eines Mitarbeiters aus, dass die Früchte schneller in eine geeignete Kühlzone gebracht werden müssen. Im zukünftigen Warenfluss werden diese mit einer höheren Priorität versendet. Die kontinuierliche Echtzeit-Überwachung des MHD ermöglicht Rückschlüsse auf die Güte der Transportkette. Wurden Waren in der Vergangenheit falsch verladen oder unzureichend auf ihrem Transportweg gekühlt, kann dies automatisch als Ereignis aus dem Gesamtsystem hervorgehen. Jedem Akteur in diesem System werden andere Informationen bereitgestellt. Einen Lagerist betreffen kurzfristige Planänderungen. In der administrativen Ebene ist es wichtiger zu wissen, dass der Einzelhandel benachrichtigt werden muss, dass Südfrüchte mit verringertem MHD eintreffen.

Für die gegebenen Anwendungsfälle der Logistik hat sich eine Umsetzung auf Basis von CEP angeboten. Nicht zuletzt die voneinander unabhängigen Datenquellen RFID-Reader und Lichtschranke machten eine Datenfusion (Verknüpfung der Ereignisse) nötig. Auch der hohen Frequenz der Ereignisse musste mit einer entsprechenden Softwarearchitektur Rechnung getragen werden. Nach einer Evaluation verschiedener CEP-Engines (z. B. JBoss Drools und Esper) und der Gegenüberstellung mit den Anforderungen an eine solche Komponente, kam Esper zu Einsatz. Nicht zuletzt, da Esper einen einfacheren Einstieg sowie eine umfassende Dokumentation bietet.

Um die Einfachheit der Beschreibung von Überwachungsregeln mit Esper zu belegen, sollen im folgenden einige Regelfragmente exemplarisch für das System vorgestellt werden.

```
SELECT * FROM TagEvent
WHERE (id == '1234 5678' AND temp > 3.0 AND temp < 5.0)
OR (id == '2468 3579' AND temp > 11.0 AND temp < 13.0)
```

Obige Regel informiert darüber, ob die jeweiligen RFID-Tags ihren gültigen Temperaturbereich über- bzw. unterschreiten. Für die skizzierten Anwendungsfälle, war es zusätzlich notwendig, kausale Abfolgen von Ereignissen zu definieren. Dies ließ sich mit den ent-

sprechenden Ausdrucksmöglichkeiten der Esper-EPL gut umsetzen.

```
SELECT * FROM pattern [every SensorEvent(triggered, name='A') ->
    SensorEvent(NOT triggered, name='A') -> middleevent=TagEvent
    until SensorEvent(triggered, name='B') -> SensorEvent(NOT
    triggered, name='B')]
```

Erfasst werden alle RFID-Tags zwischen dem Ereignis *A*, dem Überschreiten der Kühltemperatur und dem abschließendem Ereignis *B*, dem Erreichen der optimalen Temperatur.

6 Fazit und Ausblick

Der Anwendungsfall Dynamisches MHD zeigt aktuelle Herausforderungen in lebensmittellogistischen Prozessen. Durch die vorgestellte Softwarearchitektur und den Einsatz von CEP steht eine leistungsfähige Möglichkeit zur Verarbeitung der anfallenden Daten zur Verfügung, die auch mit hohen Datenvolumen zurecht kommt, wie sie beispielsweise beim sogenannten Bulk Reading von ganzen Paletten RFID-ertüchtigter Waren die Regel sind. Die moderne multimodale Responsive-Design-Web-Oberfläche ermöglicht eine übersichtliche Situationsbewertung sowie ein transparentes Management des gesamten IT-Systems.

Zukünftige Arbeiten auf Basis der vorgestellten Hardware und Software umfassen eine Verteilung der Systemfunktionalitäten auf Softwareagenten des objektfunktionalen SCALA-Multiagentensystem SMAS[6] sowie eine 3D-Visualisierung im parallel dazu entwickeltem 3D-SCADA-System für industrielle Anwendungen [Bor12]. Durch die bereits umgesetzten SMAS-Agenten-Plug-Ins für semantische Datenverarbeitung auf Basis von OWL-Ontologien, soll die Modellierung der logistischen Prozesse sowie die Überwachung dieser Prozesse auf Basis von Beschreibungslogik umgesetzt werden. Die Umsetzung des Daten- und Ereignismodells einer industriellen Fertigungsanlage stellen eine sinnvolle Ausgangsbasis für die beschreibungslogische Modellierung von Logistikprozessen und deren Ereignissen dar.

Danksagung Wir bedanken uns bei dem Team der Abteilung für Supply Chain Technologies des Fraunhofer-Instituts für Integrierte Schaltungen IIS, welches uns ihm Rahmen des dieser Arbeit zugrundeliegenden Studienprojektes tatkräftig mit der notwendigen technischen Ausstattung unterstützt hat und uns auch über das Projekt hinaus bei der Erstellung dieser Arbeit beratend zur Seite stand.

Literatur

[BD10] Ralf Bruns und Jürgen Dunkel. *Event-Driven Architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse*. Springer, Berlin, 1. Auflage, Mai 2010.

[6] https://github.com/scala-multi-agent-system sowie [Lie12]

[Bor12] Benjamin Bortfeldt. *Verarbeitung und Visualisierung von Ereignissen für Industrieanlagen*. Masterarbeit, Hochschule für Angewandte Wissenschaften Augsburg, Augsburg, (noch nicht erschienen), 2012.

[Bun12] Bundesministerium für Ernährung, Landwirtschaft und Verbraucherschutz. Das Mindesthaltbarkeitsdatum ist kein Verfallsdatum, 2012.

[Fra12] Fraunhofer-Gesellschaft, Food Chain Management. Gassensorik für die Food Chain. http://www.fcm.fraunhofer.de/de/beispiele1/kompakte_messsystemefuerlebensmittelschnelltests.html, 2012.

[GMRS07] F. Gandino, B. Montrucchio, M. Rebaudengo und E.R. Sanchez. Analysis of an RFID-based Information System for Tracking and Tracing in an Agri-Food chain. In *RFID Eurasia, 2007 1st Annual*, Seiten 1 –6, sept. 2007.

[GSBB08] Pablo Ezequiel Guerrero, Kai Sachs, Stephan Butterweck und Alejandro P. Buchmann. *Performance Evaluation of Embedded ECA Rule Engines: A Case Study*. 2008.

[GSC+07] Pablo Ezequiel Guerrero, Kai Sachs, Mariano Cilia, Christof Bornhövd und Alejandro P. Buchmann. Pushing Business Data Processing Towards the Periphery. In Rada Chirkova, Asuman Dogac, M. Tamer Özsu und Timos K. Sellis, Hrsg., *ICDE*, Seiten 1485–1486. IEEE, 2007.

[HCC06] Han-Pang Huang, Ching-Shun Chen und Tien-Ying Chen. Mobile Diagnosis based on RFID for Food Safety. In *Automation Science and Engineering, 2006. CASE '06. IEEE International Conference on*, Seiten 357 –362, oct. 2006.

[HHX09] Zhu Huaji, Wu Huarui und Sun Xiang. Research on the Ontology-Based Complex Event Processing Engine of RFID Technology for Agricultural Products. In *Artificial Intelligence and Computational Intelligence, 2009. AICI '09. International Conference on*, Jgg. 1, Seiten 328 –333, nov. 2009.

[Ins12] Institut für Siedlungswasserbau, Wassergüte- und Abfallwirtschaft. Ermittlung der weggeworfenen Lebensmittelmengen und Vorschläge zur Verminderung der Wegwerfrate bei Lebensmitteln in Deutschland – Kurzfassung. http://www.bmelv.de/SharedDocs/Downloads/Ernaehrung/WvL/Studie_Lebensmittelabfaelle_Kurzfassung.html, 2012.

[Jon06] P. Jones. Networked RFID for use in the Food Chain. In *Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on*, Seiten 1119 –1124, sept. 2006.

[LCTP10] S.I. Lao, K.L. Choy, Y.C. Tsim und T.C. Poon. A RFID-based decision support system for food receiving operations assignment. In *Supply Chain Management and Information Systems (SCMIS), 2010 8th International Conference on*, Seiten 1 –7, oct. 2010.

[Lie12] Rico Lieback. *Objekt-funktionale Plattform für Cyber-Physical Systems*. Bachelorarbeit, Hochschule für Angewandte Wissenschaften Augsburg, Augsburg, 2012.

[Luc07] David Luckham. *The power of events: an introduction to complex event processing in distributed enterprise systems*. Addison Wesley, 2007.

[SMS05] Thorsten Schöler und Christian Müller-Schloer. First steps towards organic computing systems: monitoring an adaptive protocol stack with a fuzzy classifier system. In *Proceedings of the 2nd conference on Computing frontiers*, CF '05, Seiten 10–20, New York, NY, USA, 2005. ACM.

[SSMW10] C. Seel, J. Schimmelpfennig, D. Mayer und P. Walter. Conceptual modeling of complex events of the Internet of Things. In *eChallenges, 2010*, Seiten 1 –8, oct. 2010.

Temporale Konsistenz von Kontextinformationen in Datenstromsystemen

Dennis Geesen, Marco Grawunder, H.-Jürgen Appelrath

Abteilung Informationssysteme
Department für Informatik
Universität Oldenburg
{dennis.geesen, marco.grawunder, appelrath}@uni-oldenburg.de

Abstract: Bei Entscheidungen in ereignisorientierten Systemen, wie bspw. in Smart Homes, wird neben kontinuierlichen und sich schnell ändernden Ereignisdatenströmen häufig auch auf Kontextinformationen zurückgegriffen, die den Zustand bestimmter Dinge wie z.B. Türen oder Lampen charakterisieren. Da sich diese im Gegensatz zu Ereignisdatenströmen sehr selten ändern, wird der zeitliche Zusammenhang zwischen Kontextinformationen und Ereignissen meistens vernachlässigt. In einigen Systemen ist jedoch diese temporale Konsistenz wichtig. In diesem Beitrag werden zwei Konzepte betrachtet, um solche Kontextinformationen zu verwalten und Ereignisdatenströmen damit anzureichern. Der geläufige Ansatz mit Fenstern bietet zwar temporale Konsistenz, ist jedoch nachteilig bei sich selten ändernden Kontextinformationen, da er das System blockiert. Wir zeigen daher alternativ einen Ansatz, der zwar temporale Konsistenz nicht stets zusichern kann, aber dafür nicht blockierend ist.

1 Einleitung

Smart Homes verwenden Sensoren, um Zustände und Bewegungen von statischen und beweglichen Objekten zu erkennen. Solche Sensordaten werden dann von verschiedenen Anwendungen genutzt, um bestimmte Situationen zu erkennen und daraufhin eine passende Aktion auszuführen. Ein einfaches Beispiel ist eine Anwendung, die geschlossene Türen öffnet, wenn sich jemand davor bewegt. Dazu greift diese Anwendung zum einen auf den Zustand der Tür und zum anderen auf die aktuellen Sensordaten von Bewegungssensoren zurück. Im Gegensatz zu den Bewegungsdaten, die kontinuierlich als Datenstrom ausgewertet werden, ändert sich der Zustand der Tür nur selten. Dabei hebt jeder nachfolgende Zustand den aktuell gültigen Zustand auf. Die Anwendung benutzt den aktuellen Zustand als sogenannte Kontextinformation und reichert damit den Bewegungsdatenstrom an. Der daraus entstehende Datenstrom enthält dann sowohl die Information über die Bewegung und den Zustand der Tür und kann anschließend als Entscheidung dienen, ob die Aktorik der Tür diese öffnen soll. Hierbei muss jedoch darauf geachtet werden, dass der zeitliche Kontext korrekt ist und der verwendete Türzustand auch während der Bewegung gültig gewesen ist. Ein Ansatz, um den zeitlichen Kontext zu berücksichtigen, ist die Verwendung von Gültigkeitsintervallen (vgl. [EN11, EMR+10, Kra07, ACc+03] u.a.).

Dabei ist jedes Ereignis für ein bestimmtes Zeitintervall gültig und zwei Ereignisse, also bspw. eine Bewegung und ein Türzustand, fanden zur selben Zeit statt, wenn sich deren Gültigkeitsintervalle überschneiden. Das Gültigkeitsintervall einer Bewegung ergibt sich z.B. durch die Dauer der Bewegung oder durch die Periodendauer der Sensormessungen. Bei der Tür, dessen Zustand sich diskret ändert, ist das Gültigkeitsintervall abhängig vom nachfolgenden Ereignis. Ist also eine Tür geschlossen worden, blockiert das System bis die Tür wieder geöffnet wurde, um damit das Gültigkeitsende für das Schließen bestimmen zu können. Da der Zustand der Tür im System also asynchron zum wirklichen Zustand ist, würde die Entscheidung zum Öffnen zu spät getroffen werden und auch die Aktorik zum Öffnen der Tür stets zu spät angesprochen werden. Die zuvor genannte Anwendung könnte dann die Tür jedoch nicht rechtzeitig öffnen. Als Alternative zu diesem blockierenden Verhalten, wäre ein optimistischer Ansatz möglich, indem man annimmt, dass der verwendete Zustand noch gültig ist. Hierbei können wiederum inkonsistente Anreicherungen entstehen. In diesem Beitrag untersuchen wir daher solche zeitlichen Inkonsistenzen. Dazu führen wir zunächst in Abschnitt 2 in Grundlagen und verwandte Arbeiten ein, bevor wir in Abschnitt 3 sowohl temporale Konsistenz als auch zwei mögliche Ansätze dazu, wie damit bei Anreicherung von Kontextinformationen umgegangen werden kann. Abschließend analysieren wir diese Ansätze in Abschnitt 4 und geben in Abschnitt 5 eine Zusammenfassung.

2 Grundlagen und Verwandte Arbeiten

Eine Reihe von Systemen, wie z.B. Care [ABC+04], Nexus [GBH+05], das CML Framework [Hen03], Gaia [RHC+02], ACoMS [HIR08] oder MAIS [CCMP06], fokussieren die Verwaltung von Kontextinformationen in einem gemeinsamen Kontextmodell. Sie betrachten dabei jedoch keine datenstrombasierten Kontextmodelle mit zeitlichen Semantiken. Unsere Arbeit hingegen beschäftigt sich mit dem Aktualisieren dynamischer Kontextinformationen und wie sich dieses auf die zeitliche Semantik auswirkt. Die zeitliche Semantik bei der Verarbeitung von Datenströmen spielt in Datenstrommanagementsystemen (DSMS), wie Aurora [ACc+03], STREAM [ABB+03], Borealis [AAB+05], PIPES [Kra07] oder Odysseus [AGG+12] eine wesentliche Rolle. Solche DSMS bieten eine universelle und flexible Möglichkeit zur Verarbeitung von potenziell unendlichen Datenströmen bereit. Ein Ansatz der Systeme, um mit potenziell unendlichen Datenströmen umgehen zu können, sind Fenster, in dem nur Ausschnitte des Datenstroms betrachtet werden. Die Umsetzung solcher Fenster kann bspw. durch Intervalle (vgl. [Kra07, AGG+12]) oder durch Positive/Negative Marker (vgl. [ACc+03]) erfolgen. In dieser Arbeit betrachten wir den Intervall-Ansatz nach [Kra07], da dort genau das beschriebene Problem mit der Blockierung auftreten kann. Beim Positiv/Negativ-Ansatz blockiert zwar das DSMS nicht, weil ein eingehendes Element direkt mit einem positiven Marker versehen wird, damit gültig ist und dann direkt verarbeitet werden kann. Jedoch muss dann die Anwendung nach dem DSMS warten bis das passende Element mit negativem Marker auftritt. Entsprechend wird die Blockierung einfach nur in die Anwendung verschoben. Wie beim Intervall-Ansatz, ist hier ein Datenstrom S als eine Menge von Tupeln $(e, [t_s, t_e))$ definiert,

die jeweils aus Nutzdaten e und einem Gültigkeitsintervall $[t_s, t_e)$ bestehen. Ein Ereignis e ist vom Startzeitstempel t_s bis zum Endzeitstempel t_e gültig, dessen Ereignisse chronologisch anhand von t_s aufsteigend sortiert ist.

3 Temporale Konsistenz

Wenn zwei oder mehr Ereignisse in Beziehung zu einander kommen, indem sie bspw. verknüpft werden, muss deren zeitlicher Kontext zueinander passen. So betrachtet der Intervall-Ansatz für eine eindeutige Semantik nur solche Ereignisse zusammen, dessen Gültigkeitsintervalle sich überschneiden. In Anlehnung daran definieren wir temporale Konsistenz für ein bereits zusammengesetztes Ereignis. Ein Ereignis $(e, [t_s, t_e))$ ist dann temporal konsistent, wenn dessen Teile $(e_a, [t_{s_a}, t_{e_a}))$ und $(e_b, [t_{s_b}, t_{e_b}))$ zum gleichen Zeitpunkt gültig waren, sodass $t_s = max(t_{s_a}, t_{s_b})$, $t_e = min(t_{e_a}, t_{e_b})$ und $t_e \geq t_s$ gilt.

Neben vorhandenen Operationen wie Joins oder Aggregationen (vgl. [Kra07]), muss auch das Anreichern mit Kontextinformationen temporale Konsistenz zusichern. Werden bspw. Bewegungsdaten mit dem aktuellen Zustand der Tür angereichert, so muss der verwendete Zustand der Tür auch den gleichen Zeitraum wie die Bewegungsdaten betrachten. Abbildung 1 zeigt verschiedene Möglichkeiten, wie der aktuelle Zustand der Tür mit Bewegungsereignissen von John, Mary bzw. Bob verknüpft werden können. Entsprechend

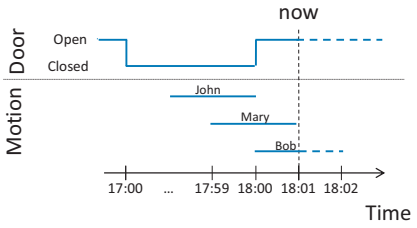


Abbildung 1: Beispiele für mögliche Anreicherungen

stellt Tabelle 1 die zum Zeitpunkt *now* gültig Ereignisse dar. Dabei ist zu sehen, dass Tür *4*

Türereignisse	Bewegungsereignisse
<4, opened, 10:00>	*<John, Flur, 4, [17:58, 18:00)>*
<4, closed, 17:00>	*<Mary, Flur, 4, [17:59, 18:01)>*
<4, opened, 18:00>	*<Bob, Flur, 4, [18:00, 18:02)>*

Tabelle 1: Beispiele für Tür- und Bewegungsereignisse

um *10:00* geöffnet und um *17:00* Uhr wieder geschlossen wurde. Die Bewegungsereignisse zeigen, dass sich *John* von *17:58* bis *18:00* Uhr im Flur in Richtung Tür *4* bewegt hat. In diesem Beispiel lassen sich drei Probleme erkennen, die beim Anreichern auftreten können. Das erste Bewegungsereignis von *John* betrachtet einen älteren Zeitraum von *17:58*

bis *18:00* Uhr und schneidet damit nicht den aktuellen Zeitpunkt *now* bzw. *18:01*. Damit das angereicherte Ergebnis temporal konsistent ist, ist ein veralteter Türzustand nötig. Das zweite Bewegungsereignis von *Mary* zeigt, dass sich der Zustand der Tür während der Bewegung ändern kann, so dass eine Anreicherung beide Zustände berücksichtigen muss. Das dritte Bewegungsereignis von *Bob* zeigt die Unsicherheit bzgl. der Zukunft. So ist zum aktuellen Zeitpunkt *18:01* noch unbekannt, wie der Zustand der Tür zum Zeitpunkt *18:02* sein wird. Um die temporale Konsistenz in allen drei Fällen zu gewährleisten, gibt es zwei mögliche Ansätze. Ein gebräuchlicher Ansatz in DSMS sind blockierende Fenster, die zwar temporale Konsistenz zusichern können, aber nicht optimal für selten auftretende Ereignisse sind. Als Alternative stellen wir einen optimistischen Ansatz vor, in dem zwar nicht stets temporale Konsistenz zugesichert werden kann, dafür jedoch nicht blockierend ist.

3.1 Blockierender Ansatz

Der blockierende Ansatz entspricht dem für intervallbasierte DSMS übliche Ansatz, so dass entsprechende Fenster eingesetzt werden. Hierbei wird für jede Tür ein gleitendes elementbasiertes Fenster der Größe 1 verwendet, welches dann für zwei Dinge sorgt. Zum einen sorgt es dafür, dass zu einem Zeitpunkt nur ein Zustand gültig sein kann, indem es das Ende des Gültigkeitsintervalls auf den Anfang des nachfolgenden Zustands setzt. Zum anderen wartet das Fenster so lange, bis dessen Zustand eindeutig ist, um dadurch Unsicherheiten bzgl. der Zukunft zu vermeiden. Abbildung 2 zeigt ein Beispiel, wie ein elementbasiertes Fenster für zustandswechselnde Ereignisse eingesetzt werden kann, die in einem *Store* gespeichert werden sollen. Kommt das erste Ereignis *(open, [10:00, ∞))*

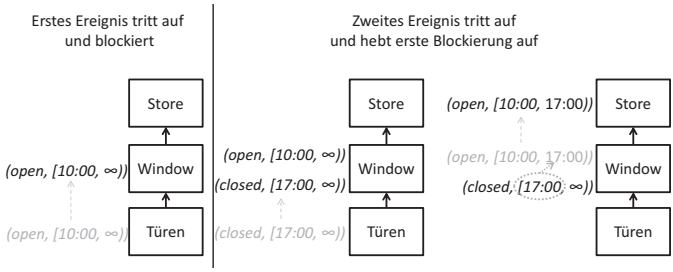


Abbildung 2: Beispiel eines blockierenden Fensters

aus der Quelle *Door* beim *Fenster* an, wartet es dort auf das zweite Ereignis, da derzeit noch nicht bekannt wie lange der Zustand *open* gültig ist. Kommt das zweite Ereignis *(closed, [17:00, ∞))* beim Fenster an, bekommt das erste Ereignis den Startzeitstempel *17:00* als Endzeitstempel zugewiesen und das erste Ereignis kann dann zur Verwendung weitergeleitet werden. Hierbei ist zu erkennen, dass man dadurch zwar eindeutig weiß, dass die Tür den Zustand *open* von *10:00* bis *17:00* Uhr hatte, jedoch muss in diesem Fall dafür bis zum Zeitpunkt *17:00* gewartet werden. Wird der Zustand während des Wartens benötigt, um z.B. Bewegungsdaten anzureichern, müssen auch diese Bewegungsdaten bis zu diesem

Zeitpunkt warten, um die temporale Konsistenz zu gewährleisten. Dieses Verhalten beim Anreichern wird in Abbildung 3 dargestellt, in dem ein Kontextspeicher eingesetzt wird, um den letzten Zustand eines Ereignisses zu speichern. In dem dargestellten Beispiel wur-

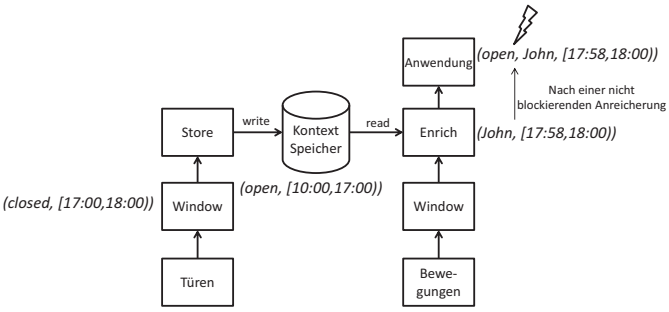


Abbildung 3: Eine temporale inkonsistente Anreicherung mit Blockierung

de der Kontextspeicher jedoch noch nicht aktualisiert, sodass das *Enrich* zum Anreichern den Wert *open* verwenden würde, wenn dort nicht die Zeit berücksichtigt wird. Betrachtet man jedoch die Gültigkeitsintervalle, hätte das Ereignis *(John [17:58, 18:00))* nicht mit *open* sondern mit *closed* angereichert werden müssen. Entsprechend hätte der *Enrich* wie beim Fenster auch blockieren müssen, um ein temporale konsistentes Ergebnis zu erzeugen. Des Weiteren macht sich dies auch bei dem nötigen Speicher bemerkbar. So muss das Fenster für die Türereignisse zwar nur ein Ereignis von *10:00* bis *17:00* Uhr zwischenspeichern, jedoch muss dadurch auch der *Enrich* blockieren. In diesem Fall würden also auch alle Bewegungsdaten zwischen *10:00* und *17:00* zwischengespeichert werden.

Zusammenfassend lässt sich also sagen, dass ein solcher Ansatz zwar temporale Konsistenz zusichern kann, aber dafür an verschiedenen Stellen warten muss. Ferner blockiert das System, welches sich einerseits negativ auf den benötigten Speicher auswirkt und andererseits die Reaktionsfähigkeit des Systems merklich vermindern kann und auch bis zur vollständigen Unbenutzbarkeit führen kann.

3.2 Optimistischer Ansatz

Da der blockierende Ansatz mit einem einelementigen Fenstern nicht für langsam ändernde Kontextwechsel brauchbar ist, betrachten wir einen alternativen Ansatz. Hierbei wird optimistisch davon ausgegangen, dass im Kontextspeicher stets der aktuell gültige Zustand vorhanden ist. Dabei wird das Ende des Gültigkeitsintervalls eines Zustands nicht auf einen festen Zeitpunkt festgesetzt, sondern ist solange gültig bis er explizit überschrieben wird. Abbildung 4 zeigt diese Variante. Obwohl das Ergebnis in diesem Beispiel bezüglich der Definition von formal korrekt und damit temporal konsistent ist, hätte das Ereignis *(John, [17:58, 18:00))* mit dem Zustand *(closed, [17:00, ∞))* angereichert werden müssen, da es den eigentlichen Zustand widerspiegelt. Analog zum Transaktionsmanagement in DBMS kann dies als Dirty Read bezeichnet werden. Jedoch kann so ein Dirty

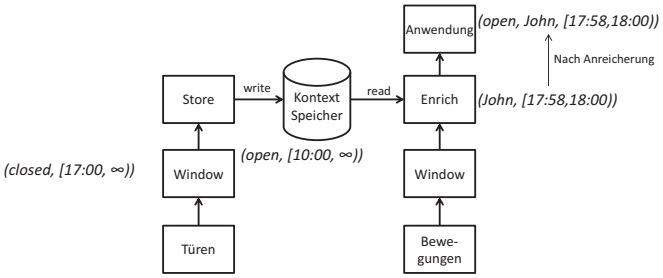


Abbildung 4: Dirty Read bei Anreicherung ohne Blockierung

Read jedoch nur auftreten, wenn das neue Türereignis, in diesem Fall *(closed, [17:00, ∞))* nicht rechtzeitig genug von der Quelle zum Kontextspeicher gelangen kann.

Da es auch mehr als einen lesenden Zugriff auf den Kontextspeicher geben kann, kann es sein, dass dasselbe Ereignis mit zwei verschiedenen Zuständen angereichert wird. Dies ist ähnlich vergleichbar zu einem Non-repeatable Read bei DBMS und zeigt Abbildung 5. Hier liegen mit *(open, [10:00, ∞))* und *(closed, [17:00, ∞))* zwei Zustände der Türen vor.

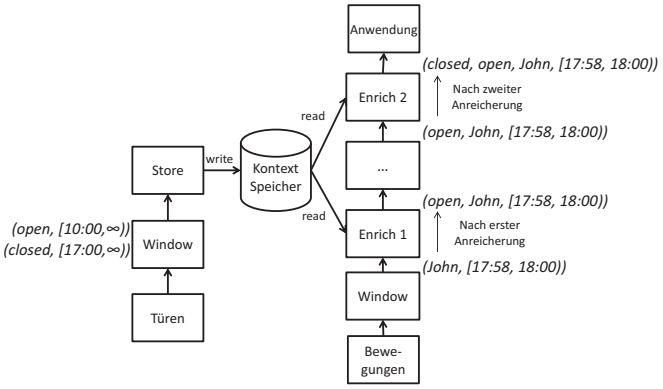


Abbildung 5: Non-repeatable Read bei Anreicherung ohne Blockierung

Wird das erste Ereignis im Kontextspeicher abgelegt, dann kann in *Enrich 1* bei der ersten Anreicherung verwendet werden. Wenn dann, bevor das angereicherte Tupel *Enrich 2* erreicht, der Zustand *(open, [10:00, ∞))* im Kontextspeicher durch *(closed, [17:00, ∞))* ersetzt wird, wird der neue Zustand für die zweite Anreicherung verwendet. Da bei einem Non-repeatable Read stets ein Dirty Read vorausgegangen sein muss, betrachten wir jedoch im Folgenden nur Dirty Reads.

Zusammengefasst erlaubt der optimistische Ansatz einen direkten Zugriff auf den letzten bekannten Zustand ohne Blockierung. Jedoch muss dafür in Kauf genommen werden, dass ggf. nicht der eigentlich korrekt Zustand verwendet wird, wenn dieser zu lange zum eigentlich Kontextspeicher benötigt.

3.3 Integration

Für die Integration in ein DSMS bietet sich ein Key-Value basierter Speicher an, wie er in Abbildung 6 dargestellt wird. Der Kontextspeicher übernimmt dann die Verwaltung der

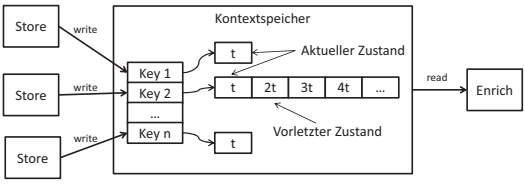


Abbildung 6: Zugriffsstruktur des Kontextspeichers

Kontextinformationen, sodass je nach Anwendung beide Ansätze verwendet werden können. Beispielsweise können zum einen verschiedene Speicherstrategien verwendet werden, wie es in der Abbildung für *Key 1* und *Key 2* verdeutlicht wird. So wird für die Kontextinformation in *Key 1* stets nur der letzte Zustand gespeichert. Bei *Key 2* hingegen kann zusätzlich die Historie gespeichert werden, um ggf. auch ältere Informationen für den *Enrich* bereitstellen zu können. Eine solche Historie ist bspw. notwendig, um auf non-repeatable reads reagieren zu können. Das Säubern der Historie kann bspw. auf Grundlage des Zeitstempels aus dem anzureichernden Ereignis im *Enrich* gemacht werden. Alternativ bieten sich bspw. auch logarithmische Speicherverfahren an, die Daten zusammenfassen je älter sie sind.

4 Analytische Betrachtungen

Die Wahl zwischen blockierendem und optimistischem Ansatz ist abhängig von der Anzahl der Kontextwechsel. Für Kontextinformationen, die ihren Zustand in Millisekunden ändern, bietet der blockierende Ansatz die bessere Wahl, da er in jedem Fall temporale Konsistenz zusichern kann und dies bei steigender Frequenz auch notwendiger wird. Gleichzeitig relativiert sich auch die Blockierung bei steigender Frequenz. Bei selten auftretenden Kontextwechseln ist die Blockierung hinderlich und die temporale Konsistenz weniger notwendig. Entsprechend wäre der optimistische Ansatz die bessere Wahl, obwohl es nicht immer temporale Konsistenz sicherstellen kann. Letztlich ist es ein Kompromiss zwischen temporaler Konsistenz bzw. Unsicherheit und Verfügbarkeit der Ergebnisse. Wie hoch die Unsicherheit beim optimistischen Ansatz ist, wird im Folgenden analysiert. Dazu zeigt Abbildung 7 einen einfachen Aufbau, anhand derer man die Anzahl der Dirty Reads bzw. die Genauigkeit der angereicherten Ergebnisse berechnen kann. Ob ein Ereignis zu spät im Kontextspeicher ankommt und damit ein Dirty Read erzeugt wird, liegt an der Verarbeitungszeit für das Ereignis, die das System vom Eintreffen bis zum Kontextspeicher benötigt. Diese Latenz wird in Abbildung 7 mit l_c für die Kontextinformation und mit l_e für das anzureichernde Ereignis bezeichnet. Latenzen, die z.B. durch Übertragung auch außerhalb des Systems entstehen, können in diese Latenzen hinzuge-

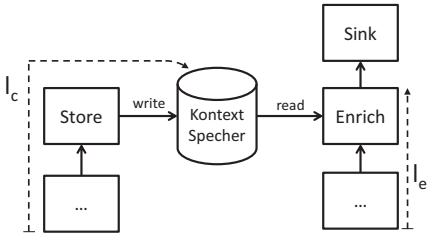


Abbildung 7: Relevante Latenzen

rechnet werden. Des Weiteren sind Latenzen stark abhängig von den Verarbeitungsschritten. Während eine Selektion relativ gleiche Latenzen verursacht, sind Latenzen bei einem Join stark von den Daten abhängig. Durch solche Operatoren kann dann auch eine Kontextinformation nicht rechtzeitig beim Kontextspeicher ankommen und damit ein Dirty Read verursachen. Die Zeitspanne, die eine Kontextinformation eigentlich veraltet ist aber der Kontextspeicher noch nicht aktualisiert wurde, wird als *kritische Phase* bezeichnet und ergibt sich aus $d := l_c - l_e$, wenn $l_c > l_e$ und sonst $d := 0$. Um die Anzahl möglicher Dirty Reads zu bestimmen, betrachten wir die Periodendauer p_e der anzureichernden Ereignisse. Bei einem Kontextwechsel, also während einer kritischen Phase d können dann $r := \left\lceil \frac{d}{p_e} \right\rceil$ Dirty Reads erzeugt werden. Betrachtet man dieses Verhalten über einen Zeitraum o, in der insgesamt n Kontextwechsel auftreten, so können maximal $\frac{r \cdot n}{o}$ Dirty Reads passieren. Hierbei sei noch einmal erwähnt, dass Latenzen und Periodendauer in der Regel nicht konstant sind. Beobachtet man jedoch l_c, l_e und p_e über den Zeitraum o, können deren statistischen Ausprägungen für verschiedene Szenarien verwendet werden. Tabelle 2 zeigt eine Übersicht über die benötigten Werte. Möchte man den schlimmsten Fall (worst case),

	l_c	l_e	p_e
Best case	Minimum	Maximum	Maximum
Average case	Average	Average	Average
Worst case	Maximum	Minimum	Minimum

Tabelle 2: Auswahl der Werte für Anzahl Dirty Reads

also die maximale Anzahl an Dirty Reads berechnen, verwendet man das Maximum von l_c und das Minimum von l_e bzw. p_e. Für $d = 0$ gilt stets der beste Fall.

Im Folgenden zeigen wir ein Beispiel, bei der die Tür während einer Stunde mehrmals verwendet wird, so dass $n \in \{1, 10, 100, 1000\}$ und $o = 1h = 3.600.000ms$ Des Weiteren soll der Bewegungssensor zwischen 1Hz, 10Hz, 100Hz und 1kHz periodisch Daten senden, sodass $p_e \in \{1000, 100, 10, 1\}$. Wir wählen eine relativ hohe kritische Phase von 20 ms. Tabelle 3 zeigt die prozentuale Anzahl von Dirty Reads für dieses Beispiel. Hat der Bewegungssensor eine Frequenz von 100 Hz, sodass $p_e = 10ms$, dann gibt es maximal $\left\lceil \frac{20}{10} \right\rceil = 2$ Ereignisse während einer kritischen Phase. Hat man entsprechend 10 Kontextwechsel, so kann es insgesamt e=20 Dirty Reads geben. Des Weiteren gibt es insgesamt $c = \frac{o}{p_e} = \frac{3.600.000ms}{10ms} = 360.000$ Ereignisse pro Stunde. Entsprechend ist der Anteil an mög-

Anzahl (n) der Kontextwechsel	Zeit zwischen Ereignissen (p_e)			
	1000ms	100ms	10ms	1ms
1	0.02778	0.00278	0.00056	0.00056
10	0.27778	0.02778	0.00556	0.00556
100	2.77778	0.27778	0.05556	0.05556
1000	27.77778	2.77778	0.55556	0.55556

Tabelle 3: Beispiel für Dirty Reads in Prozent

lichen Dirty Reads $\frac{e}{c} = \frac{20}{360.000} = 0.00556\%$. Es lässt sich in dem dargestellten Beispiel erkennen, dass die Anzahl der Dirty Reads durch höhere Frequenzen verringert werden kann, sodass in der größeren Menge der Ereignisse ein einzelner Fehler weniger signifikant ist. Jedoch gilt dies nur für Fälle mit $p_e \geq d$, wie auch in Tabelle 3 für $p_e = 1ms$ und $p_e = 10ms$ kein Unterschied zu sehen ist. Dies liegt daran, dass durch die höhere Frequenz mehr Ereignisse während einer kritischen Phase auftreten können.

5 Zusammenfassung

Dieser Beitrag beschäftigt sich mit dem Problem temporaler Konsistenz, welches bei Verknüpfung von mehreren Ereignissen gewährleistet werden soll, um dadurch den zeitlichen Zusammenhang der Realwelt korrekt widerzuspiegeln. Die Verwendung Fensteransätze, wie sie üblicherweise bei Datenstromsystemen verwendet wird, sichert zwar diese temporale Konsistenz zu, ist jedoch auf Grund der Blockierung nicht bei selten auftretenden Ereignissen sinnvoll. Alternativ haben wir daher einen optimistischen Ansatz vorgestellt, der zwar nicht stets temporale Konsistenz zusichern kann, dafür jedoch nicht blockierend ist. Ergänzend haben wir analysiert, wann und wie oft bei bestimmten Konfigurationen temporale inkonsistente Ergebnisse erzeugt werden können.

Das Problem der Blockierung bei gleichzeitiger temporaler Konsistenz ist bei uns aufgetreten, als wir ein DSMS als Teil einer Middleware in einem Smart Home einsetzen wollten. Entsprechend werden wir die Umsetzung dieses Ansatzes mit dem Kontextspeicher in einem *Living Lab* für Smart Homes integrieren und evaluieren. Das Konzept selbst betrachtet derzeit nur einen Store-Operator pro Kontextinformation, um dadurch Schreibkonflikte wie z.B. Lost-Updates zu vermeiden. Hier muss man entsprechend weitere Konzepte zur Transaktionskoordination entwickeln. Des Weiteren ist der Kontextspeicher in Kombination mit weiteren Verfahren wie z.B. Out-of-Order Verarbeitung zu untersuchen. Es bietet sich hierbei insbesondere auch die Verarbeitung mit Unsicherheiten [JKV07] an, indem der prozentuale Anteil an Dirty Reads (vgl. Tabelle 3) verwendet wird, um auszusagen, wie korrekt der angereicherte Wert ist. Der hier vorgestellte Ansatz erlaubt eine direkte Annotation von Tupeln mit der entsprechenden Unsicherheit.

Literatur

[AAB+05] D. Abadi, Yanif A., M. B., U. Cetintemel, M. Cherniack, J. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing und S. Zdonik. The Design of the Borealis Stream Processing Engine. In *CIDR 2005*, 2005.

[ABB+03] A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, I. Nishizawa, J. Rosenstein und J. Widom. STREAM: the stanford stream data manager. In *Proc. of SIGMOD*, 2003.

[ABC+04] A. Agostini, C. Bettini, N. Cesa-Bianchi, D. Maggiorini, D. Riboni, M. Ruberl, C. Sala und D. Vitali. Towards Highly Adaptive Services for Mobile Computing. In *Proc. of IFIP*, Jgg. 158, 2004.

[ACc+03] D. J Abadi, D. Carney, U. Çetintemel, M. Cherniack, S. Convey, C.and Lee, M. Stonebraker, N. Tatbul und S. Zdonik. Aurora: a new model and architecture for data stream management. *The VLDB Journal*, 12(2), 2003.

[AGG+12] H. Appelrath, D. Geesen, M. Grawunder, T. Michelsen, D. Nicklas et al. Odysseus: a highly customizable framework for creating efficient event stream management systems. In *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, Seiten 367–368. ACM, 2012.

[CCMP06] C. Cappiello, M. Comuzzi, E. Mussi und B. Pernici. Context Management for Adaptive Information Systems. *Electronic Notes in Theoretical Computer Science*, 146(1), 2006.

[EMR+10] Opher Etzion, Yonit Magid, Ella Rabinovich, Inna Skarbovsky und Nir Zolotorevsky. Context aware computing and its utilization in event-based systems. In *Proc. of the DEBS*, Seiten 270–281, New York, 2010. ACM.

[EN11] Opher Etzion und Peter Niblett. *Event Processing in Action*. Manning, 2011.

[GBH+05] M. Grossmann, M. Bauer, N. Honle, U. P Kappeler, D. Nicklas und T. Schwarz. Efficiently Managing Context Information for Large-Scale Scenarios. In *Proc. of Pervasive Computing and Communications*, 2005.

[Hen03] K. Henricksen. *A Framework for Context-Aware Pervasive Computing Applications*. Dissertation, School of Information Technology and Electrical Engineering, The University of Queensland, September 2003.

[HIR08] P. Hu, J. Indulska und R. Robinson. An Autonomic Context Management System for Pervasive Computing. In *Intl. Conf. on PerCom*, 2008.

[JKV07] TS Jayram, S. Kale und E. Vee. Efficient aggregation algorithms for probabilistic data. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Seiten 346–355. Society for Industrial and Applied Mathematics, 2007.

[Kra07] J. Kraemer. *Continuous Queries over Data Streams-Semantics and Implementation*. Dissertation, Philipps-Universität Marburg, 2007.

[RHC+02] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell und K. Nahrstedt. A Middleware Infrastructure for Active Spaces. *IEEE Pervasive Computing*, 1(4), 2002.

Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS)

Jana Dittmann, Arno Fischer, Gunter Saake, Claus Vielhauer

The 1st Workshop on Databases in Biometrics, Forensics and Security Applications (DBforBFS) is a satellite, half-day workshop of the BTW 2013 at the Otto-von-Guericke University of Magdeburg in Germany, March 11/12, 2013. The workshop is intended for discussing new challenges, innovations, recent experiences and knowledge in the areas of databases in the focus for biometrics, forensics and security. The workshop goal is to bring special, important research topics together enhancing the general BTW conference scope. Motivated by today's challenges from the disciplines six papers have been accepted covering topics about: authentication challenges, forensic trace requirements and its privacy, GPU co-processing and information leakage aspects. We hope, that the workshop stimulates exchange and discussion between different experts. We thank all BTW organizers, workshop chairs and committees for accepting and supporting the workshop as well as all authors for their valuable contributions.

Program Chairs

Jana Dittmann (Otto-von-Guericke University Magdeburg, Germany)
Arno Fischer (Brandenburg University of Applied Science, Germany)
Gunter Saake (Otto-von-Guericke University Magdeburg, Germany)
Claus Vielhauer (Brandenburg University of Applied Science, Germany)

Program Committee

Ruediger Grimm (University of Koblenz, Germany)
Dominic Heutelbeck (FTK, Germany)
Stefan Katzenbeisser (Technical University Darmstadt, Germany)
Claus-Peter Klas (FTK, Germany)
Sviatoslav Voloshynovskiy (unige, Switzerland)
Edgar R. Weippl (sba-research, Austria)

Physical object authentication with correlated camera noise

F. P. Beekhof, S. Voloshynovskiy, M. Diephuis, F. Farhadzadeh

CUI
University of Geneva
route de Drize 7
1227 Carouge
svolos@unige.ch

Abstract: In this paper, we address the problem of physical object authentication based on surface microstructure images. The authentication is based on digital content fingerprints computed from the microstructure images. We analyze the impact of camera distortions which follow additive correlated Gaussian noise. The optimal decision rule is derived and the performance is analyzed in the direct and transformed domain. The theoretical derivations are supported by experimental results obtained for the FAMOS dataset.

1 Introduction

Counterfeit products are increasingly present in the market, leading to a number of issues for legitimate manufacturers and consumers alike. Although manufacturers might be primarily interested in preventing any loss of income, the well-being of consumers can be seriously affected by counterfeit products such as car brakes or medication.

Although solutions exist where original items or their packaging are modified, such as by attaching holograms or embedding a digital watermark, a recent development has been to base protection systems on forensic features that require no modification of the items. Forensic techniques are based on the intrinsic features of the objects to be protected. A particularly attractive option is the use of microstructures, which can be acquired by commonly available optical equipment. Microstructures have been shown to be present in almost all materials and to be robust against rough treatment of the material [BCJ+05].

Although the use of microstructures is an attractive option, their usage directly entails a number of drawbacks, most notably the large storage space required, concerns about the safety of the stored data, and the effort required to process the data during queries. A solution to these concerns exists in the form of *digital content fingerprints*, which are short, robust and informative representations of these images.

A similar approach is taken in the field of biometrics [TSE07, Ign09], where noisy data acquired from a person is transformed into a binary *template*, which must be protected due to the great importance of privacy in biometric applications. Due to the fact that microstructures are acquired in the form of images, there is a strong link with multimedia

security, and particularly robust image hashing [BL96, Fri00, FLL02, SMW06].

Fingerprints are typically stored in conjuction with an identifier, usually the index in a table of fingerprints, in a relational database. In this context it is clear that the identifier is a primary key in the database. Note that a *key* in the context of this work refers to a database key, not a secret key as used in cryptography, biometrics or multimedia security. Although the identifier is a valid database key, the fingerprint should satisfy many of the same constraints, namely, a fingerprint always has a value and is expected to be uniquely associated with the item it is derived from. A difference between traditional database keys is that fingerprints are random, determined by the natural randomness of the items from which they are derived and partially by noise. This implies several differences with traditional database keys. First, the uniqueness of fingerprints is not guaranteed, whereas a simple automatic increment is sufficient to guarantee the uniqueness of traditional database keys. Second, the fingerprints calculated from different observations of the same item differ slightly due to the noise. This can be likened to executing database operations on a machine where data are corrupted due to failing memory modules. The presence of noise has two very serious consequences: first, the time-complexity of a database lookup increases from $O(1)$ to exponential in the fingerprint length; second, the lookup might return an erroneous result due to the fact that the introduced uncertainty renders it impossible to confirm that a match in correct.

The problem wherein the identity of the item under investigation is established by matching a query fingerprint against the full database of fingerprints, is refered to as the identification problem. In this work, we address a simpler case, the authentication problem, which only establishes if an unknown item is in fact a specific item in the database indicated by the user. This can be seen as a query with a fingerprint against one specific row of the fingerprint table in the database, or against a table with one row. In this work we will investigate the probability of error of a query based on fingerprints in the authentication setup, and propose improved matching rules to decrease that probability of error.

In previous work, the authors have released *FAMOS* [VDB$^+$12], a forensics dataset of images of microstructure of cardboard boxes, and made several theoretical and practical contributions to the field of fingerprinting [Bee12].

Although significant progress has been made in the field, there is still room for improvement of the fingerprinting technology in terms of robustness and the precision of matching. In this work, we propose enhanced observation models of the noise, and corresponding rules for the matching of observations and fingerprints, leading to improved performance.

1.1 Notation

Bold capitals $\mathbf{X}$ denote vector random variables. Corresponding small letters $\mathbf{x}$ denote their respective realizations. The binarized version of $\mathbf{x}$ is represented by $\mathbf{b}_\mathbf{x}$. $\mathbf{X} \sim f(\mathbf{x})$ indicates that the random vector follows distribution $f(\mathbf{x})$. The identity matrix is denoted as $\mathbb{I}$.

2 Model of Authentication

Although other solutions exist, physical objects can be protected against counterfeiting through authentication services, which determine if a particular object is truly the authentic object m that it is claimed to be.

An overview of authentication for physical objects is given in Figure 1. There are two phases in the authentication system: *enrollment*, where authentic items are produced and their fingerprints are placed in a database; followed by *verification*, wherein it is claimed that an unknown object is the authentic object with identify m, and it must be decided if this is true or not. During the enrollment, an image of the microstructure of an authen-

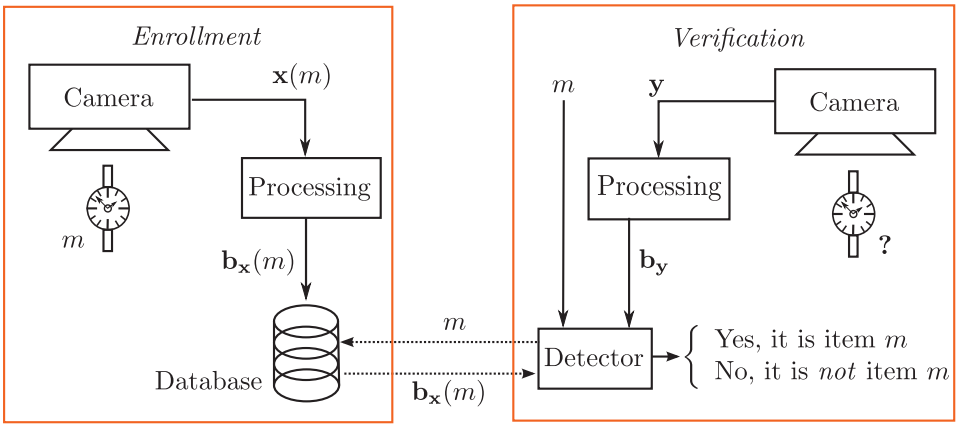


Figure 1: An authentication scheme based on fingerprinting of microstructures.

tic object, which is given an identity m, is acquired. The image obtained from item m is represented by the vector $\mathbf{x}(m)$, which is transformed into a fingerprint $\mathbf{b}_\mathbf{x}(m)$. The fingerprint is then stored in a database.

During verification, the noisy data $\mathbf{y}$, acquired from the object under investigation, is used to calculate a fingerprint $\mathbf{b}_\mathbf{y}$. The claimed identity m is used to retrieve $\mathbf{b}_\mathbf{x}(m)$, the content fingerprint of the authentic object m, from the database. The decision about the authenticity of the object under verification is then made by comparing $\mathbf{b}_\mathbf{x}(m)$ with $\mathbf{b}_\mathbf{y}$.

We can then formulate the authentication problem as a binary hypothesis test to answer the question whether or not the observed item is the authentic item m, i.e.:

$$\begin{cases} \mathcal{H}_0 : \text{No, it is } \textit{not} \text{ the authentic item } m, \\ \mathcal{H}_m : \text{Yes, it is the authentic item } m. \end{cases} \tag{1}$$

2.1 Mathematical formulation of Authentication

The process shown in Figure 1 can be expressed mathematically as shown in Figure 2, which details the calculation of the fingerprint. During the enrollment stage, a source $\mathbf{X}$

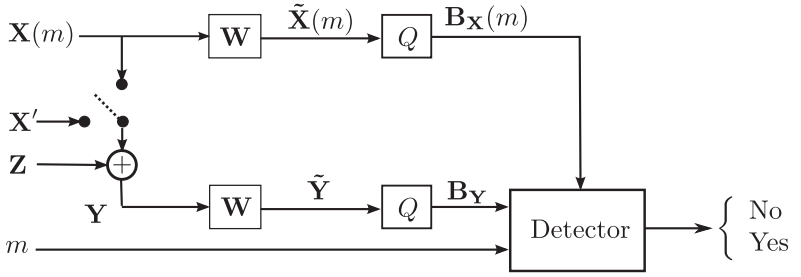


Figure 2: Mathematical overview of authentication based on content fingerprinting.

generates a vector of features associated with m, the identity of the object, resulting in a realisation denoted as $\mathbf{x}(m)$. The calculation of the fingerprint is modeled as a two-step process: in the first stage, the data is transformed by a matrix $\mathbf{W}$, producing $\tilde{\mathbf{X}}(m)$, which can be quantized by a function Q to produce the fingerprint $\mathbf{B}_{\mathbf{X}}(m)$.

During verification, either the authentic object $\mathbf{x}(m)$ is observed, or any other item denoted as $\mathbf{X}'$. We assume that $\mathbf{X}'$ is generated from the same source that produced $\mathbf{x}(m)$. We assume that the counterfeiters have the same equipment or the technological details of the manufacturing process. However, we assume that they can not control the physical randomness of microstructures, which is a strong assumption. Mathematically, this can be expressed by assuming that counterfeiters can produce counterfeit items $\mathbf{X}'$ such that $\mathbf{X}' \sim f(\mathbf{X})$. The acquisition during verification may introduce additive noise $\mathbf{Z}$ that is independent of the item under investigation, producing a vector $\mathbf{Y}$. The noisy vector $\mathbf{Y}$ is transformed using $\mathbf{W}$ and quantized by Q, producing $\mathbf{B}_{\mathbf{Y}}$.

We will investigate the authentication problem in the direct and transformed domains. In the *direct* domain, the data evaluated directly, i.e. the data is not modified. In the *transformed* domain, the data first undergoes a transformation, represented by $\mathbf{W}$ in Figure 2, and the transformed data is evaluated.

2.2 Direct Domain

In mathematical terms, we can reformulate the hypothesis test in the direct domain as:

$$\begin{cases} \mathcal{H}_0 : \mathbf{Y} = \mathbf{X}' + \mathbf{Z}, \\ \mathcal{H}_m : \mathbf{Y} = \mathbf{x}(m) + \mathbf{Z}. \end{cases} \tag{2}$$

In previous work [VKP08, VDB+12, Bee12], it was assumed that both the source $\mathbf{X}$ and the noise were $\mathbf{Z}$ are i.i.d. Gaussian, from which it follows that the correlation is a sufficient statistic in the direct domain. In this work, we relax these assumptions and allow

for correlated noise, i.e., we assume that $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{zz})$. We make no assumptions about the counterfeit items $\mathbf{X}'$ other than that they are generated from the same distribution as authentic items. Consequently, the hypothesis test can be reformulated in terms of the distributions:

$$\begin{cases} \mathcal{H}_0 : \mathbf{Y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx} + \mathbf{K}_{zz}), \\ \mathcal{H}_m : \mathbf{Y} \sim \mathcal{N}(\mathbf{x}(m), \mathbf{K}_{zz}). \end{cases} \tag{3}$$

Following the Neyman-Pearson framework, we can then formulate the decision rule for a chosen threshold γ which can be developed using Bayes' rule:

$$\frac{\Pr[\mathcal{H}_m \mid \mathbf{y}]}{\Pr[\mathcal{H}_0 \mid \mathbf{y}]} > \gamma \qquad \Leftrightarrow \qquad \frac{\frac{f(\mathbf{y}|\mathcal{H}_m)p(\mathcal{H}_m)}{f(\mathbf{y})}}{\frac{f(\mathbf{y}|\mathcal{H}_0)p(\mathcal{H}_0)}{f(\mathbf{y})}} > \gamma. \tag{4}$$

Assuming $p(\mathcal{H}_0) = p(\mathcal{H}_m) = \frac{1}{2}$, which implies the greatest uncertainty about the hypothesis in force, yields:

$$\frac{f(\mathbf{y} \mid \mathcal{H}_m)}{f(\mathbf{y} \mid \mathcal{H}_0)} > \gamma \tag{5}$$

$$\frac{\frac{1}{\sqrt[N]{2\pi}\sqrt{|\mathbf{K}_{zz}|}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{x}(m))^T \mathbf{K}_{zz}^{-1}(\mathbf{y} - \mathbf{x}(m))\right)}{\frac{1}{\sqrt[N]{2\pi}\sqrt{|(\mathbf{K}_{xx}+\mathbf{K}_{zz})|}} \exp\left(-\frac{1}{2}\mathbf{y}^T(\mathbf{K}_{xx} + \mathbf{K}_{zz})^{-1}\mathbf{y}\right)} > \gamma, \tag{6}$$

where $|.|$ denotes the determinant of the matrix, and which can be further developed to obtain a sufficient statistic [Kay98]:

$$t(\mathbf{y}) = \mathbf{y}^T(\mathbf{K}_{xx} + \mathbf{K}_{zz})^{-1}\mathbf{y} - (\mathbf{y} - \mathbf{x}(m))^T \mathbf{K}_{zz}^{-1}(\mathbf{y} - \mathbf{x}(m)) \tag{7}$$

$$= \mathbf{y}^T(\mathbf{K}_{xx} + \mathbf{K}_{zz})^{-1}\mathbf{y} - \mathbf{y}^T\mathbf{K}_{zz}^{-1}\mathbf{y} + 2\mathbf{y}^T\mathbf{K}_{zz}^{-1}\mathbf{x}(m) - \mathbf{x}^T(m)\mathbf{K}_{zz}^{-1}\mathbf{x}(m). \tag{8}$$

A simplified sufficient statistic $s(\mathbf{y})$ can be formulated by assuming that the terms $\mathbf{y}^T(\mathbf{K}_{xx} + \mathbf{K}_{zz})^{-1}\mathbf{y}$, $\mathbf{y}^T\mathbf{K}_{zz}^{-1}\mathbf{y}$ and $\mathbf{x}^T(m)\mathbf{K}_{zz}^{-1}\mathbf{x}(m)$ are all constant for the expected observations:

$$s(\mathbf{y}) = \mathbf{y}^T\mathbf{K}_{zz}^{-1}\mathbf{x}(m). \tag{9}$$

2.3 Transformed Domain

The calculation of a fingerprint typically involves a transformation, which we assume to be linear and orthogonal. An example of such a transform is the DCT, but random projections, which are approximately orthogonal [VKB+10, FVK10] can be considered as well. Let

$$\tilde{\mathbf{X}} = \mathbf{W}\mathbf{X}, \quad \tilde{\mathbf{Y}} = \mathbf{W}\mathbf{Y}, \quad \text{and } \tilde{\mathbf{Z}} = \mathbf{W}\mathbf{Z}, \tag{10}$$

then

$$\tilde{\mathbf{Y}} = \mathbf{W}\mathbf{Y} = \mathbf{W}(\mathbf{X} + \mathbf{Z}) = \mathbf{W}\mathbf{X} + \mathbf{W}\mathbf{Z} = \tilde{\mathbf{X}} + \tilde{\mathbf{Z}}, \tag{11}$$

where $\tilde{\mathbf{X}} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\tilde{x}\tilde{x}})$, $\tilde{\mathbf{Z}} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\tilde{Z}\tilde{Z}})$, and $\mathbf{K}_{\tilde{x}\tilde{x}} = \mathbf{W}\mathbf{K}_{\tilde{x}\tilde{x}}\mathbf{W}^T$ and $\mathbf{K}_{\tilde{z}\tilde{z}} = \mathbf{W}\mathbf{K}_{\tilde{z}\tilde{z}}\mathbf{W}^T$ [Jai89]. The hypothesis test in the transformed domain is then:

$$\begin{cases} \mathcal{H}_0 : \tilde{\mathbf{Y}} = \tilde{\mathbf{X}}' + \tilde{\mathbf{Z}}, \\ \mathcal{H}_m : \tilde{\mathbf{Y}} = \tilde{\mathbf{x}}(m) + \tilde{\mathbf{Z}}, \end{cases} \tag{12}$$

which can be reformulated as in the direct domain:

$$\begin{cases} \mathcal{H}_0 : \tilde{\mathbf{Y}} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\tilde{x}\tilde{x}} + \mathbf{K}_{\tilde{z}\tilde{z}}), \\ \mathcal{H}_m : \tilde{\mathbf{Y}} \sim \mathcal{N}(\tilde{\mathbf{x}}(m), \mathbf{K}_{\tilde{z}\tilde{z}}), \end{cases} \tag{13}$$

leading to a sufficient statistic in the transformed domain:

$$s(\tilde{\mathbf{y}}) = \tilde{\mathbf{y}}^T \mathbf{K}_{\tilde{z}\tilde{z}}^{-1} \tilde{\mathbf{x}}(m). \tag{14}$$

The challenge for systems designers is to choose a transform that leads to the most informative and robust fingerprints, which can be evaluated in information-theoretical terms, using the framework introduced in earlier work [Bee12].

3 Experimental Results

In this work, we have opted to use the DCT as transform because of its energy-compacting properties, which offer a good future perspective for dimensionality reduction.

As stated earlier, the statistic derived for i.i.d. Gaussian noise is the sum-inner-product, both in the direct and in the transformed domain [VKP08]:

$$r(\mathbf{y}) = \mathbf{y}^T \mathbf{x}(m), \quad r(\tilde{\mathbf{y}}) = \tilde{\mathbf{y}}^T \tilde{\mathbf{x}}(m). \tag{15}$$

There are two sources of error in authentication: first, an item that is *not* item m may nonetheless be accepted as such; second, the authentic item m may be wrongfully rejected. The latter case is refered to as a *miss*.

Let the probability of miss p_m, and false acceptance p_f be defined for each of the different statistics as:

$$p_m = \Pr\left[r(\mathbf{Y}) < t \mid \mathcal{H}_m\right] \qquad p_f = \Pr\left[r(\mathbf{Y}) \geq t \mid \mathcal{H}_0\right], \tag{16}$$

$$p_m = \Pr\left[r(\tilde{\mathbf{Y}}) < t \mid \mathcal{H}_m\right] \qquad p_f = \Pr\left[r(\tilde{\mathbf{Y}}) \geq t \mid \mathcal{H}_0\right], \tag{17}$$

$$p_m = \Pr\left[s(\mathbf{Y}) < t \mid \mathcal{H}_m\right] \qquad p_f = \Pr\left[s(\mathbf{Y}) \geq t \mid \mathcal{H}_0\right], \tag{18}$$

$$p_m = \Pr\left[s(\tilde{\mathbf{Y}}) < t \mid \mathcal{H}_m\right] \qquad p_f = \Pr\left[s(\tilde{\mathbf{Y}}) \geq t \mid \mathcal{H}_0\right], \tag{19}$$

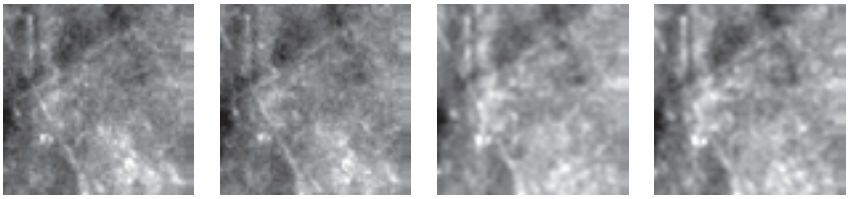
where t is a threshold that must be derived from the chosen threshold γ for each statistic. The authentication performance will be analysed in terms of a Receiver Operating Characteristic (ROC) curve.

3.1 Authentication Performance on Synthetic Data

The performance has first been evaluated for synthetic data, where $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ and $\mathbf{Z}$ was generated by a stationary Gauss-Markov process with $\rho_Z = 0.85$, i.e. $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{zz})$. The length of the vectors was 128, and 4096 different vectors were enrolled, then transformed with a one-dimensional DCT. The results are visible in Figure 5a, indicating in all cases that the use of the proposed measures $s(\mathbf{y})$ and $s(\tilde{\mathbf{y}})$ lead to greater precision than the use of regular inner products $r(\mathbf{y})$ and $r(\tilde{\mathbf{y}})$. It also demonstrated that there is no difference between the direct domain and the transformed domain. The transform does not alter the performance, i.e. the performance of $r(\mathbf{y})$ and $r(\tilde{\mathbf{y}})$ are similar, and the same holds for the performance of $s(\mathbf{y})$ and $s(\tilde{\mathbf{y}})$.

3.2 Authentication Performance on the FAMOS dataset

The FAMOS[1] dataset contains images of 5000 cardboard patches acquired with two different cameras [VDB+12]. Each cardboard patch is acquired three times with the same camera, resulting in acquisitions A, B and C and a total of 30000 images. Examples of these acquisitions can be seen in Figure 3.



(a) Camera 1, set A (b) Camera 1, set B (c) Camera 2, set A (d) Camera 2, set B

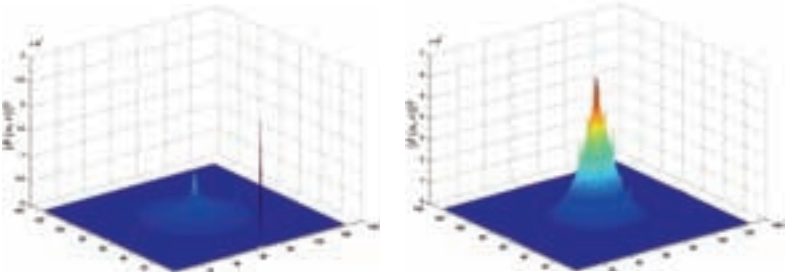
Figure 3: Multiple acquisitions of a single microstructure sample. Histogram equalization was used for visualisation purposes.

A justification for the assumption of correlated noise can be seen in Figure 4, where the spectra of the differences between acquisition images of an identical sample are shown. The non-uniformity of the spectra confirms the dominance of low frequencies, indicating a correlation between the elements of the noise.

In these experiments, the acquisition sets A and B were compared for both cameras to produce the results. The images have been pre-processed to render them zero-mean and unit variance to compensate for any variations in the lighting conditions, which could result in potentially varying means and dynamic ranges. The inner product $r(\mathbf{y})$ is therefore mathematically identical to the sample cross-correlation.

Figure 5b shows the results for when the camera 1 is used for both enrollment and verifica-

[1] http://sip.unige.ch/famos



(a) Camera 1 (b) Camera 2

Figure 4: Spectra of the noise showing correlation in the noise of the FAMOS dataset.

tion, whereas Figure 5c corresponds to the case where camera 2 is used for both enrollment and verification. The curves for $s(\mathbf{y})$ and $s(\tilde{\mathbf{y}})$ are absent in Figures 5b and 5c, which indicates that either p_m or p_f can be reduced to zero when the same camera is used for enrollment and verification when using the proposed statistics.

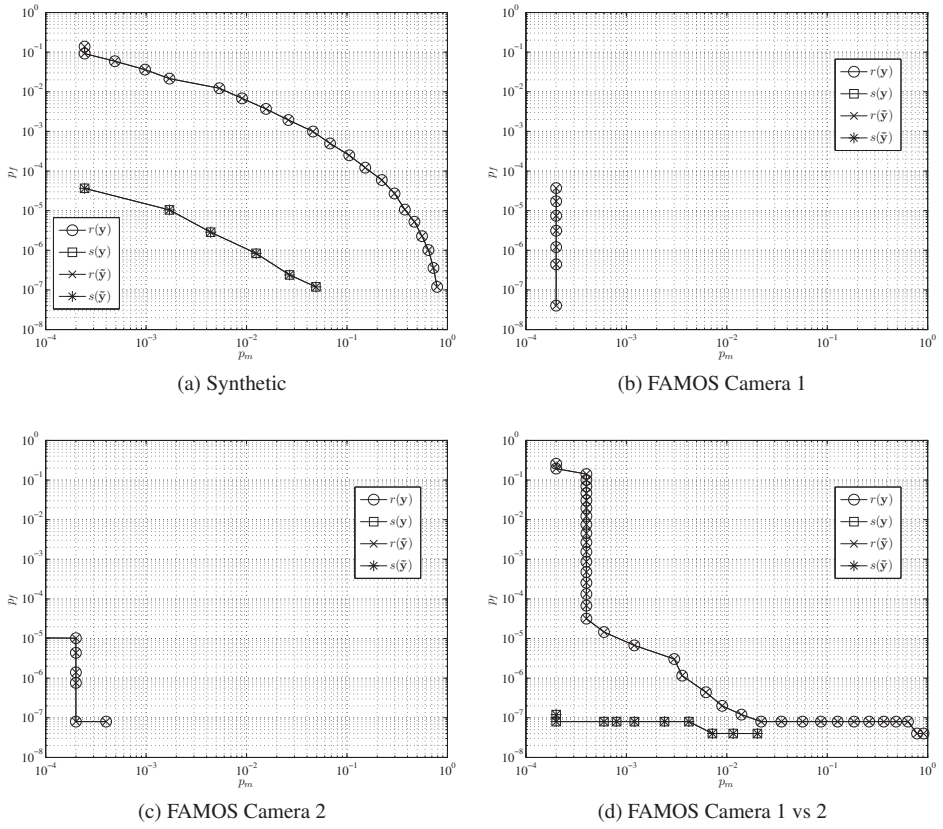
Figure 5d shows results for the case where camera 1 is used for enrollment, and camera 2 for verification. In this case, the performance is relatively decreased relative to the case where only one camera is involved, as can be seen from the corresponding ROC curves.

In all cases, the fact remains that deploying the proposed sufficient statistics $s(\mathbf{y})$ and $s(\tilde{\mathbf{y}})$ leads to a significant performance improvement over $r(\mathbf{y})$ and $r(\tilde{\mathbf{y}})$, respectively. Additionally, we see that the chosen transform does not impact the performance, neither for synthetic data or the true microstructure images from the FAMOS dataset.

4 Conclusions

In this work we have shown that a sufficient statistic based on a model with correlated noise leads to significantly better performance, on both synthetic data and on true microstructure images from the FAMOS dataset, in comparison to sufficient statistics for the case of i.i.d. additive white Gaussian noise.

There are a significant number of directions for future research. First, an important aspect of content fingerprinting that has not been addressed in this work is dimensionality reduction, which can be optimized with respect to robustness against distortions. Second, the role of quantization and sufficient statistics in the binary domain are of vital importance, as fingerprints are usually binary sequences. Third, the development and testing of improved matching techniques, specifically matching real the transformed data against binary fingerprints, can be explored. Fourth, the performance can most likely be improved even further by reconstructing $\tilde{\mathbf{X}}$ from a binary fingerprint and other information available at the detector. Furthermore, we aim to develop a thorough information-theoretical analysis based on the framework introduced in [Bee12]. Last, the results can be extented to the identification setup.



(a) Synthetic

(b) FAMOS Camera 1

(c) FAMOS Camera 2

(d) FAMOS Camera 1 vs 2

Figure 5: ROC curves of authentication on synthetic data and the FAMOS datasets.

Acknowledgment

This work is supported by SNF-grant 20021-132337 and the CRADA project between the University of Geneva and U-NICA systems.

References

[BCJ+05] James D. R. Buchanan, Russell P. Cowburn, Ana-Vanessa Jausovec, Dorothee Petit, Peter Seem, Gang Xiong, Del Atkinson, Kate Fenton, Dan A. Allwood, and Matthew T. Bryan. Forgery: 'Fingerprinting' documents and packaging. *Nature*, 436(7050):475–475, 2005.

[Bee12] Fokko Beekhof. *Physical Object Protection based on Digital Micro-structure Fingerprinting*. PhD thesis, University of Geneva, 2012.

[BL96] Robert D. Brandt and Feng Lin. Representations that uniquely characterize images modulo translation, rotation, and scaling. *Pattern Recognition Letters*, 17:1001–1015, 1996.

[FLL02] Benoit Macq Frèdèric Lefèbvre and Jean-Didier Legat. Radon soft hash algorithm. In *Proceedings of the European Signal Processing Conference*, Toulouse, France, September 2002.

[Fri00] J. Fridrich. Visual hash for oblivious watermarking. In P. W. Wong and E. J. Delp, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3971 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 286–294, May 2000.

[FVK10] Farzad Farhadzadeh, Sviatoslav Voloshynovskiy, and Oleksiy Koval. Performance Analysis of Identification System Based on Order Statistics List Decoder. In *IEEE International Symposium on Information Theory*, Austin, TX, June, 13–18 2010.

[Ign09] Tanya Ignatenko. *Secret-Key Rates and Privacy Leakage in Biometric Systems*. PhD thesis, Technical University of Eindhoven, 2009.

[Jai89] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[Kay98] Stephen Kay. *Fundamentals of Statistical Signal Processing, Vol II - Detection Theory*. Prentice-Hall, Inc., 1998.

[SMW06] Ashwin Swaminathan, Yinian Mao, and Min Wu. Robust and secure image hashing. *Information Forensics and Security, IEEE Transactions on*, 1(2):215 – 230, June 2006.

[TSE07] Pim Tuyls, Boris Skoric, and Tom Kevenaar (Eds.). *Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*. Springer, 2007.

[VDB+12] Sviatoslav Voloshynovskiy, Maurits Diephuis, Fokko Beekhof, Oleksiy Koval, and Bruno Keel. Towards Reproducible results in authentication based on physical non-cloneable functions: The Forensic Authentication Microstructure Optical Set (FAMOS). In *Proceedings of IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5 2012.

[VKB+10] S. Voloshynovskiy, O. Koval, F. Beekhof, F. Farhadzadeh, and T. Holotyak. Information-Theoretical Analysis of Private Content Identification. In *IEEE Information Theory Workshop, ITW2010*, Dublin, Ireland, August 30 – September 3 2010.

[VKP08] Sviatoslav Voloshynovskiy, Oleksiy Koval, and Thierry Pun. Multimodal authentication based on random projections and distributed coding. In *Proceedings of the 10th ACM Workshop on Multimedia & Security*, Oxford, UK, September 22–23 2008.

Proposal of a privacy-enhancing fingerprint capture for a decentralized police database system from a legal perspective using the example of Germany and the EU

Matthias Pocs[1], Mario Hildebrandt[2], Stefan Kiltz[2], Jana Dittmann[2]

[1] Stelar Security Technology Law Research UG (haftungsbeschränkt) c/o Fuchs, Fanny-Lewald-Ring 110, 21035 Hamburg
mp@stelar.de

[2]Research Group on Multimedia and Security, Otto-von-Guericke University, Universitaetsplatz 2, 39106 Magdeburg, Germany
{hildebrandt, kiltz, dittmann}@iti.cs.uni-magdeburg.de

Abstract: Innovations in biometric and forensic technology promise new use cases for the fight against crime and threats to public security. For example, the police will be able to use a new scanner to capture fingerprint traces from luggage at the airport to detect dangerous manipulations and identify known criminals. Despite these potentially great benefits, such systems also entail risks for society. One aspect of such systems is the biometric and forensic database used to compare fingerprints captured with a wanted list. This paper explores a possible decentralized database system as a solution to risks entailed by central systems. It uses the German and EU law as an example to justify technology design decisions on the basis of the legal requirements.

1 Introduction

The contact-less non-destructive acquisition and digital analysis of fingerprint traces, also known as digital dactyloscopy, creates new use cases. In this paper, a preventive use case at the airport serves as an example. Such use cases entail risks for individuals' privacy and society at large. We analyse the database aspect of such systems and a decentralized architecture as one possible solution to enhance privacy.

This paper is motivated by the privacy, policy and legal issues in biometric, forensic and security data. Only if the design of surveillance applications are strictly privacy-compliant the police will want to give orders to develop fingerprint systems for new use cases. Therefore this paper contributes to the improvement of privacy and data protection as well as the development of new database supported systems. This in turn extends the current police tool box because it means that additional data will be available for crime prevention.

This paper builds on a publication that explores the general legal requirement to use a decentralized database system [Poc11]. However that publication does not include the technical point of view. The transdisciplinary approach chosen in this paper ensures that both sides - lawyers and engineers - take each others' feedback into account. This is necessary for fairer legal regulation of privacy-enhancing technology design. Since this paper only looks at the database aspect, in Sections 3 and 4 other design aspects have to be integrated to assess the overall privacy impact. A second property of our approach is to distinguish general technical requirements (Section 4) from specific technical implementation (Section 5). This way technology is only required to follow design decisions as far as legally necessary, in this case the general technical approach (Section 4) is the legal requirement. The specific technical implementation (Section 5) shows examples of how to specify that general approach and are not binding. Nevertheless they are important because they enable technology producers to assess conformity with the legal requirement (for this approach see in detail [Poc12]).

The structure of the paper reflects the transdisciplinary approach we chose as lawyers and technologists. Section 2 outlines the technical use cases and their legal risks to derive to a design approach as a possible solution. In Section 3 this paper describes the state of the art from a technical and legal point of view. Whereas Section 4 gives the reader an idea of the general design approach, Section 5 specifies the technical implementation. Both sections build on legal arguments to justify design decisions. In the conclusion we sum up the technology design and its legal evaluation.

2 New use cases, risks and possible solution

In this paper we look at a use case that involves a fingerprint system used during luggage-handling at the airport [HDP+11]. It scans traces of fingerprints that are left on luggage. The police searches these traces on already known fingerprints from a database (of contact persons, criminals, or similar). In any case the captured data are strictly secured and deleted after the comparison. If there is a hit, the data are revealed for further investigation. This use case aims at gaining hints to detect terrorist and other criminal networks with histories of serious offences by means of identification of persons.

The use case offers opportunities of detecting dangerous persons suspected of serious crime. However, it also creates risks for the personality of individuals and society at large. In particular there are risks if one chooses to send all captured data to a central database system for comparison with a wanted list. In contrast to the IT security goals (confidentiality, integrity, authenticity, non-repudiation and availability), the main driver for this paper is the legal concept of privacy, which takes into account the following risks [MPD+13].

Sensitive data and identifiers. The capture of fingerprints and biological characteristics is already very risky for privacy. This is due to the fact that biological characteristics reveal unnecessary information about illnesses and ethnic origin (sensitive data). Additionally,

they are universal and life-long identifiers that can be misused to create a personality profile from databases and location data.

False hits. Furthermore there is a risk caused by the statistic errors involved in comparison of biological characteristics. In addition there is the risk of erroneous operation of the system by police officers. As soon as fingerprints are compared by the police with fingerprints of wanted persons, people could be confused with terrorists or other criminals.

Function creep. Moreover there is the risk of "function creep." The legislator can introduce the fingerprint system using the fight against serious crimes as a justification and subsequently allow its use for less important purposes. For example, ministers and senior police officers, who want to punish minor offences or take measures against non-criminals, can urge parliaments to allow this. The German AFIS contains fingerprints of three million people [Bun12] among which there are many that are not a terrorist or a similar criminal. The secretive capture of fingerprints even increases the risk of function creep.

Technology compliance. There is the risk that the police procure and use a noncompliant fingerprint system. There are no EU wide standards for security technologies [EDT09]. Therefore there is an increased need for checking whether the system design reducing the privacy impact is on paper only or a reality.

"Identity theft." In addition, there is the risk of "identity theft." A minister or senior police officer could "attack" the fingerprint system in order to achieve a successful result of a search, that is, use the system in a way that is not authorized by the law. Even such illegitimate aims, the law needs to consider ([PoS76], 208). They could use databases to track political activists, discriminate against ethnic groups and less important purposes without legal authorization.

Follow-up measures. Besides, there is the risk of being exposed to a follow-up police measure. This is due to fact that fingerprint traces are captured and made available to the police which were not before. Citizens can be exposed to police measures due to the risks of function creep, identity theft and false hits. They may feel like being watched and abstain from deviant behavior, which harms society at large. The fact that a large number of innocent citizens are involved due to the large "scatter" of the data processing increases all of the risks mentioned above.

A possible solution to these risks (in connection with other privacy-enhancing measures) is the approach of a decentralized database architecture where the captured data are compared within a local subsystem attached to the capture device. Only in cases of a confirmed hit they are re-examined in a central system. For such a decentralized database system, one needs to reduce the amount of data that is necessary for the performance on small IT systems.

3 State of the art and legal requirements

From a technical point of view we need to assess the state of the art of biometric and forensic systems. In biometric systems we can use one-to-many identification or one-to-one verification to identify fingerprints or similar [Viel06]. The biometric community has also explored techniques to protect biometric templates (standardized in ISO/IEC 24745:2011). However, the technical possibilities for forensic use cases are limited because until recently the community primarily considered use cases with controlled situations. That is, the system captures fingerprints or similar directly from the individual or a representation of such a direct collection. This way the quality is sufficient to properly carry out identifications and verifications.

However in forensic use cases the original fingerprint is not available. Instead the system captures latent fingerprints / fingerprint traces. In this case the fingerprint quality is usually low because the fingerprints might be only partial, smeared, distorted or similar. Thus, other matching strategies need to be applied.

One potential technique to protect the privacy of biometric data as well as taking into account the variations in the data is the application of biometric hash functions [SSM05, TFM+07]. Unlike cryptographic hash functions such techniques are designed to generate similar hashes for similar data representation, similar to the piecewise hashing outlined in [BaB11], and thus, allowing for a matching in the hashed. In [SSM05] such an authentication scheme is evaluated for the example of facial images. This particular approach combines a user-dependant one-way transformation and secure hashing algorithms. Another approach is introduced by Tulyakov et al. [TFM+07] for the example of fingerprints. This technique uses symmetric hash functions to compute biometric hashes for sets of minutia points (see e.g. [HRL11]), which results in a set of hashes for each fingerprint. The symmetric hash function has the advantage that all elements are equally weighted. Hence, the hash is not depending on the order of elements. Furthermore, each minutia is represented by a complex number with additional factors to address potential differences in rotation and translation of the fingerprint. During the matching process the hash sets are compared with each other while retaining the matches with the highest confidences. This technique might also help to address the issue of partial and smeared latent fingerprints. Thus, it is considered in our concept in Section 4. However, the potential for reverse-engineering biometric traits, such as described by Kümmel et al. [KuV10], needs to be analyzed.

As mentioned in Section 1 we cannot regard the approach of a decentralized database in isolation to improve privacy protection. We also have to consider other privacy-enhancing design approaches. New basic technologies such as the so-called “coarse scan” and “aging” promise to minimize the number of fingerprints captured in order to spare innocent people police investigations [DVU+12] [MPD+13]. Another design approach, the “Two-Offices Principle,” employs a double encryption technique [PSH12]. This ensures that the police can only use fingerprints captured when they indicate a threat to public security because they may belong to a known dangerous person. A third design aspect is the proper error management. Besides an automated matching, which is prone to errors especially for low quality fingerprint data, there are manual approaches.

In the case of the forensic analysis of fingerprints we employ a four staged investigation technique [HRL11]. This ACE-V methodology consists of the analysis, comparison, evaluation and verification steps. In order to achieve high standard of matching decisions two forensic experts follow this methodology independently of each other.

In addition to the technical assessment we have to explore the specific legal privacy requirements for new surveillance applications using digital fingerprint systems.

System suitability and effectiveness. The first privacy requirement is that the system needs to be suitable to achieve the stated purpose, which is, identifying criminals. One has to consider the system performance and economic reasonability. This means that the hits and fingerprints have to be available to the police when they need them. In order to be effective, the system needs to reduce false hits and help the police do so [MPD+13].

Use limitation. Another privacy requirement is that the system needs to limit the data use to the stated purpose of the fight against crime. The system should prevent the police from exploiting health and ethnic data from fingerprints [Art03] (see Article 8 of the European Data Protection Directive [Dir95]). The system should also prevent the police from exploiting fingerprints as uniform identifiers for personality profiles. This is prohibited by the *Bundesverfassungsgericht* (German Federal Constitutional Court).

To prevent unnecessary exploitation, the system should also erase non-hits instantly and securely. This requirement follows the *Bundesverfassungsgericht,* which rules out a privacy impact for non-hits that are not communicated to a police officer and instantly erased. In order to prevent data disclosure for secondary use, one has to ensure that only the police department in charge can use the data. This "separation of informational powers" is required by the *Bundesverfassungsgericht*.

Data security. The system needs to be secured by technological means against identity theft by considering user rights management, secure communication and state of the art cryptography. This is also a new requirement of Article 27 of the Directive Proposal for Police Data Protection [Com12a].

Transparency. Another privacy requirement is transparency. The system needs to enable courts and supervisory authorities to understand when and which police department used certain data in the past. It means that the fingerprint scanning system needs to be able to log when and where fingerprints are scanned and who collects them.

Accountability. System users need to demonstrate that the technology design required by the law corresponds with its actual realization. They have to take measures for compliance with the privacy requirements and implement mechanisms for auditors to verify the effectiveness of these measures. This is a new requirement of Article 22 of the Data Protection Regulation Proposal [Com12b]. This means that system users need to prepare source codes and documentation of programs, tools and hardware.

Data minimization and "data frugality." The system should avoid the collection of unneeded personal data or pseudo-/anonymize them. This requirement of data minimization, more precisely, "data frugality" is defined in § 3a of the *Bundesdatenschutzgesetz* (German Federal Data Protection Act). This means that the

system should use anonymous data instead of personal data and reduce the number of data categories and data retention period.

Distinction between individuals. Another privacy requirement is to distinguish criminals from innocent citizens. One has to distinguish between people having committed minor offences and serious crimes, between suspects and non-suspects as well as suspicion on the basis of mere assumptions and those based on facts. The latter two distinctions are new requirements of Articles 5 and 6 of the Directive Proposal for Police Data Protection [Com12a]. The system should help the police focus on serious criminals and spare any other person, already known to the police, follow-up measures.

Avoidance of large scatter. One has to avoid that a large number of persons is exposed to the system. This avoidance of a large "scatter" of data processing is required by the *Bundesverfassungsgericht*. This means that the system should reduce the scatter of data capture and use [HDP+11].

4 General design approach and improvement of privacy protection

For the use case mentioned in the introduction, we propose the general design approach of a decentralized database system with the setup shown in Figure 1.

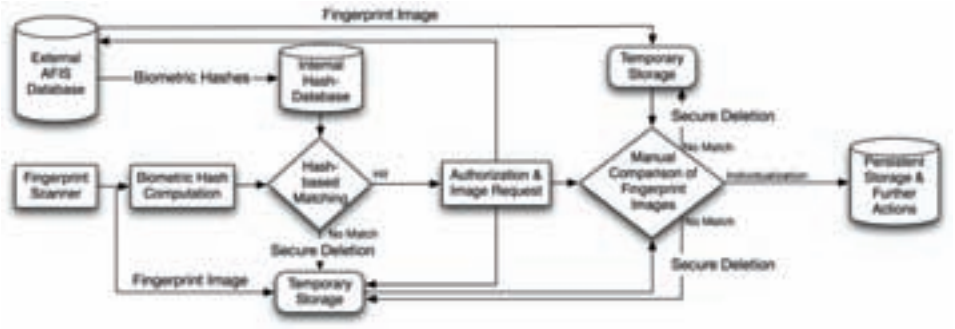


Figure 1: General design approach of a decentralized forensic fingerprint matching system

The system scans latent fingerprints at the airport using a contact-less scanner and computes the corresponding biometric hash (in line with ISO/IEC 24745:2011 (V60.60, 17.6.2011)). In parallel, the fingerprint image is stored within a temporary storage, which meets the demands regarding the security aspects of integrity, authenticity and confidentiality. Furthermore, a small internal database with biometric hashes is created from a main central database system such as AFIS (see [HRL11]). The local database stores only selected biometric hashes from the external AFIS database that relate to people having committed serious crimes or caused threats to someone's life or the state.

This small database is searched by comparing stored biometric hashes with the computed hash from the captured fingerprint. This leads to an automatic pre-selection of data sets by means of *multiple verifications of biometric data* as opposed to a full identification against a large database. In contrast to AFIS, which has the goal of a low false non-

match rate (FNMR), a low false match rate (FMR) should be achieved by this system to avoid mis-identifications. If no matching biometric is found within the local database, the fingerprint image in the temporary storage is securely deleted (see e.g. [Gut96]).

If the system finds a "hit" in the small database, this results in granting the forensic expert the permission to retrieve the image of the fingerprint from the data captured at the airport and to receive the reference image from the central database (e.g. AFIS). The expert does not have access to the entire database but only one set of fingerprint images that has been identified using the fully automated pre-selection of the data. The necessary fingerprint images for the matching samples are automatically requested from the central database. Hence, the fingerprint image from the external database needs to be stored in a temporary storage. Thus, at this stage, only an individual set of fingerprint images that indicate a threat to public security are stored.

The general approach of a decentralized database can be integrated into the overall set of privacy-enhancing design approaches. Concerning the basic technologies "coarse scan" and "aging" there is no conflict with the decentralized approach. The detailed scan necessary for the decentralized matching is independent of the "coarse scan" and "aging", which aims to preselect the fingerprints to be captured. The "Two-Offices Principle" can also coexist with the decentralized database system. These privacy protections even strengthen each other because the fingerprint images can be secured in a way that the system only releases them if a trusted third party has checked that there is a reason to do so. Also the decentralized database adds to the proper error management. After the captured fingerprint images are released to the forensic expert, a manual comparison is performed by a forensic expert. If the expert has the opinion that the fingerprints are not matching (exclusion), both fingerprint images are securely deleted from the temporary storage. If the fingerprints are matching (individualization), the fingerprints are transferred to a persistent storage and further actions are performed, e.g., arresting of the suspect and transfer of the data to central law enforcement agencies.

The implementation of the general design approach of a decentralized database improves privacy in several ways. This is because it promotes the legal requirements mentioned in Section 3.

System suitability and effectiveness. With the decentralized database the system does not lose its suitability to achieve the purpose of identifying criminals. The use of biometric hashes enables even small IT systems to perform well. The decentralized database even improves the data availability for local police departments who are in charge. However, reducing the calculation time through the use of biometric hashes can also lead to an increase in false hits caused by collisions of the hashes. However the decentralized approach also means that there is an additional round of manual re-examination by forensic experts at the local database system. Hence the false hit rate is decreased for the entire system.

Use limitation. The decentralized database ensures that the police only use the system for the purpose of the fight against crime. Exploiting health and ethnic data as well as uniform identifiers can be better prevented in isolated local IT systems that only serve a specific purpose than in a large-scale central system with multiple purposes and multiple

users. Also the instant and secure data erasure is only possible in such systems. Moreover using decentralized databases the system can ensure that only the police department that is in charge can use the data (separation of informational powers).

Data security. The decentralized database improves data security. In comparison to a multi-purpose system like a national AFIS, a local system with a more specific purpose can better ensure correct user rights management. With the decentralized database the local system does not communicate the large number of data about innocent citizens to a central system. This saves costs for the securing of communication channels and the stored data. However the choice of multiple isolated local systems increase the costs for keeping the encryption algorithms up-to-date.

Another advantage for data security is the use of the improved possibility to delete unneeded data. Both the captured fingerprint images and the reference fingerprint images are stored in a local temporary storage. Only if the forensic experts confirm the matching of fingerprints in a manual examination procedure, they are transferred into a persistent storage. In doing so, only a few fingerprints are stored within the database and the challenge of a secure deletion within the local database is avoided.

Transparency. The use of a decentralized database improves transparency because it only works if the data exchange between police departments is properly logged. Otherwise the link between the references of the local and the central system would be lost and this would disable the police to take measures against a suspect.

Accountability. The current use cases of central database systems like AFIS have a lesser privacy impact than the new use cases at the airport. This is why there are stricter requirements for the preparation of program source code and system documentation. If one chooses to use the central system, there is the risk that the police only refer to existing documentation prepared under less strict requirements. In case of a decentralized database system the police cannot do so. They are more likely take the accountability requirements seriously.

Data minimization and "data frugality." The use of biometric hashes can anonymize the fingerprints captured. It can also reduce data categories by using an index data system for the decentralized database. Only after associating the fingerprints with the original database the police can reveal more information about the wanted person.

Distinction between individuals. With the decentralized approach one can better ensure distinction of criminals from innocent citizens. Assuming that the number of wanted persons is much smaller for the new use case at the scenario any large database is likely to include other people too. In a local system for a specific purpose the performance is limited and one can limit the storage capabilities. This way one can ensure that the police only use a small database focusing on serious criminals.

Avoidance of high scatter. The main privacy improvement is that the decentralized database enables the system to preselect relevant fingerprints, which reduces the "scatter" of the data processing to a minimum.

5 Specific technical implementation and legal evaluation

In order to implement the general approach of a decentralized database system we have to address a number of measures. These measures include the implementation of the decentralized database approach. We explore several aspects of it: the type of comparison and generation of biometric hashes, connection between the central and decentralized system, user rights management, design of the local devices and central system as well as encryption of data and communication. In addition we look at the integration of the decentralized approach into the entire system including a backup strategy. Whenever a certain design decision is legally necessary, we refer to the legal requirements described in the previous section.

In contrast to other fingerprint identification systems such as AFIS, the "identification" is performed using multiple verifications (one-to-one comparison) and no ranked list is displayed. This is necessary, because ranked lists need to be verified by experts manually, which would cause a lot of work at the decentralized system. Furthermore, a list with 15 candidates contains at least 14 mis-identified samples. In a one-to-many comparison all samples with a limited similarity are selected, which are used to create the ranked list (see e.g. [JFN+12]). Usually, such approaches include special index structures to avoid the more time consuming one-to-one comparison. Such an approach would lead to a high scatter, which should be avoided within a preventive system.

The system we propose performs the comparison using biometric hashes to reduce the amount of data which is necessary for the performance on small IT systems. The local database contains one table, "reference_data", with biometric hashes as primary keys and corresponding authorization codes to request fingerprint images from the external database. However, we can also identify several limitations of using techniques of biometric hash, in particular, the false match rates. Whereas the goal of AFIS is a low false non-match rate (accidently missing the matching trace), the goal of this system needs to be a low false match rate for *avoidance of high scatter*. For *system suitability and effectiveness* we need to strike a balance. The use of biometric hashes has to provide the performance needed for small IT systems but also must ensure a low number of false hits. Reducing the information in hashes also serves other legal requirements. It improves *use limitation* since it removes sensitive data. For *data security* it is less likely to be able to reconstruct the original fingerprint image from a hash with reduced information. For *data minimization and "data frugality"* they are more likely to be considered anonymous data.

In particular the system architecture needs to rule out the possibility to reconstruct the original fingerprint image as reported e.g. by Cappelli et al. [CML+07] for the reconstruction from biometric templates. In the recent years, many researchers analyzed the reversibility of several fingerprinting and robust hashing algorithms. The hashes used in our system architecture should be robust to a so-called reversibility attack and spoofing to avoid attacks as described e.g. in [KuV10] for a biometric hash for handwriting. In general such reconstruction approaches lead to samples that are likely to be matched using automated systems. However, the risk of a reversibility attack is not

high because for human experts several differences in the fingerprint trace and reference are visible.

There also needs to be a link between the biometric hash and the image representation of the fingerprint. Since the local database only contains biometric hashes and authorization codes that associate data sets with those in the central database, the system is restricted to the minimum amount of information. Such an index data architecture meets the requirement of *system suitability and effectiveness*. If the local reference database reveals all information about wanted persons, criminals can use that information to jeopardize the purpose of the fingerprint system. In case of index data the local database does not reveal the fingerprint image or biographic information, only the biometric hash and an index number. The index data approach also meets the *transparency* requirement. If forensic experts follow up on a hit using the full fingerprint image, they need to request the reference fingerprint image from the central system. This way another police department obtains knowledge about the fact that the police department at the airport found a hit. For *data minimization and "data frugality"* the index data means that the system reduces data categories to the category of hashes and index data.

The user rights management for the database has to ensure that forensic experts can only access the fingerprint images in case of a hit. Several databases need to be secured from unauthorized use. The access to the "reference_data" database has to be restricted in such a manner that only the biometric hashes are accessible using a special database view for the user. In addition the local fingerprint scanner captures fingerprint images. For each image, a biometric hash is computed. A second table, "capture_data", needs to store the biometric hashes generated from the local fingerprint scanner, as well as meta data and a reference to the fingerprint image within the temporary storage. This database also needs to be secured from unauthorized access.

In a local system one can better control the storage capabilities. The size of the temporary storage can be limited in a way that the "reference_data" does not exceed a certain population size. This way only a small selection of wanted persons can be detected and a routine identification of a large number of citizens is avoided. For *distinction between individuals,* this is beneficial because only the most important criminals should be addressed [Poc10]. Otherwise a large reference database could be used, which, does not sufficiently distinguish people causing serious threats from others such as victims, witnesses and contact persons.

For the temporary storage devices appropriate deletion mechanisms need to be used. Such techniques are dependant on the utilized storage concept. In general the stored data should be overwritten with new data. Thus, storage media with wear levelling mechanisms, such as solid state disks should be avoided because of the uncertainty with which such devices manage deleted data [BeB10]. Another option is the encryption of each stored fingerprint image using an individual key. If the key is deleted from the storage, the image data is rendered inaccessible. The key itself must be overwritten but this is much faster compared to the deletion of the entire fingerprint image due to the limited amount of data of the key.

In order to create the table “reference_data” the system needs to download the selected biometric hashes in regular time periods from the central system. Whereas there needs to be a connection between the central and the local system, the data should only flow in one direction. One must not be able to upload the data captured at the airport from the decentralized database to the central system (read only) or pull those data from the central system. Otherwise the legal advantage of the general decentralized design approach is lost. The restriction to read-only access can be easily achieved using rights management. The connected user of the local system should gain access to a limited view which allows for accessing one particular image that corresponds to the authorization code from the local database “reference_data”. In addition this should be ensured by restricting the list of references in the central system too. Just like the limitation to a selection of most important criminals in the local system’s table “reference_data” mentioned above, a list of potential hashes should be provided within the central database to limit the amount of accessible data to a list of most important targets.

We have to look at the need to encrypt the database and communication channels. Each captured fingerprint image in the temporary storage can be encrypted using an individual key. In this case the key needs to be stored within the database as well. The temporary storage can be a separate database or a special file system. Either way a sophisticated access restriction and secure deletion mechanisms are necessary. For *use limitation* such additional encryption can improve secure data erasure. For *data security* it can prevent “identity theft.” Although there is little information included in the biometric hash, the need for securing the database is still urgent because the system also stores the fingerprint images. This is different for the securing of the communication channels between the central and the local system. In relation to *data security* the local system does not communicate the large number of data about innocent citizens to a central system. The need to secure the communication channels is less urgent. Regarding the database encryption, for *data security* it is required to keep the encryption algorithms up-to-date. The communication channels within the local system, as well as the communication channel between the local system and the central database need to be encrypted with state-of-the-art algorithms, too. Furthermore, each communication partner should be authenticated to avoid any unauthorized access to the data. This also applies for the system that is used for the manual comparison since it needs to display both images. The displayed data must be deleted securely from any storage devices. The disclosure of data has to be limited by organizational means (e.g., avoiding the displays to be photographed) and by technical means (e.g., disabling USB ports).

The measures for the decentralized database can be integrated into the other privacy-enhancing design approaches, most notably, the proper error management in case of a match. The automated matching of the biometric hashes is performed using a stored procedure, which executes the comparison during the insert query. If a matching pair of biometric hashes is found in the database, the authorization code of the matching fingerprint is used to request the corresponding fingerprint image from the external database. This image is stored in another temporary storage device. Furthermore, in a third table, “investigation_data”, the matching pair of biometric hashes is stored, accompanied by the references to the fingerprint images within the temporary storages.

Additionally, an examiner is alerted that a matching pair of fingerprints is found, which requires a manual investigation. The examiner can access the relevant data from the table "investigaton_data" containing the references to the fingerprint images, as well as necessary encryption keys. In addition the system has to generate secure log files (with regard to the security aspects of integrity and authenticity) that enable supervisory authorities to check the police measures. If the examiner concludes that the images are matching, the data is stored within a separate database to allow for further actions. If no match is found, the fingerprint images in both temporary storages are securely deleted and the status of the investigation in the table "investigation_data" is deleted. If the database contains any encryption keys, a secure deletion of such keys is also necessary.

With this step the local investigation is completed. If the fingerprints are matched, there is a reasonable suspicion to continue a forensic investigation. This involves law enforcement agencies and thus, the transfer of the collected data. To ensure the validity of the data as potential evidence, it is necessary to fulfill all requirements regarding the security aspects of integrity and authenticity. Furthermore, the local expert should have a proper training similar to forensic experts at the forensic laboratories, to avoid mis-identifications. This also includes the documentation of the analysis according to forensic standards.

Another factor is the application of backup and recovery strategies. Since the reference hashes can be obtained from the central database, no backup is required. Furthermore, biometric hashes from acquired fingerprints should be stored temporarily, thus, those data do not require a backup, either. If the biometric hash based identification resulted in a hit, there is a vital interest of preserving the data until the investigation is finished. Thus, the table "investigation_data" as well as the temporary storages need to employ backup strategies. Since the match is not confirmed at this stage of the investigation, measures such as redundant storage, e.g., redundant arrays of independent disk (RAID) devices, and protection against external influence factors, e.g., the application of uninterruptible power supplies and shock absorbers, should be used. Further techniques should be used after the individualization of the fingerprint. Here, the data needs to be protected until it is handed over to the law enforcement agency. This could include the storage on WORM (write once, read many) media. Such media can be also used for the transfer to the law enforcement agencies. The suggested backup strategies and their implementation however have to meet the legal requirements, e.g., the sensitive data and identifiers (see Section 2) must be securely deleted (see Section 4). This applies either if the suggested hit is a false match or the earmarked and justified time of data retention is reached.

The entire design needs to be evaluated towards its feasibility. It is necessary that potential collisions of the biometric hash do not lead to mis-identifications. Furthermore, such collisions should only occur rarely because otherwise an increased amount of manual analysis time would be necessary.

6 Conclusions and future work

In this paper we explored how technology design decisions can improve the legal privacy requirements in fingerprint systems for new surveillance use cases. For their database aspect we demonstrated that it is legally preferable to take a decentralized database approach. In addition, we discussed specific decisions of technical implementation using legal arguments. Technology producers can use the descriptions of this paper for the documentation of program source code, tools and hardware. This way the paper also helps the police and other users comply with the legal requirement of accountability.

We integrated the database requirements into the other elements of the system so that legislators can assess the overall privacy impact of future digital dactyloscopy. In addition, we distinguished the general approach of a decentralized database from the specific measures. This way legislators can draw a clear dividing line between legal obligations and legal incentives. If technology producers take the specific measures we developed, conformity is presumed, but they can also develop more innovative measures to fulfill the general requirement.

The local database system explored in this paper contributes to a future digital dactyloscopy that reduces the privacy impact for a large number of innocent citizens at the airport.

In future work the performance of biometric hashes for the verification and identification of biometric traits in forensics should be analyzed. Furthermore, potential weaknesses of such approaches towards the reconstruction of biometric data, as well as collisions of hashes should be investigated.

References

[Art03] Article 29 Working Party on Data Protection (European Board of Data Protection Authorities): Opinion on Biometrics (WP83), EU Publications Brussels, 2003, no. 3.7.

[BaB11] Baier, H.; Breitinger, F.: Security Aspects of Piecewise Hashing in Computer Forensics. In 6th International Conference on IT Security Incident Management and IT Forensics (IMF), Stuttgart, Germany, IEEE Computer Society, ISBN 978-1-4577-0146-7, DOI=10.1109/IMF.2011.16, 2011, pp. 21-36.

[BeB10] Bell, G.; Boddington, R.: Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery? In Journal of Digital Forensics, Security and Law, Vol 5. No. 3, 2010, pp. 1-20.

[Bun12] Bundeskriminalamt (German Federal Criminal Police Office): http://www.bka.de/nn_227308/DE/ThemenABisZ/Erkennungsdienst/Daktyloskopie/AFIS/afis__node.html?__nnn=true, Wiesbaden, 2012.

[Com12a] European Commission: Proposal for a Directive of the European Parliament and of the Council on data protection in the police sector (COM(2012)10 final), EU Publications Brussels, 2012.

[Com12b] European Commission: Proposal for a General Regulation of the European Parliament and of the Council on data protection (COM(2012)11 final), EU Publications Brussels, 2012.

[CML+07] Cappelli, R.; Maio, D.; Lumini, A.; Maltoni, D.: Fingerprint Image Reconstruction from Standard Templates. In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 9, 2007, pp. 1489-1503.

[Dir95] Directive 95/46/EC of the European Parliament and of the Council on data protection. Official Journal of the European Union L 281, 1995, p. 31.

[DVU+12] Dittmann, J.; Vielhauer, C.; Ulrich, M.; Pocs, M.: Fingerspuren in der Tatortforensik. In digma - Zeitschrift für Datenrecht und Informationssicherheit, 2012, p. 80-83.

[EDT09] ECORYS NL; DECISION Études et Conseil; TNO: Study for European Commission on the Competitiveness of the EU security industry, EU Publications Brussels, 2009.

[Gut96] Gutmann, P.: Secure Deletion of Data from Magnetic and Solid-State Memory. In Proc. 6th Usenix Security Symposium, San Jose, California, USA, USENIX Association, 1996, pp. 77-95.

[HDP+11] Hildebrandt, M.; Dittmann, J.; Pocs, M.; Ulrich, M.; Merkel, R.; Fries, T.: Privacy preserving challenges - New Design Aspects for Latent Fingerprint Detection Systems with contact-less Sensors for Future Preventive Applications in Airport Luggage Handling. In Proc. BioID 2011, Springer Lecture Notes on Computer Sciences (LNCS) Vol. 6583, Springer, 2011, p. 286.

[HRL11] Holder, E.H.; Robinson, L.O.; Laub, J.H.: The Fingerprint Sourcebook, U.S. DoJ, Office for Justice Programs, 2011, Chapter 6.

[KuV10] Kümmel, K.; Vielhauer, C.: Reverse-engineer methods on a biometric hash algorithm for dynamic handwriting. In Proc. 12th ACM workshop on Multimedia and security, 2010, pp. 67-72.

[MPD+13] Merkel, R.; Pocs, M.; Dittmann, J.; Vielhauer, C.: Proposal of Non-Invasive Fingerprint Age Determination to Improve Data Privacy Management in Police Work from a Legal Perspective using the Example of Germany. In (Garcia-Alfaro, N. et al. eds.) Proc. 7th International Workshop Data Privacy Management (DPM 2012), Pisa, Italy, 2012, Lecture Notes in Computer Science, Vol. 7122, 2013 (forthcoming).

[Poc10] Pocs, M.: Gestaltung von Fahndungsdateien - Verfassungsverträglichkeit biometrischer Systeme. In Datenschutz und Datensicherheit (DuD), 2010, p. 163-168.

[Poc11] Pocs, M.: Abgleich im Erfassungsgerät. In (Schartner, P.; Taeger, J. eds.): Proc. D-A-CH Security, Oldenburg, 2011, syssec, 2011; p. 358.

[Poc12] Pocs, M.: Will the European Commission be able to standardise legal technology design without a legal method? In Computer Law & Security Review, 2012 (forthcoming).

[PoS76] Podlech, A.; Steinmüller, W.: Informationsrecht und Informationspolitik, Oldenbourg Verlag, 1976, p. 211.

[PSH12] Pocs, M.; Schott, M.; Hildebrandt, M.: Legally compatible design of digital dactyloscopy in future surveillance scenarios. In (Schelkens, H. et al. eds.) Proc. Optics, Photonics and Digital Technologies for Multimedia Applications, SPIE Photonics 8436, Brussels, 2012, p. 84360Z.

[SSM05] Sutcu, Y.; Sencar, H.T.; Memon, N.: A Secure Biometric Authentication Scheme Based on Robust Hashing. In Proc. 7th ACM workshop on Multimedia and Security, 2005, pp. 111-116.

[TFM+07] Tulyakov, S.; Farooq, F.; Mansukhani, P.; Govindaraju, V.: Symmetric hash functions for secure fingerprint biometric systems. In Pattern Recognition Letters 28, 2007, pp. 2427-2436.

[Vie06] Vielhauer, C.: Biometric User Authentication for IT-Security - From Fundamentals to Handwriting. In Advances in Information Security, Springer Science+Business Media Inc., ISBN 0-387-26194-X, 2006.

Database and Data Management Requirements for Equalization of Contactless Acquired Traces for Forensic Purposes

Stefan Kirst, Martin Schäler

Otto-von-Guericke University of Magdeburg
Universitätsplatz 2
39106 Magdeburg
stefan.kirst@iti.cs.uni-magdeburg.de
martin.schaeler@iti.cs.uni-magdeburg.de

Abstract: The importance of fingerprints and microtraces within the field of criminalistics and forensics is well known. To increases probative values of such evidences several techniques of lifting, enhancement and feature extraction exist. An upcoming field is the contactless acquisition of traces, because the integrity of traces is preserved. Hence, dealing with digital representations of traces is challenging because of the amount of data and their complexity. A further issue from such an acquisition method is the potential presence of perspective distortions, which we already started to deal with in [KCDV12]. Within the scope of a productive use of contactless acquisition methods, pre-processing steps like the equalization come along. In this short motivational paper we want to give a perspective on requirements for an underlying database and database management system to support the methods of [KCDV12] as a potential real-case scenario. Thereby, we point out possible starting points for parallelization potential and frequent queries, especially when using filter masks.

1 Introduction

In criminalistics fingerprints and microtraces are important evidence but the traditional way of capturing them, like the usage of brush and powder, alters the trace. A solution are contactless methods, that assure a non-invasive and repeatable acquisition. However, these processes struggle with distortions due to the perspective of the sensor or non-planar surfaces. Variations of features like the relative position of minutiae, ridgeline distances or length and width of microtraces are possible changes due to perspective distortion. As an outcome, algorithms based on affected features may not work properly. There are non-distorting techniques like the non diffractive element based approach for smooth curved surfaces in [KM09], but those methods are only applicable on specially shaped surfaces. Our recent achievements [KCDV12] in equalizations of such distortions using confocal microscopy[1] demonstrate the resolvability of this challenge. After evaluating our approach

[1] for all scans the confocal laser microscope "'Keyence VK-X110"' [Key] was used; intensity and topography data were used for equalization

on different surfaces and shapes, we accomplished to use our methods on fingerprints and microtraces[2]. Significant reduction of a perspective distortion induced error in equalized images could be presented.
Providing information of detected distortions as well as equalized images we are able to support a forensic investigator's work. Since an equalization may take up to ten minutes[3] a productive application of these techniques in productive scenarios might be exhausting as crime scenes with a single trace seem to be unusual. The delays in all measurements scale when using faster systems, but will not dissolve entirely.
So far our recent work on equalization of non-planar surfaces is concentrated on qualitative results. In this paper aspects of database- and database management systems are motivated using the outlined scenario. Based on experiences of our first exemplary evaluations of our new methods, we highlight the potential and challenges that arise from a database point of view, when it comes to performance issues in the productive use of forensic methods.
After a short summary of our equalization methods we concentrate on the following points of interest:

- potential of parallelization for inter- and intra trace computations
- frequent requests, under the perspective of filter masks

2 Equalization of forensic traces on non-planar surfaces - an application scenario

The following presentation of the basic background is necessary for later explanations on amount of data and suggestions to improve our approach. Besides, the conclusions of this paper rely on our experience on methods for equalization. Hence, a general knowledge of our approach and its capabilities is mandatory.

In [KCDV12], we dealt with perspective distortions as they arise in contactless scans of non-planar or non-perpendicular placed surfaces to preserve topological features. These distortions alter the topology[4] of traces, so that topology based processing methods, like NBIS[5] [WGT+] or ridge density based sex determination [Gun07], might produce unreliable results. The presented approach divides an inhomogenously shaped surface into small quasi planar surfaces. After equalizing them separately, we try to reassemble them according to their previous topology. The slopes are determined by analyzing the topography data in a blockwise manner. We were able to adapt these techniques on fingerprints and microtraces. For evaluation purposes we introduced a landmark based relative error to show the impact of our introduced pre-processing methods. This error describes the distance of fixed points, called landmarks, in comparison to the very same points in an undistorted image. First evaluations of our approach indicate significant reductions of this

[2] a partial human hair, see figure 2 and table 1
[3] equalization of a fingerprint(20 degrees, area: 14,26mm x 9,49mm): 10.30 minutes - using: Intel(R) Core(TM) i7-2670QM @ 3.1GHz, 8GB RAM; javaVM @ 4GB, WDC WD6400BPVT-60HXZ
[4] e.g. ridgeline distance, relative position of minutiae
[5] "'NIST Biometric Image Software"' - see [online]: http://www.nist.gov/itl/iad/ig/nbis.cfm

relative error for all scans after the equalization. Due to reasons of space, we are not able to present more information on the equalization process, as well as on the evaluation procedure, and kindly refer to our previous work [KCDV12].

At first, we applied our approach on different types of surfaces[6] before we used it on combination of traces (see table 1).

type of trace	test object	number of scans	area
latent fingerprint right pointing finger	platter	4 scans (angles: 0, 20, 40, 60)	14266,36µm x 10972,92µm
partial human hair	quasi-planar application on platter	4 scans (angles: 0, 20, 40, 60)	1346,73µm x 4333,62 µm

Table 1: test objects: surface with traces [KCDV12]

The results for the equalization of fingerprints based on a landmark based relative error were significant[7] (see figure 1).

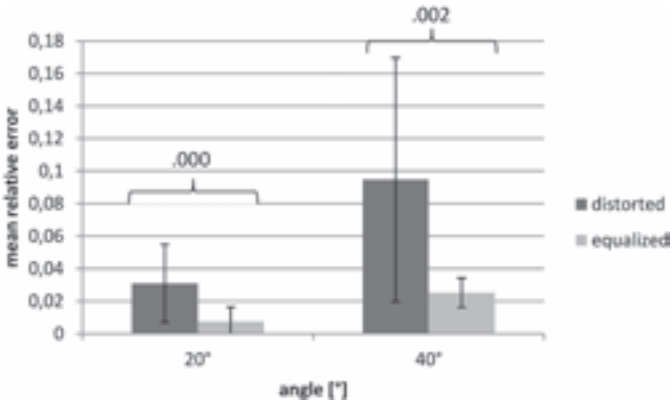


Figure 1: results for the relative error in scans of a fingerprint in different angles - minutiae-based set of landmarks [KCDV12]

The application of our approach on microtraces is promising as well. It should be noted, that all equalizations on this microtrace were calculated using only the surface of the human hair itself, which is a quiet challenging approach regarding to its rough and flaky surface. We were also able to reduce the relative error distinctly for the human hair[8] (see figure 2).

[6] a planar surface in different angles, two curved surfaces, a spherical surface - see [KCDV12]
[7] significances are shown in curly braces over each bar in the diagram
[8] no standard deviation or significance could be calculated

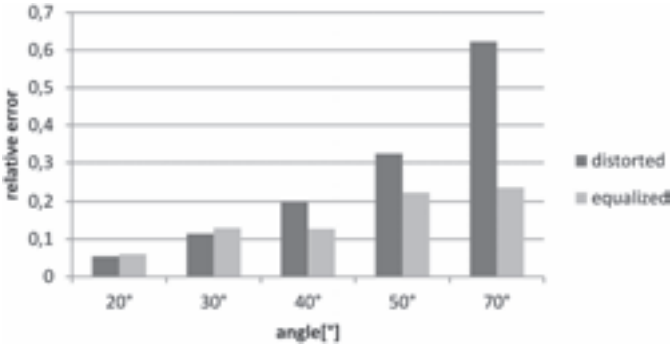


Figure 2: equalization of a human hair on a platter using its length as a metric [KCDV12]

By recovering the topology of surfaces and applied traces we can support an investigators work with additional useful information. Despite long acquisition times[9], the equalization of a surface takes an inconvenient amount of times as well. Hence, to enable productive use of our results we have to address both challenges. In this paper, however, we focus on potential improvements for the equalization process. For more results and an examplary presentation of distorted and equalized images, see [KCDV12].

2.1 Summary of the used concept

First of all, we introduced a definition of types of perspective distortion. These types describe the presence and dimension of perspective distortion starting with no distortion up to irregular distortions in two dimensions[10].
Using this graduation we then formalized the sensors view on a surface with traces by forming tuples of feature classes following the scheme used in [KLD06]. These classes are:

- the surface, the trace is applied on
- the trace pattern, that describes the trace itself
- the gradients of the surface
- and environmental influences

Based on this formalization the three main steps in figure 3 of an equalization were developed.

[9] using the "'Keyence VK-X110, a scan of a fingerprint of the test set of table 1 easily exceeds 20 hours
[10] two dimensional perspective distortions can not be projected into a distortion free representation, since none of the surfaces principal curvatures is zero

Figure 3: equalization pipeline

First the **topography data is filtered** by subtracting a Gaussian filtered version of the topography data to manage outliers and noise, because topography data is the base of our equalization approach. The resulting filter mask covers all useable areas for the following **blockwise determination of gradients**. In each block all pixel to pixel slopes are measured and combined into a block global gradient. The results of all blocks on a coherent homogenously distorted area form a global gradient, that is used for the **rescaling** process **of the original data**.
It should be mentioned that all the investigators decisions still base upon the original unprocessed data. The new equalized versions are only a helpful support to provide a well-founded different view on a trace.

To evaluate the improvement of an equalization we introduced a landmark based relative error to compare distorted and equalized representations of a scan. Therefore, landmarks were placed precisely within a test set. Subsequently, all distances between all landmarks were computed and compared to a planar scan to describe their deviation.

This proceeding requires topography and intensity or color data, which in our case were captured with the confocal laser microscope "'Keyence VK-X110"' [Key]. All methods are implemented in Java.

3 Statistics on Equalization and a Potential Productive Use

The following section gives an overview on the amount of data, that is used and created during an equalization. We also look into the calculation times and deduct requirements on database systems and database management systems.
As shown in our previous work, this first evaluation on equalization of non-planar surfaces gives a promising outlook on its potential of topology correction [KCDV12].
Future test sets need to be significantly greater in order to form a general conclusion and to approach to a potential productive use in the field of forensics. For a potential daily use the acceptance of theses methods and contactless fast acquisition of traces are preconditioned.

Such an application of equalization methods for a daily use with more than one crime scene or a crime scene with a high rate of traces might result in a great number of traces. Assuming an emergence of 1000 traces a huge amount of data and process time will be needed. Supporting a forensic investigator and his work, as the main goal of the equalization process, fast and structured access to all traces is needed. The actual needed amount of time for a contactless and perspectively distorted trace to be read and equalized is about minutes. However, based on our contacts with local, federal and international law enforcement agencies, seconds instead of minutes would be practicable instead. Below an

outlook is given to motivate techniques in database systems and database management, that support fast file transfers and parallel calculations.

All statistical information used in the following section were documented using VisualVM [VVM][11]. If not mentioned otherwise these statistics were collected on the equalization of 8 consecutive processed combinations of traces (see table 1).

In Table 2, the amounts of data for an equalization of non-planar surfaces with applied traces are presented for the acquired area, the raw data and the resulting maximum heap data for a consecutive processing of the given test set (see table 1).

	minimum	**maximum**
scan area	1342.80µm x 1824.89µm	9390.39µm x 14250.64µm
intensity data	5.4 MB	74 MB
topography data	5.4 MB	74 MB
used heap data	0 MB (starting point)	1.05 GB

Table 2: summary of the minima and maxima of the acquired area, the amounts of used raw and heap data

The size of used raw data[12] for given test set (see table 1) vary from 10,8MB[13] to 148MB[14]. The whole process splits into the import of raw data und the actual equalization calculations. After the import and during determination of distortion and rescaling, the amount of used data easily increases by the factor of 7. In our tests the resulting amount of used heap space during a single equalization rose up to 1.05GB[15]. So the overhead of data are not a few kilobytes, but rather hundreds of megabytes up to gigabytes.

The data we deal with in this case are the raw intensity and topography data in their original form, as well as filtered, subtracted and binarized topography data, resulting filter masks, data about block calculation/ positioning and the rescaled versions of intensity/ topography data.

Despite large acquisition times due to measurement speed of the used equipment, the processing of this data in a daily use seems also impractical as we analyze the CPU times in the following. Table 3 summarizes the maximum, minimum and mean CPU times for the equalization of the given test set(see table 1).

[11] on a system, using: Intel(R) Core(TM) i7-2670QM @ 3.1GHz, 8GB RAM; JavaVM @ 4GB, WDC WD6400BPVT-60HXZ

[12] data size is given in total: sum of size of intensity- and topography data

[13] scan of a partial human hair at 70 degrees (area: 1.35mm x 1,83mm)

[14] complete scan of a fingerprint at 20 degrees (area: 14,26mm x 9,40mm)

[15] value refers to the equalization of a complete scan of a fingerprint at 20 degrees (area: 14,26mm x 9,40mm)

amout of time	equalization	import	computation
maximum	10.30min	9.27min	2.50min
minimum	0.39min	0.28min	0.08min
mean	2.65min	2.07min	0,59min

Table 3: maximum, minimum and mean CPU times: entire equalizations, import of raw data and calculation steps; based on scans of given test set, see table 1

The filter masks for equalizations of microtraces were created in a previous manual step, so that additional data and CPU time might be addable. Since the resulting filter masks for the human hair only cover the small area of the hair itself, the calculation part of the equalization does not take a long time.

The information in the following figure 4 were measured using VisualVM[16] [VVM]. The average CPU time is visualized for hot spot classes.

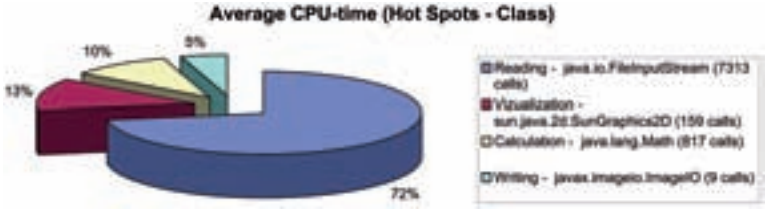


Figure 4: Average CPU-time: Hot Spots - Class

The biggest amount of time is needed to just read original and cached data. Of course the used system is to blame for that, however imagining not only a single processing of a trace surface at a time but several queries, the throughput will increase distinctly, since the majority of calls is also focussed on read operations.

Subsequently, we highlight requirements and starting points for potential improvements for our approach.

3.1 Frequent Data and Filter Masks

As shown above, reading data takes most of the time, so the access on all this data and cached data needs to provided with a high performance. Especially a fast mapping of the filter mask onto the filtered topography data would increase the access to all calculation blocks without a need for a try and error approach for every possible block.
Among other things, Table 3 contains the computation times within an equalization. The equalization of fingerprints takes the most time within our test set, because ridgelines and their potential of a good diffuse reflection creates a big amount of usable topography data.

[16]The average CPU-time was calculated for 8 consecutive equalizations (based on scans of given test set, see table 1)

On the other hand we are forced to use small block sizes (20 to 20 pixels or below) to be able to place them on the thin ridgelines, which significantly increases the number of blocks and thereby the calculation time. This dependency between the number of blocks and the present trace or surface endorses an internal mapping of filter masks on the topography data.

3.2 Potential Improvement of Parallelization

As pointed out in Table 3 already the import of the original intensity and topography data takes a decent amount of time. Hence to that amount, a parallel import of raw data (intensity and topography data) is obvious.

As mentioned above the necessary CPU time depends on the number of used blocks, which itself depends on the amount of usable areas in the topography data[17], the size and resolution of the scanned surface and on the used block size. The bigger blocks are on an evenly shaped part of the acquired surface, the more reliable the result of the determination of gradients will be. Since big consecutive areas in the topography data are rare the block size is limited, which results in a greater number of blocks.
Since the blockwise calculation of gradients is done sequentially, more time can be spared by parallelizing their computation. An additional benefit from this approach is that with all valid blocks the largest coverable area of topography can be determined. The number of blocks affects the potential of parallelization, though.

Another aspect of potential parallelization is the parallel equalization of multiple traces. Starting with just a few blocks for every surface to check on their slopes, traces with better quality and less perspective distortions can be processed first to provide an investigator with more reliable results first.

3.3 Forensic Data Treatment

For forensic purposes, it is necessary that no original data may be altered or deleted to preserve its integrity and to assure the confirmability of the whole process. Possible solutions might be the use of forensical file formats like AFF4 [CS10] or the invertible fragile watermarking approach from [SSM+11], which could be used on non-trace areas of the image to obtain the integrity without altering any trace information. This also implies that a forensical secure way of logging and access of data is mandatory. When it comes to logging - all parameters, used functions and the equalization result need to be stored.

[17] in case of quasi-planar applied traces, the slope of the substrate might be used as well; using a HDD platter in the present test set eliminates this option due its high reflection characteristics

4 Summary and outlook

Perspective distortions are one of the main problems of contactless trace acquisition. On the other hand the benefits of integrity-preservation and repeatability make such non-invasive acquisition methods valuable for forensic purposes. When aiming for a productive use of the presented pre-processing methods, adjustments to the underlying architecture are important to provide an investigator with fast and secure additional information within a decent amount of time.

What are the requirements on a database and a database management system to fulfill the requirements of the presented scenario?
In the previous section we gave a perspective on the amounts of data and used processing time, that are necessary for the equalization of non-planar surfaces in order to deal with perspective distortions. Especially matching of data when using filter masks or addressing calculation blocks is a important aspect. Furthermore the potential of parallelization for blockwise gradient determination and parallel equalization of multiple traces were motivated to be supported by the database and database management system. Not least the storage of all used data and logging of all processing steps and parameters is mandatory to cope with the requirements of a forensic investigation. That includes integrity and authenticity issues as well as further security aspects.

Acknowledgments

The work in this paper has been funded in part by the German Federal Ministry of Education and Science (BMBF) through the Research Program under Contract No. FKZ:13N10818, FKZ:13N10817 and FKZ:13N10816.

References

[CS10] Michael Cohen and Bradley Schatz. Hash based disk imaging using AFF4. *Digit. Investig.*, 7:S121–S128, August 2010.

[Gun07] Dr. S. Gundagin. Sex Determination from Fingerprint Ridge Density. *Internet Journal of Medical Update*, 2007.

[KCDV12] Stefan Kirst, Eric Clausing, Jana Dittmann, and Claus Vielhauer. A first approach to the detection and equalization of distorted latent fingerprints and microtraces on non-planar surfaces with confocal laser microscopy. pages 85460A–85460A–12, 2012.

[Key] Keyence. Keyence VK-X110. [online] http://www.keyence.de/products/microscope/laser/vkx100_200/ vkx100_200_specifications_1.php, last seen: 02.07.2012.

[KLD06] Stefan Kiltz, Andreas Lang, and Jana Dittmann. Klassifizierung der Eigenschaften von Trojanischen Pferden. D-A-CH Security 2006, Düsseldorf, Deutschland, Syssec, ISBN: 3-00-018166-0, Patrick Horster (Eds.), 2006.

[KM09] Peiponen K.-E. Kuivalainen, K. and K. Myller. Application of a diffractive element-based sensor for detection of latent fingerprints from a curved smooth surface. In *Measurement Science and Technology 20(7),077002*, 2009.

[SSM+11] Martin Schäler, Sandro Schulze, Ronny Merkel, Gunter Saake, and Jana Dittmann. Reliable provenance information for multimedia data using invertible fragile watermarks. In *Proceedings of the 28th British national conference on Advances in databases*, BNCOD'11, pages 3–17, Berlin, Heidelberg, 2011. Springer-Verlag.

[VVM] VisualVM Version 1.3.5. [online] http://visualvm.java.net last seen: 12.11.2012.

[WGT+] Craig I. Watson, Michael D. Garris, Elham Tabassi, Charles L. Wilson, R. Michael Mccabe, Stanley Janet, and Kenneth Ko. User's Guide to NIST Biometric Image Software (NBIS).

Secure Deletion: Towards Tailor-Made Privacy in Database Systems

Alexander Grebhahn[1], Martin Schäler[2], Veit Köppen[2]

[1] Brandenburg University of Applied Sciences
P.O. Box 2132, 14737 Brandenburg, Germany
grebhahn@fh-brandenburg.de

[2] Department of Technical and Business Information Systems
Otto-von-Guericke University Magdeburg
P.O. Box 4120, 39016 Magdeburg, Germany
{schaeler, vkoeppen}@iti.cs.uni-magdeburg.de

Abstract: In order to ensure a secure data life cycle, it is necessary to delete sensitive data in a forensic secure way. Current state of the art in common database systems is not to provide secure deletion at all. There exist academic demonstrators that address some aspects of secure deletion. However, they are limited to their deletion approach. We argue, due to different data sensitivity levels (probably even on attribute level) and differences in policies (e.g., time when and how a data item has to be deleted), it is necessary to have a standardized, user defined opportunity to enforce secure data deletion in a forensic secure manner. Our literature analysis reveals that most approaches are based on overwriting the data. Thus, in this paper, we examine how it is possible to integrate user-defined overwriting procedures to allow a customizable deletion process based on existing default interfaces to minimize the integration overhead. In general, we propose an extension of SQL and a page propagation strategy allowing the integration of a user defined deletion procedure.

1 Introduction

The amount of data stored in computer-aided systems increases continuously. This inherits challenges with respect to privacy of sensitive data. Laws and guidelines exist for handling private information, like the HIPAA [Con96] or the Directive 2006/24/EC of the European Parliament [Eur06]. Forensic secure data deletion is an important privacy challenge. For instance, due to mentioned EU directive, it is only allowed to store data for a fixed period of time. After this, data have to be depersonalized or removed. Consequently, to ensure a forensic secure life cycle data have to be deleted in a forensic secure way [DW10].

Additionally, it is necessary to use a system that supports access control, additional security mechanisms (e.g., public key infrastructures), and recovery components, to store sensitive data. Because database systems support these requirements, they are a good choice for storing sensitive data. That is why we focus on database systems, in this paper.

Using a database system, new challenges for secure deletion arise. These challenges are due to copies of original data, for instance, in backups, roll back segments used in transaction management, or due to implementation details of database operations. To quantify the impact of these challenges or to enable forensic investigations in database system, there have been several examinations of existing database systems (e.g., [Fow08]), revealing that there is currently no secure deletion at all. Thus, there are several approaches proposed in literature that are mainly based on overwriting the data in a predefined way that is limited to this approach and is enforced for all data in the system (e.g., [SML07]).

However, there are use cases, where it is not useful to delete all data stored in a database in a forensically secure way, apply alternative overwriting patterns, or to delete specific data after different time slots. Typical reasons are different sensitivities of data, performance issues, new research results, or changing guidelines for secure deletion. As a result, it is necessary to have a choice to enforce user-defined, fine-grained data-privacy constraints that state the way of forensic secure deletion.

In this paper, we examine: How to integrate such user-defined, fine-grained data-privacy constraints based on existing standard interfaces? Moreover, we want to provide a solution that optimally introduces no or a limited performance overhead.

The remainder of this paper is structured as follows: In Section 2, we present background information of deleting data from a database system. Beside this, we give an overview of different data artifacts that remains in the database system. In Section 3, we give an overview of related work addressing forensic secure deletion from a hard disk and present results and limitations of existing prototypes. In Section 4, we present some SQL extensions to have tailor-made data privacy within a database system. Additionally, we present requirements on the propagation strategy used in a database system to forensic deleting data. Within this part, we also present some possibilities to improve performance of this strategy. Finally, we conclude our paper and present future work.

2 Background

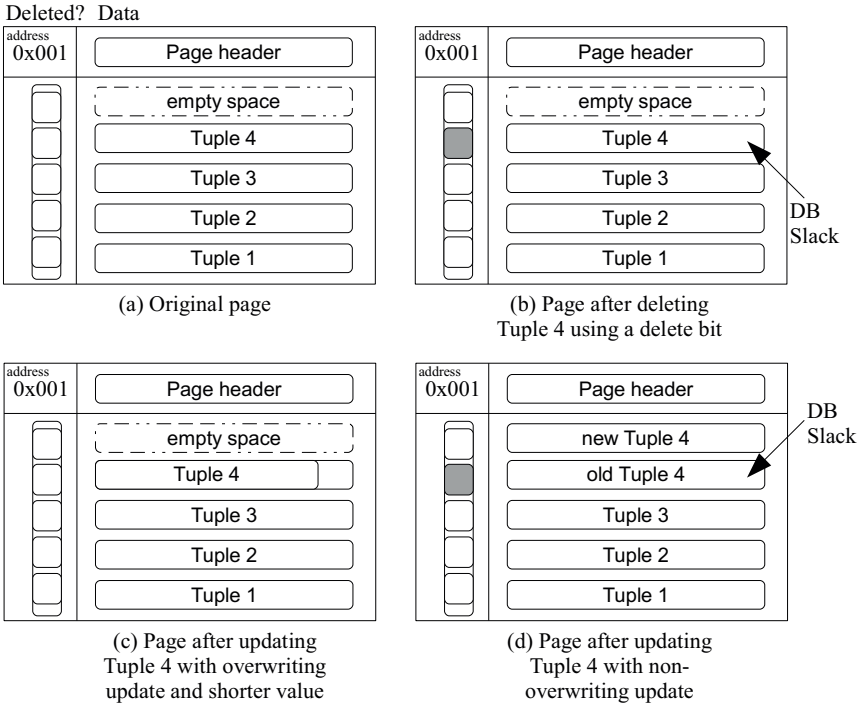
In the following, we list known challenges for forensic secure deletion from a database system to provide an overview of the complexity of this topic. Major challenges are hidden direct copies of data items or that some parts of the data are used (possibly in aggregated form) to create specific data structures out of the database system: For instance, such as indexes to improve performance of the overall system. Last, derived from the presented challenges, we present challenges, that are not in the scope of this paper.

2.1 Deletion of tuples from the database itself

With database, we mean the physical representation of data on persistent storage (e.g., hard disk) in the format of the database system. This is the obvious copy, we have to remove. However, in databases, supporting SQL, there are several operations that, in fact, require

deletion of a data item. Examples for these operations are: `DELETE, DROP TABLE, TRUNCATE TABLE, VACUUM`, and `UPDATE`. In the following, we unify our explanations for `DELETE, DROP TABLE, TRUNCATE TABLE`, since the last two operations enforce a deletion of all tuples equivalent to a `DELETE FROM <table>` command.

Pages in databases. Before we start to discuss details of the implementation of the single operations, we explain *pages*. In databases, a page is an abstraction of a physical segment of the persistent storage and the elementary unit that can be read or written. Consequently, whenever the system needs to access a single tuple the whole page is read. Hence, all our explanations are based on this elementary unit. In Figure 1.(a), we give an example for such a page. In our example, tuples are inserted from bottom to top.



(a) Original page

(b) Page after deleting Tuple 4 using a delete bit

(c) Page after updating Tuple 4 with overwriting update and shorter value

(d) Page after updating Tuple 4 with non-overwriting update

Figure 1: Creation of database slacks.

DB Slacks: Deleting a tuple. In databases, deleting a tuple usually means setting a delete bit such as in SQL Server [Fow08], Oracle 10g [Lit07a], MySQL 5.1.32 using the InnoDB storage engine [FHMW10, SML07], or PostgreSQL [Gre12]. In Figure 1.(b), we exemplary delete *Tuple 4* from Figure 1.(a) causing no change of the original tuple, but a modification of the delete bit. Hence, with simple analysis of databases and basic knowledge on database internals, we can reconstruct the deleted data item [SML07, Gre12,

FHMW10, Lit07a]. According to Stahlberg et al., we refer to these remains of a tuple as *database (DB) Slack* in the sequel.

DB Slacks: Updating a tuple. By updating a tuple, two different modifications of the page the item is located on can occur [SML07]. Firstly, the old version may be overwritten by its newer version (Figure 1.(c)). Secondly, the new version of the data item is stored at another physical storage segment (Figure 1.(d)). This new location is not necessarily at the same page. As result of the second alternative, the old version of the data item still exists after updating it. Even with the first behavior, information of the old version may remain on the disk, if the new version of the data item needs less space than the old one. In many major database systems, such as PostgreSQL[1], MySQL using the InnoDB storage engine, the second variant because they use the multi-version concurrency control protocol [SML07].

FS Slacks: The Vacuum command. The existence of a large amount of DB Slacks consumes unnecessary disk space and reduces overall performance of database systems, because there are deleted data items on pages which have to be read/written. To overcome this challenge there are operations like `VACUUM` (e.g., in PostgreSQL) that minimizes the space used by database systems. When performing this operation, firstly, active tuples are reorganized to minimize the overall number of needed pages. Secondly, disk space of no more needed pages is returned to the file system. We call these pages and respective tuples according to Stahlberg et al. [SML07] *files system (FS) Slacks*. To sum up, to delete a tuple in a forensic secure way, the database system either has to prevent or to be aware of all copies of a tuple that exist in DB Slacks or FS Slacks. If these copies exist, they have to be deleted too.

2.2 Hidden copies in database systems

Besides database, metadata of a database system and the log file have to be considered. In general, metadata information can be divided in two groups [Gre12]. The first group are data *independent* information not relevant for secure deletion. Examples for this group are table schemata or integrity constraints. The second group, however, data *dependent* information relevant for secure deletion. Examples for the second group are histograms and indexes.

Histograms. In contrast to the database itself, metadata do not necessarily contain a whole copy of a data item. For example, in a histogram only information of one column of data items is stored. In general, considering histograms, it is only possible to retrieve information in which range, the value of a deleted tuple was located. Nevertheless, in some database system (e.g. SQL Server [Fow08]), the values of existing tuples are used

[1] http://www.postgresql.org/docs/9.1/static/storage-vm.html

to set the border of two histogram buckets. Thus, by deleting the tuple, it is necessary to modify the border of the buckets.

Indexes. Moreover, index structures have to be considered, too. One of the most famous index structure is the B^+-Tree. In a B^+-Tree, values of tuples are used as decision criterion for searching in the tree. That means, deleting these values from the database, it is necessary to modify the structure of the tree, too. In worst case, this reconstruction can cause a complete new construction of the tree.

Materialization of intermediate results. Some database systems store additional data dependent information. Examples for this are intermediate results, stored for speed up query performance. A database system storing this information is MonetDB [IKNG09]. From this information, conclusions about values of deleted data items can be drawn. As a result, it is necessary to consider this information by forensic deleting data items.

Database log. Finally, database logs have to be considered. Here, it is necessary to differentiate whether a logical or a physical log protocol is used. By using a physical log protocol, for every modification a BEFORE image of the unmodified data item is stored. In contrast, in a logical protocol, functions are stored which describe modifications to be executed. At this point, one has to be aware, such functions cannot only modify one item [BHG87]. As a result, different solutions have to be used for the different protocols.

2.3 Challenges beyond the scope of this paper.

Besides the challenges we introduced so far, there are several effects that raise the difficulty for forensic secure deletion. However, these challenges are, for instance, due to operations system behavior and thus, it is hardly possible to address them by means of a database system. Although, these challenges are important, they are beyond the scope of this paper:

- Swap main-memory to persistent storage.
- Copying a page to a different location on the persistent storage device like on SSDs [WGSS11].
- Treatment of copies in backups, for instance, stored magnetic tapes.
- Reconstruction of the data through the use of microscopes like in [WKSR08]

3 Analysis of related work

In this paper, we do not want to provide a new approach how to delete data in a specific database, but abstraction from current solutions to provide a more general approach.

Furthermore, we want to introduce a general solution that can be integrated into existing database systems due to the modification of default interfaces. Therefore, we analyze existing approaches to identify similarities to provide such a generalized solution as well as to point out shortcomings of the state of the art to motivate the necessity of our new approach. As a consequence, we first present related work on forensic secure deletion from a hard disk in general.

Forensic secure deletion from hard disk Deleting data in a database system means removing the data traces from persistent memory (e.g., HDD). Thus, we have to consider related work on secure deletion from disk as well. There are approaches to physically destroy the storage medium and modifications that affect neighboring storage cells that shall not be removed. However, these approaches are infeasible for our purpose. Moreover, cryptographic approaches rely on keeping the encryption key secret and there may be a successful attack to break the encryption in the near future. Hence, we do not consider these approaches. Finally, there has been a lot of research how to overwrite data in a way it cannot be restored (e.g., [Gut96, WKSR08, DW10]). Since this approach allows secure deletion without destroying the storage medium, and is possible to implement with means of the database system as shown by Stahlberg [SML07] and demonstrated in our own work [Gre12], we consider it as the *appropriate technique for forensic secure deletion*. However, there are different results on how to overwrite data including the number of overwrites (PASSES) and bit sequences (PATTERN) that have to be applied [DW10]. For instance, Gutman suggests performing 35 PASSES using predefined and random PATTERN. In contrast to this, Wright et al. proposes, that one overwriting pass with random data is adequate for secure deletion [WKSR08].

Database specific forensic analysis Within the last years, a lot of work was done to analyze, which information from a database system can be used to allow forensic examinations.

Different database systems store different specific information with different granularities, most of the work thus focuses on specific database systems. For instance, Lichfield analyzes different parts of oracle database[2]. In some parts of his examination, he focus for example on dropped objects [Lit07a] or on undo segments [Lit07b]. Fowler analyzes the information stored by SQL Server [Fow08]. Frühwirt et al. describe the structure of MySQL Database 5.1.32. with InnoDB Storage Engine [FHMW10]. In their work, they give a detailed overview on the structure of the information and show how to use this information to reconstruct parts of a database table after data items are deleted. However, since the main goal of all these work is to enable forensic examinations of databases, which is in direct opposition to our goal of forensic secure deletion, they do not provide solutions how to perform such a deletion. Nevertheless, their basic research is a fundamental prerequisite to enable forensic secure deletion.

[2] http://www.databasesecurity.com/oracle-forensics.htm

Solutions for forensic secure deletion in database systems. Miklau et al. consider forensic deletion of items from a database in a more theoretical way [MLS07]. They examine, which challenges a database designer have to handle with, by creating a database system that has a securely management of its history.

To the best of our knowledge, Stahlberg et al. [SML07] have conducted the most comprehensive study and provide the most general solution for forensic secure deletion. They consider secure deletion of data items for MySQL using the InnoDB storage engine. Their solution for forensic deletion is *overwriting* values of the item at the moment the storage space of the data is set to be free for storing new items. Furthermore, they considered one index to highlight generality of their solution. When evaluating their method, they could not prove a performance penalty, because always the whole page is written to persistent storage independent whether a delete bit is changed or the data itself shall be overwritten. As a result, overwriting values of a deleted data item in MySQL using InnoDB can be done without performance penalties. However, even their solution is restricted to one (common) database system, and does not address different granularities for forensic secure deletion or temporal constraints when a tuple has to be deleted.

In prior work, we show that Stahlberg's solution also works for different database systems [Gre12]. Firstly, we modify PostgreSQL version 9.1. In our implementation, we overwrite the values of the data items at the time, when no transaction is performed that can access the data to delete. At this moment, the whole page of the item is written to disk. For that reason, we do not measure a performance penalty. So, the solution presented by Stahlberg et al. can applied to PostgreSQL. Secondly, we extend HyperSQL[3] with this method. Evaluating our extension, we recognize a time penalty, because of the page propagation strategy of HyperSQL. Results of a committed transaction are not materialized at commit time due to performance issues. The results are materialized, when the item was not used recently and the buffer, the results are stored in exceeds the assigned main memory. In worst case, this is never done, because of high amount of main memory. Thus, we modify the propagation strategy in a way that after each committed transaction, deleted and old versions of updated items are overwritten. Because of this modification, we introduce an additional I/O operation. We notice a nearly constant performance penalty, when measuring the performance of our modified HyperSQL version.

Additionally, to support a secure data management where no unnecessary information about history of the database is stored in metadata, history-independent index structures have to be used. Within this type of structures no information is stored about operation sequence that lead to current state of the index [NT01].

To sum up, implementing an overwriting strategy for forensic secure deletion is possible using a modified page propagation strategy. Moreover, based on the way the propagation is implemented, single overwrites do not affect the performance at all or introduce a constant overhead. However, since performance is a critical issue, optimizing page propagation especially with multiple overwrites is an important task.

[3] http://hsqldb.org/

3.1 Limitations and resulting generalization of related work

According to our literature research, current solutions for secure deletion are limited w.r.t. three criteria:

1. Database system specific: All solutions are currently limited to a small number of database systems since they demonstrate the feasibility of the proposed approach.
2. Fixed granularity: All current solutions do not allow a fine-grain definition of the deletion mechanisms to enforce at table or attribute level, nor they are able to define different deletion dates based on data sensitivity.
3. Fixed overwrite procedure: Current approaches specify a fixed overwriting procedure (containing number of PASSES and PATTERN), however due to new research results, evaluation purposes, and nation-specific guideline, such as [Gov06], we need a more flexible approach.

4 User defined privacy

In this section, we describe extensions a database system needs to support different policies defined by guidelines to delete data items forensic secure. Due to limited space, we only focus on two abstract interfaces provided by a database system. Firstly, we focus on the SQL interface. Through this interface, a user communicates with the database system. For offering a user a choice to tailor her privacy constraints, it is necessary to extend the SQL syntax supported by the system. This is because, to the best of our knowledge, tailor-made privacy in not covered by the SQL standard. Secondly, we focus on the layer propagating cached pages on a persistent storage.

In particular our solution offers:

1. Generality: Due to an extension of the SQL language and the page propagation strategy.
2. Granularity: Definition of the deletion procedure at table and attribute level, separately.
3. Tailoring: An SQL extension to specify user-defined overwriting procedures (including PASSES and respective PATTERN).

In Figure 2, we give a schematic overview of the structure. Within the persistent data storage, three tables are located. In table 2 and 3 sensitive data is stored that have to be deleted in a forensically way. Due to the fact, that different policies have to be supported through the deletion performed from table 2 and 3, different delete algorithms have to be supported. We discuss details of our solution in the following sections.

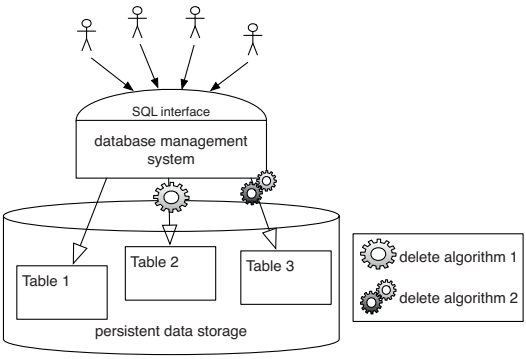


Figure 2: Schematically structure of our solution.

4.1 Tailor-made privacy through SQL commands

In order to have tailor-made privacy and to support different guidelines, it is necessary to offer a user an interface where she can define privacy constraints that have to be supported by the system. Thus, database systems have an interface for communicating with the user, namely the SQL interface, we extend this interface. It is necessary to define different overwriting pattern and to use these pattern in a specific sequence. After defining these pattern and passes, it is necessary to use them to delete a data item in a forensic secure way. As a result, it is not only necessary to insert new SQL commands, but also to extend existing one.

In general, there are two different moments when to propagate to the system, that data items have to be deleted forensic secure. Firstly, in creating the database and secondly, when performing a modification on the data item. Well, before creating the database, one has to be aware of the information stored and the guidelines that have to be supported. Thus, it is possible to extend the syntax of the `CREATE TABLE` statement. Additionally, when performing operations that have an impact on the physical representation of data items from different domains, one has to be aware of all regulations, if they are not propagated at creation time of the tables. In other words, if for example a `VACUUM` operation has to be performed on all tables, a user has to extend the SQL command with all needed constraints defined through policies. In databases having many tables, this can become very confusing. Moreover, if privacy constraints are known before inserting data, the amount of persistent stored metadata can be minimized. Thus, it is possible, not to store indexes or histograms persistent. As a result, we focus on SQL solutions propagating privacy constraints as soon as possible.

To summarize, for supporting a fine-grain, tailor-made forensic deletion of data items, the following definitions have to be supported via SQL interface:

1. Definition of pattern and overwriting passes,
2. Definition of secure tables, using predefined passes, and

3. Selective deletion of data items and columns of data after a fixed time

Definition of pattern and overwriting passes. In Listing 1, we present an extension of the SQL syntax for defining specific pattern to overwrite data. In detail, we extend SQL with two additional keywords, first `PATTERN` and second `WITH`. With the `PATTERN` keyword, it is possible to define a new overwrite pattern with a name (e.g., `p1` in Line 1). To define the design of this pattern, we use the keyword `WITH`. In this statement, it is possible to define sequences of `0`s and `1`s (see, for example, Line 1 and Line 2 in Listing 1). Additionally, it has to be possible, to reuse existing predefined pattern (Line 3). Due to the fact that pattern have to be used for data items of different length, pattern have to be repeated until the whole item is overwritten.

Listing 1: Definition of pattern.

```
1 CREATE PATTERN p1 WITH 0;
2 CREATE PATTERN p2 WITH 100;
3 CREATE PATTERN p3 WITH p1,p2;
```

After defining pattern, it is necessary to use these patterns in a defined sequence. For defining these sequences, we propose the syntax of the SQL statements given in Listing 2. We propose the `PASS` keyword. By the use of this keyword, new sequences of pattern can be defined for forensically deleting data items. Again, we use the `WITH` keyword, as proposed earlier, to define the design. In the `WITH` clause, it should be possible, (a) to use predefined pattern, (b) to define new pattern, (c) to define a pass containing of random data (`RANDOM()` see Line 1 in Listing 2), and (d) to reuse already defined passes (see for example Line 2).

In Line 1, we give an example for the syntax of a pass supporting the ”Declassifying Electronic Data Storage Devices” of Government of Canada [Gov06], to delete protected information of class B (”Compromise could result in grave injury, such as loss of reputation or competitive advantage”)[4]. Within three passes, first, data is overwritten with 0s, second with 1s, and last with a pseudo random pass.

Listing 2: Definitions of overwriting passes.

```
1 CREATE PASS over1 WITH p1, 1, RANDOM();
2 CREATE PASS over2 WITH p2, over1, p3;
```

Definition of a forensic table. Next, it is necessary to use overwriting passes and related pattern for forensic secure deletion data. Thus, we prefer solutions defining sensitive information at creation time of a table, thus we extend the `CREATE TABLE` statement. In Listing 3, we present our syntax for using predefined overwriting passes. As you can see, we extend the SQL syntax with two additional words. Firstly, with the `FORENSIC` keyword, we state that sensitive information is stored in the table. Secondly, by the use of keyword `USE`, we define which overwriting pass has to be used, when forensically deleting data items. To support a fine-grain column wise specification, we give the possibility

[4] http://www.tbs-sct.gc.ca/pol/doc-eng.aspx?id=16557&section=text

to use different overwriting passes for different columns of one table. We give an example in Line 2 in Listing 3.

Listing 3: Usage of overwriting passes in a CREATE TABLE statement.

```
1 CREATE FORENSIC TABLE t1(c1 varchar(40), c2 int) USE
    over1;
2 CREATE FORENSIC TABLE t2(c1 varchar(40) USE over1, c2
    int USE over2);
```

If two different overwriting passes are used within one table, some implementation challenges have to be focused. If the database system uses a column oriented storage, finding the specific storage segment of value of a given column is unproblematic. In contrast to this, if data items are stored tuple wise, identifying the storage region of a data item is much more complex. As a result, is may be necessary, to extend the meta information stored about storage length of data items.

Definition of a forensic table with a fixed time. According to some policies defined through guidelines (e.g. Telekomminikationsgesetz [Bun04]), it is necessary to delete information after a fixed time. For automatic forensic data deletion after given time, it is necessary to know when and how to delete. In Listing 4, we provide an example. As stated earlier, we use again the `FORENSIC` and the `USE` keyword and extend the syntax with the `FOR` keyword. In the part of this keyword, it is possible to define a time slot a data item retrains in the database. For internal management of this constraint, it is possible to use existing techniques, like jobs from an oracle database system. Moreover, it may be necessary to forensic delete only a column of data items after a fixed time, while the rest of the item still remains in the database. Well, this is only allowed, if null is allowed as value of the column. However, this partial deletion brings additional challenges in propagation of the data item, we discuss later.

Listing 4: Storage for a specific time.

```
1 CREATE FORENSIC TABLE t3(c1 varchar(40), c2 int) USE
    over1 FOR 10*60*24;
2 CREATE FORENSIC TABLE t4(c1 varchar(40) USE over1 FOR
    10*60*24, c2 int);
```

4.2 Propagation of data items

After offering the user the possibility to define several overwriting passes for forensic data deletion, it is necessary to perform multiple propagation of a page to overwrite the storage segments multiple times. As a result, it is necessary to modify the propagation strategy used of the database system. In contrast to existing propagation strategies, it is necessary to differentiate between inserting a new data item on a page and deleting an existing one. This is simply because, when inserting a new item, it is only necessary to propagate the

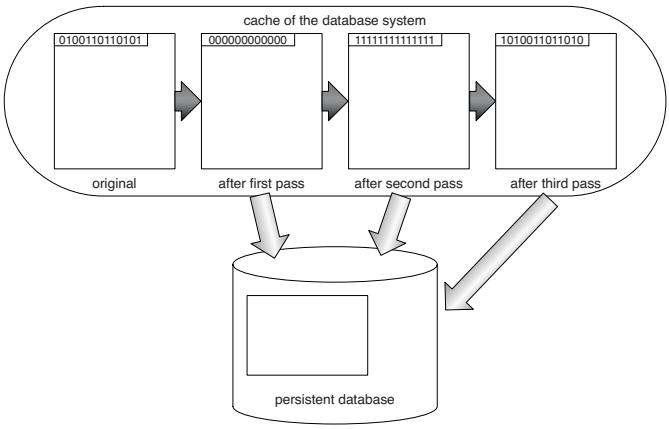


Figure 3: Schematic overview of propagation performed for a page to delete an item.

item once. In contrast to this, it is necessary to overwrite the physical storage segment of a data item multiple times if the item has to be deleted. In Figure 3 we give a schematic overview of the multiple propagation of one page to delete a data item. In detail, we show exemplarily the propagation needed to support the "Declassifying Electronic Data Storage Devices" of Government of Canada [Gov06]. First, after the data item was deleted from the original page, the space used by the data item is overwritten with 0s. Next, it is necessary to propagate the page with the overwritten data. After propagating the page, the storage segment of the page where the item is located have to be overwritten with 1s, then again propagated. Last, the segment is overwritten with random data and propagated again. Beside `INSERT` and `DELETE`, there are other operations having an impact on the physical representation of a data item. On the pages level, each operation can be split into insert and delete operations. For example, an update of a data item is nothing more than an insert of the new version and a delete of the old one. As a result, the old version has to be overwritten multiple times and the new one has to be inserted once. Thus, in this section, we focus only on delete and insert of data items on a page.

Propagation of data modifications. As stated earlier, for supporting a policy, it is necessary to overwrite storage segments multiple times with different predefined pattern in a defined sequence. Thus, each write operation performs an additional I/O-operation, performance of the database system decreases dramatically, if synchronous page propagation is performed. To overcome this, we prefer an asynchronous propagation of modified data items. Through this, it is possible to modify data stored at pages while performing deletion of another item stored at the same page. Yet, this asynchronous propagation brings some drawbacks for user experience. This is because, there is a time delay between deleting a data item through a SQL command with receiving feedback from the system and the real physical overwriting of the data, in write intensive systems. Consequently, it is necessary to define a maximum time delay that is allowed by the system.

However, due to the fact that multiple operations with different pattern have to be per-

formed to forensically delete one item, it is necessary to store additional information on the page. For example, it is necessary to store number of already performed overwriting operation for a data item, because after a system error occurred, it is necessary to continue with not finished overwriting passes.

Partial deletion from data items. After performing a forensic deletion of a whole data item, it is uninteresting, which values are stored at the storage location of the data item. In contrast to this, when performing a partial secure deletion, some values might bring problem. This is simply because, the rest of the data item has to be readable after deleting the values of the column and overwriting parts of the data random values might produce unpredictable side effects. To overcome this, different solutions can be performed. Firstly, after all write operations are performed to delete the data secure, an additional pass containing of binary nulls can be performed. Secondly, the new version of the data item can be stored at a new storage segment without the values of the column of interest and the old version of the item can overwritten with the defined pattern.

One disadvantage of the first version is the additional I/O-operation performed in marking the no more used storage segment as null. An additional disadvantage is the unused storage segment that might occur within data items. For example, after the secure deletion of a value of Column c1 from a data item stored in Table t4 (see Listing 4), the data item only needs space to mark that c1 is null and to store value of c2. In contrast to the required space, the data item uses the space it needs before deleting c1. In contrast to this, the second solution overcomes this problem. A drawback of the second solution is the modification of the location the data was stored at. As a result, it is necessary to update all references pointing to the old location of the data item. To sum it up, there are different solutions how to perform a secure deletion of parts of a data item.

Optimizations of the propagation. To minimize amount of write operations, it is possible to merge propagation of different data items stored at the same page. Especially in write intensive systems, this brings a huge performance benefit. In general, merging two different operations is only possible, if different storage segments are used for the data items. For example, it is only possible to store a new data item at a specific storage segment after all overwriting passes of an old data item former located at the same storage segment where performed. However, when merging propagation operations of two or more data items, one have to differentiate between (1) insert and delete operation performed on the same page and (2) multiple delete operations. For simplicity, we only describe how to merge two operations, but the proposed method can be extended to more than two data items without loss of generality.

1. If a new data item is added to the page, while performing the delete passes of another one, it is possible, to integrate the propagation of the new item into the I/O-operation the next deletion pass of the old one performs.

2. If more than one data item has to be deleted from the page, it is possible to merge passes of the delete operations. In detail, let i be the number of deletion passes

already performed for the first data item, j be the number of passes that have to be performed for every data item and P_i the pattern for the ith delete operation. Then perform delete operation $i + k$ $(1 \leq k \leq j)$ with pattern P_{i+k} for the first data item and operation P_k for the second data item with the same I/O-operation.

Until now, merging of propagation is only performed if two operations having an impact on the same page have to be performed at the same time. To minimize number of I/O-operations in systems having less write operations, it might be possible to wait with the write operations of one data item for a specific time until another data item has to be deleted from the same page.

5 Conclusion & Future work

In this paper, we explained what has to be considered, when forensically deleting data from a database system. In particular, it is necessary to delete information of all data inventories created by the system. To support deletion strategies proposed in different guidelines in one database system, it is necessary to have the possibility to perform multiple propagation of a modified data item. As a result, we propose a propagation strategy. For giving a user the possibility to tailor her privacy constraints, we extend the SQL syntax with (a) the possibility to define overwriting pattern, and passes, (b) define forensic tables and (c) to extend definition of forensic tables for storing data only for a fixed time.

In future work, we want to extend HSQLDB with the presented propagation strategy, extend the SQL syntax, and evaluate the performance. Due to the fact that different database operations have different number of I/O-operations performed by the propagation strategy, we want to perform benchmarks and evaluate performance.

6 Acknowledgments

The work in this paper has been funded in part by the German Federal Ministry of Education and Science (BMBF) through the Research Programme under Contract No. FKZ:13N10816 and FKZ:13N10817.

References

[BHG87] Philip A. Bernstein, Vassos Hadzilacos, and Nathan Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

[Bun04] Bundesministeriums der Justiz. Telekommunikationsgesetz, 2004.

[Con96] United States Congress. Health Insurance Portability and Accountability Act (HIPAA), 1996.

[DW10] Sarah M. Diesburg and An-I Andy Wang. A survey of confidential data storage and deletion methods. *ACM Compututing Surveys*, 43(1):2:1–2:37, December 2010.

[Eur06] European Parliament and the Council of the European Union. Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC. *Official Journal of the European Union*, L 105:0054–0063, 2006.

[FHMW10] Peter Frühwirt, Marcus Huber, Martin Mulazzani, and Edgar R. Weippl. InnoDB Database Forensics. In *AINA*, pages 1028–1036, Washington, DC, USA, 2010. IEEE Computer Society.

[Fow08] Kevvie Fowler. *SQL Server Forensic Analysis*. Addison-Wesley Professional, 2008.

[Gov06] Government of Canada. Clearing And Declassifying Electronic Data Storage Devices (ITSG-06). UNCLASSIFIED publication, issued under the authority of the Chief, Communications Security Establishment (CSE), July 2006.

[Gre12] Alexander Grebhahn. Forensisch sicheres Löschen in relationalen Datenbankmanagementsystemen. Master thesis, University of Magdeburg, Germany, 2012. In German.

[Gut96] Peter Gutmann. Secure Deletion of Data from Magnetic and Solid-State Memory. In *USENIX Security*, pages 77–89, 1996.

[IKNG09] Milena G. Ivanova, Martin L. Kersten, Niels J. Nes, and Romulo A.P. Gonçalves. An Architecture for Recycling Intermediates in a Column-store. In *SIGMOD*, pages 309–320, New York, NY, USA, 2009. ACM.

[Lit07a] David Litchfield. Oracle Forensics Part 2: Locating Dropped Objects. *NGSSoftware Insight Security Research (NISR) Publication, Next Generation Security Software*, 2007.

[Lit07b] David Litchfield. Oracle Forensics Part 6: Examining Undo Segments, Flashback and the Oracle Recycle Bin. *NGSSoftware Insight Security Research (NISR) Publication, Next Generation Security Software*, 2007.

[MLS07] Gerome Miklau, Brian Neil Levine, and Patrick Stahlberg. Securing history: Privacy and accountability in database systems. In *CIDR*, pages 387–396, 2007.

[NT01] Moni Naor and Vanessa Teague. Anti-persistence: History Independent Data Structures. *IACR Cryptology ePrint Archive*, 2001:36, 2001.

[SML07] Patrick Stahlberg, Gerome Miklau, and Brian Neil Levine. Threats to Privacy in the Forensic Analysis of Database Systems. In *SIGMOD*, pages 91–102, New York, NY, USA, 2007. ACM.

[WGSS11] Michael Yung Chung Wei, Laura M. Grupp, Frederick E. Spada, and Steven Swanson. Reliably Erasing Data from Flash-Based Solid State Drives. In *FAST*, pages 105–117. USENIX, 2011.

[WKSR08] Craig Wright, Dave Kleiman, and Shyaam Sundhar R.S. Overwriting Hard Drive Data: The Great Wiping Controversy. In *ICISS*, pages 243–257, Berlin, Heidelberg, 2008. Springer-Verlag.

Forensics on GPU Coprocessing in Databases – Research Challenges, First Experiments, and Countermeasures

Sebastian Breß, Stefan Kiltz, Martin Schäler

Faculty of Computer Science
Otto-von-Guericke University Magdeburg, Germany
{bress,kiltz,schaeler}@iti.cs.uni-magdeburg.de

Abstract: Recently, using GPUs for coprocessing in database systems has been shown to be beneficial. However, information systems processing confidential data cannot benefit from GPU acceleration yet because knowledge of security issues and forensic-examinations on GPUs are still fragmentary. In this paper, we point out key challenges and research questions related to forensics and anti-forensics on GPUs. Our results and discussion are based on analogies from similar computation environments, and experiences. Initial experimental studies indicate that data in GPU RAM is retrievable by other processes. This can be done by creating a memory dump of device memory. Hence, application data is accessible by users without access permissions, by bypassing the access control system of the database management system. Finally, we discuss approaches, how our results can be used in forensic and anti-forensic scenarios.

1 Motivation

Nowadays, information systems process an increasing amount of personal data, which has to be protected for unauthorized access. The core parts of information systems typically consist of one or more database systems, because they (1) store the application data and (2) handle efficient query processing on the data. Since database systems need to fulfill application's performance requirements, a lot of work focuses on increasing performance of database systems. Recent research focus on accelerating query processing of database systems using *Graphics Processing Units* (GPUs) as coprocessor [BS10, HLY+09, HYF+08, WWN+10]. GPUs are used to accelerate query processing [AW10, BS10, DWL+12, GGKM06, HLY+09, HYF+08, HY11, KLMV12, Pir12, PMK11, PSMK12, LDKA10] as well as query optimization [HM12].

Though GPU acceleration provides significant speedups [HLY+09], some systems only benefit from GPU coprocessing if and only if the confidentiality of data is ensured, e.g., in a scenario of bank accounts or fingerprint databases. GPUs have highly parallel computing capabilities with their own private memory and specialized programming interfaces (e.g., the proprietary CUDA [NVI12a]). A user *Mal* may access data of user *Alice* without permission, because multiple programs of different users can utilize the GPU at the same time. To verify this hypotheses, we investigate whether security protocols of a database system could be bypassed if no additional security measures are taken. In this paper, we take a

first step to allow forensic examination on GPUs as well as preventing users/programs to access data, which they must not access.

Beebe identified the need for forensic methods for "Non-Standard Computing Environments" such as mobile devices, cloud computing, and virtualization [Bee09]. A GPU is a non-standard environment for computations. However, GPU acceleration is not only database specific. Most of the research on General Purpose Computing using Graphics Processing Units (GPGPU) is done for different scenarios. For instance, see the survey of Owens et al. on GPGPU [OLG+07].

We expect that data with access restrictions will be processed on GPUs in the near future. This data may be the purpose of forensic investigation, or shall be removed from the GPU for privacy reasons in a forensically secure way. Additionally, direct access to GPUs may inflict additional security problems, e.g., because of programming or design errors. Approaches already exist, which utilize the GPU for accelerating forensic analysis, e.g., [JB06, MRR07]. However, to the best of our knowledge, there is no noticeable work investigating forensic analysis on GPUs. One of the core challenges is the closed nature of APIs that allow the direct access to the GPU. Hence, we cannot analyze the code statically to prove our assumption that data of previous processes is retrievable. Nevertheless, we can analyze the APIs themselves and derive conclusions. The contributions of the paper are as follows:

1. Raise attention and initiate discussion on forensic examinations of GPUs.
2. State a list of relevant research questions, underlying hypotheses, and respective experimental set-up and methodologies to prove or reject these hypotheses.
3. Perform first experiments and report lessons learned that are valuable for future research.
4. Furthermore, we suggest an approach to counter our discovered security threat.

The paper is structured as follows. In Section 2, we discuss preliminary work, such as background on GPUs, requirements for forensic investigations and our forensic model. We introduce challenges for forensic and anti-forensic when an application utilizes GPU coprocessing in Section 3. We conduct experimental evaluation of the most important challenges of forensic and anti-forensic in Section 4. Section 5 discusses the consequences of our experimental results for forensic and anti-forensic. Since we are not aware of any work on forensics of GPUs, we provide a more general overview of approaches that utilize the GPU for forensic examinations in Section 6. The paper closes with future work in Section 7 and a conclusion in Section 8.

2 Preliminary Considerations

In this Section, we discuss preliminary work. First, we provide background over GPUs. Second, we discuss legal and IT-forensic requirements. Finally, we provide a brief summary of our model of the forensic process in main memory investigation.

2.1 Graphics Processing Units

GPUs are specialized processors, which were initially designed to support graphical applications. Due to their parallel nature, they can be also used for general purpose computation using frameworks such as CUDA [NVI12a], OpenCL[1], or AMD APP SDK[2]. The most important properties of GPUs are: (1) They have more cores than CPUs and a significantly higher computing power than CPUs with the same price. (2) They target high throughput by performing parallel execution of tasks using multi-threading with thousands to ten-thousands of threads. (3) They have a dedicated memory, the GPU RAM (4). GPUs are optimized for computation with high arithmetic density. On the downside, a lot of control flow brakes the performance of a GPU kernel [SK10]. The Compute Unified Device Architecture (CUDA) is a framework for GPGPU. It provides a C interface and enables developers to write their parallel application code running on the GPU in CUDA C, an extension of the C programming language [NVI12a]. The GPU stores and processes data, which may contain relevant information for a forensic examination (e.g., screen content, application data). Data processing, especially if personal data is concerned, and forensic methods in particular need to meet a number of requirements. An exemplary chosen selection is outlined in the following.

2.2 Legal and IT-Forensic Requirements

Systems that operate on person-related data have to adhere to data protection legislation in most countries, such as the Data Protection Act 1998[3]. In order to assure data protection, suitable measures that ensure the security aspect of confidentiality [Bis02] and a working rights management have to be in place. This, in turn, means that any new development in DBMS, such as the proposed GPU coprocessing, has to provide means to ensure that data protection requirements are met.
For electronic evidence to be accepted in a court of law, they have to pass a test of admission. In the United States, the judge decides about the admissibility of evidence based on the Daubert challenge. The scientific validity of a technique or method is judged using originally five Daubert factors [DG01]:

- Peer review and publication,
- General acceptance in the relevant expert community,
- Potential for testing or actual testing,
- Known or potential rate of error,
- Existence and maintenance of standards controlling the use of the technique or method.

[1] http://www.khronos.org/opencl/
[2] http://developer.amd.com/tools/hc/appmathlibs/Pages/default.aspx
[3] http://www.legislation.gov.uk/ukpga/1998/29

Furthermore, for electronic evidence to be accepted, a comprehensive documentation (e.g., by using container formats such as [CGS09]) is mandatory. The concept of the Cyber Forensic Assurance (CFA) [Dar10] enforces this further by assembling *information security*, *information assurance*, the *Parkerian Hexad* and *information quality* into four main considerations and eight components [Dar10]. Component I contains Confidentiality and Possession, Component II of Integrity and Authenticity, Component III of Availability and Utility, and Component IV of Completeness and Accuracy.

The new approaches outlined in this article should integrate those considerations from inception to application in the field.

2.3 Models of the Forensic Process and Main Memory Investigation

To ensure a systematic course of action during a forensic investigation and that a maximum of case-relevant data is addressed, we have to use a model for forensic processes. Several models for digital forensics exist, see [Pol07] for a cursory overview. Kiltz et al. proposed a model that is already applied in main memory forensics and consists of three major components [KHDV09]:

1. *Phases* - grouping of investigation steps into 6 categories:
   - *Strategic preparation SP*
   - *Operational preparation OP*
   - *Data gathering DG*
   - *Data investigation DI*
   - *Data analysis DA*
   - *Documentation DO*
2. *Classes of methods* - grouping of forensic methods into 6 categories:
   - *Operating system OS*
   - *File system FS*
   - *Explicit means of intrusion detection EMID*
   - *IT-application ITA*
   - *Scaling of methods for evidence gathering SMG*
   - *Data processing and evaluation DPE*
3. *Data types* - differentiation of forensically relevant content into 8 layers:
   - *Hardware data* DT_1
   - *Raw data* DT_2
   - *Details about data* DT_3

- *Configuration data* DT_4
- *Communication protocol data* DT_5
- *Process data* DT_6
- *Session data* DT_7
- *User data* DT_8

The idea is to capture the forensic process in a comprehensive way. The central concept is that the system to be investigated is seen as a data source containing forensically relevant information in its data. *Methods* from the classes of methods (e.g., data processing and evaluation, DPE) process this data. The data is layered into the *data types*, targeting a particular data type and transforming it (typically in a more abstract layer, e.g., from raw data DT_1 to process data DT_6), depending on the particular *phase* (e.g., data investigation DI). This model also supports finding new means of locating, acquiring, investigating and analyzing forensically relevant data. This is done by providing a structured way of looking at data sources and methods to process them in the context of the forensic workflow. One such new opportunity to gain access to data sources and process them in a forensically sound way is outlined in this article in the case of the GPU-subsystem, which is suggested to act as a coprocessor to DBMS. In this case, access to the video memory stream managed by the GPU is needed for forensic investigations and access protection against IT-security related incidents. When looking for existing forensic tools or create entirely new ones, this categorization helps to identify the sources and where the methods provided reside. For instance, when trying to gather evidence from systems with no strategic preparation (i.e., no additional forensic methods are available, e.g., explicit means of intrusion detection EMID) the forensic expert is left with what the operating system OS, the file system FS and maybe the IT-application ITA may have gathered. Depending on the phase of the examination, not evaluated data is irretrievably lost (e.g., in the case of data gathering DG) or not taken into account (e.g., in data investigation DI or data analysis DA, violating the demand for completeness, see [Dar10]).

3 Challenges for Database Coprocessing and Security

In this Section, we discuss challenges for secure database coprocessing as well as for GPU Forensics.

3.1 Challenges for Secure Coprocessing

If a database systems utilizes coprocessing techniques, then it might be exposed to potential IT-security risks by circumventing access controls of a DBMS as demanded by Codd's rules [Cod82], or violating the IT security aspects (Confidentiality, Integrity, Availability, Authenticity, Non-Repudiation). To ensure secure GPU coprocessing, we identify the following challenges:

Enforcing Security Protocols: Can the access control component of a DBMS be bypassed to access data without the necessary access rights?

Ensuring Data Integrity: Can confidential data be secretly modified without using the interfaces of a database system?

Secure Processing: How can a GPU be utilized as crypto-processor using known techniques?

To enable database systems, which process confidential data, to benefit from GPU acceleration, the identified challenges have to be addressed.

3.2 Challenges for GPU Forensics

New forensic methods are required to acquire, examine, and analyze data processed on the GPU. Therefore, a detailed research of the GPU memory and proprietary APIs is necessary. To illuminate an incident, it is crucial to have knowledge about recoverability of data, e.g., under which conditions can application data still be accessed and examined. If we want to reconstruct the program flow, we have to be able to interpret the memory of the GPU. Therefore, we have to identify the GPU kernels and program stack as well as internal structures of a GPGPU framework API. Using this considerations, we may be able to reconstruct the program flow. Finally, it is desirable to have a similar tool support for GPU RAM forensic as for CPU RAM forensic, e.g., *Memoryze*[4] or the *Volatility Framework*[5]. Furthermore, a GPU could be used as crypto-processor, which leaves no discernible traces in main memory, resulting in a challenge for IT-forensics. In summary, we identify the following challenges to allow forensic investigations on GPUs:

Recoverability of Data: Under which circumstances is application data still dormant in GPU RAM, e.g., after deallocating memory or rebooting the machine?

Memory Interpretation: How can an acquired memory dump be efficiently interpreted?

- **GPU Kernel Identification:** Can we locate the GPU kernel in the GPU RAM?
- **Accessibility of Internal Structures:** Are internal CUDA/OpenCL/. . . structures accessible?

Reconstructibility of Program Flow: Can we reconstruct the program flow of an executed GPU kernel?

[4] `http://www.mandiant.com/resources/download/memoryze`
[5] `https://www.volatilesystems.com/default/volatility`

4 Experimental Studies

In this Section, we describe the experiments we conducted and present our results for two common CPU main memory forensic techniques applied on GPU device memory.

4.1 First Steps Towards the Acquisition and Examination of GPU Memory

In our experiments, we have to ensure a minimally invasive procedure. That means, we have to avoid modification of the object of investigation for reasons of data integrity, in this case the GPUs device memory. To investigate the GPU device memory, we have to use a GPGPU framework API to access the data in GPU RAM. We selected CUDA, because it is currently the most widely used GPGPU framework for database coprocessing, e.g., [BS10, HLY+09, HYF+08, HY11, LDKA10, MHS+11].

Since the CUDA API is closed source, we cannot verify, which changes the calls to the CUDA API cause in the GPU RAM. We can only derive assumptions from the documentation stating that the used API functions do not modify the GPUs device memory. In further research, however, this assumption should be verified. For all experiments, we used the following CUDA functions: **cudaMalloc**, **cudaMemcpy**, **cudaFree**, and **cudaMemGetInfo**.

cudaMalloc allocates a user specified number of bytes in the global GPU RAM [NVI12b]. **cudaMemcpy** copies data from CPU RAM to GPU RAM or vice versa. Note that when data is copied from non page-locked host memory, the data is first copied in page-locked memory in the CPU RAM and is then transferred to GPU RAM[6] [SK10]. The additional copy operation in CPU RAM may alter data relevant for a forensic investigation. **cudaFree** deallocates memory that was allocated by using **cudaMalloc**. **cudaMemGetInfo** provides information about the size of the currently available and total memory of the GPU RAM. The official CUDA documentation provides further details on the used CUDA API functions [NVI12b].

Note that since we are not concerned about forensic investigations in main memory, we omit the respective considerations in this paper. However, in a real world scenario, one has to consider main memory and GPU RAM forensics while also ensuring *integrity* and *authenticity* for the object of investigation in a heterogeneous system together with a comprehensive process accompanying documentation.

To ensure generality of our observations, we conducted the experiments on two machines with NVIDIA GPUs of different generations: (1) on a machine with an NVIDIA GeForce 9600 GT (compute capability 1.0) and Ubuntu 10.04 (32 bit) operating system and (2) on a machine with a NVIDIA GeForce GT 640 (compute capability 2.1) and Ubuntu 12.04 (32 bit) operating system.

[6] Usually, application data is stored in non page-locked host memory.

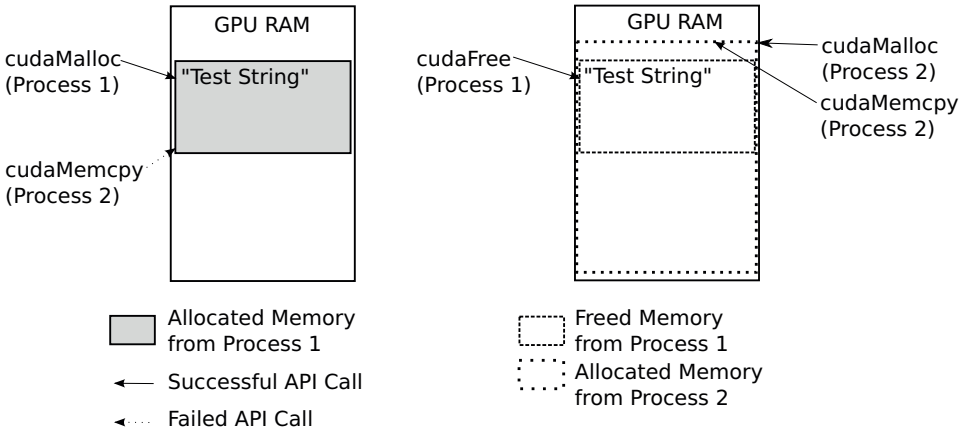


Figure 1: Experimental Results

4.2 Experiments

We now discuss our experiments in detail. We address the main challenges for forensics and anti-forensics. Therefore, we evaluate the challenges **Enforcing Security Protocols** (anti-forensics) from Section 3.1 and **Recoverability of Data** (forensics) from Section 3.2. Note that both challenges complement each other. That means, if security protocols are enforced, recovery of data from GPU RAM is not possible and vice versa. Therefore, our experiments targets the recoverability of data from GPU RAM. In our first experiment, we test whether it is possible to access memory that is not currently allocated by a program. In the second experiment, we examine whether data from terminated programs is accessible by a program. Finally, we apply our forensic model from Section 2.3 to the GPU RAM, because we have to verify to which degree our methods fulfill the legal requirements for forensic workflows.

4.2.1 Accessing non Allocated Memory in GPU RAM

In this experiment we investigate whether it is possible to access GPU RAM, which was previously allocated by **cudaMalloc** and afterwards freed by **cudaFree**.

Our program first allocates memory in the GPU RAM by using **cudaMalloc**. Afterwards, it copies a fixed user defined string in the GPU RAM by using **cudaMemcpy**. To be able to assert that the string was correctly written in GPU RAM, the program copies the data back to the CPU RAM by using **cudaMemcpy** afterwards. Since **cudaFree** sets a device pointer to zero, we copy the device pointer, which contains the address of the user defined string in GPU RAM, and name it *sneaky pointer*. Afterwards, we call delete on the device pointer acquired by the previous call of **cudaMalloc**. We then use *sneaky pointer* as an argument of **cudaMemcpy** to access memory that no longer belongs to the program. As a result, the **cudaMemcpy** function returned with an error code, which states an invalid

parameter values was passed. We repeated the experiment and made the same observations on both test machines. Figure 1 visualizes our results.

Apparently, CUDA has a memory protection mechanism, which does not permit the user to access currently allocated memory from another program. Furthermore, we conclude that it is not possible to access data in non allocated memory.

4.2.2 Creating GPU RAM Dump of Currently Free Memory

In this experiment, we investigate whether it is possible to access data in GPU RAM from previous executed programs, which freed all used GPU RAM correctly by calling **cudaFree**.

There are two users on our test systems, *Alice* and *Mal*. *Alice* executes a program, which writes a fixed user specified string to GPU RAM. After the program terminated, *Mal* executes a program, which computes the available GPU RAM by calling **cudaMemGetInfo** and then allocates the available memory[7]. Afterwards, the allocated memory is copied from GPU RAM to CPU RAM by using **cudaMemcpy**. Then, the program searches in the acquired memory dump for the string previously written by the program that *Alice* executed.

We repeated this experiment 1000 times with strings of different lengths on the test machines. *Mal*'s program found the string written by *Alice*'s program in all cases. We used the secure password generator of openssl[8] to generate the random strings and varied their lengths between 10 and 10000 characters in steps of 10. Note, that the user *Alice* used program A to write a string into GPU RAM, whereas user *Mal* used program B to create a memory dump. Both users had different user accounts, and none of the users had root access. Hence, we conclude that program B can extract data written to GPU RAM by a prior executed program A, even for different user accounts without root access. Therefore, we found a severe security issue for GPU coprocessing in applications in general and, therefore, for GPU accelerated database systems. This observation can be utilized for forensic investigations, as well.

4.3 Application of Forensic Model

With regards to the model of the forensic process introduced in previous work [KHDV09], the GPU RAM represents a subcomponent of the main memory. The GPU RAM shares the main characteristics of CPU RAM as a source of highly volatile data whose content is easily lost (e.g., due to a loss of power) and easily altered (e.g., by the re-use of that memory after deallocation). The described combination of **cudaMemGetInfo** and **cudaMemcpy** represents two phases. The first one is *data gathering DG* for the creation of the dump by creating *raw data* DT_1 using *forensic methods* supplied by the *operating system OS*.

[7] We have to subtract a small amount of memory (around 1MB), because the CUDA runtime does not permit to allocate the complete free space.

[8] `http://www.openssl.org/docs/apps/openssl.html`

The second part of the experiment, the string search, represents the phase of *data investigation DI* with respect to the text string search by using the *forensic method* supplied by the *data presentation and evaluation DPE*, yielding *user data* DT_8. However, those techniques used in the first experiments would not qualify as forensic methods according to the demands placed on forensic examinations (e.g., by the CFA [Dar10], see also Section 2.2) by, e.g., not assuring integrity and authenticity. The first results outline the need for further research, which in turn can be used to create forensic methods that do comply with the requirements of a forensic examination.

5 Discussion

As observed in Section 4, it is possible to create a memory dump of the available space from the GPU RAM. Furthermore, user privileges/access control can be bypassed and data processed by previous executed programs can be recovered. In this Section, we discuss the consequences of our evaluation results from two points of view. First, we treat our results as security threat for database systems, which needs to be addressed to prevent unauthorized data access. Second, we discuss our results from the view of forensic investigations to analyze incidents.

5.1 Secure Coprocessing – Anti-Forensic

Using the results of our experiments, we can conclude that DBMS access control can be bypassed for GPU coprocessing scenarios. This is because data managed by a DBMS can be retrieved by processes from other users without necessary access rights. Therefore, we need a special free function that deletes data forensically secure to address the **Recoverability of Data** challenge from Section 3.1.

We use the information provided by Gutmann [Gut01] to justify our procedures and thus to avoid the extraction of possible confidential data from GPU RAM by other programs. The main memory in general and the memory accessed by the GPU typically consists of a large array of DRAM memory cells with only the latest data fed to an internal latch (cache) of SRAM. Since the cache outputs the requested data, the retention effect of stored data inside DRAM-only cells can be neglected. There is little chance of any piece of data except the last one read DRAM accesses remaining in those latches for more than an instant, because these latches are shared across the entire DRAM [Gut01]. Thus, writing random content or zeros (for easier fabrication) should be sufficient to erase the data previously contained in a given section of main memory in general and GPU memory, in particular[9]. Therefore, we propose a secure **cudaFree** function, which first writes random values or zeros to the memory area, which is to be freed and then calls **cudaFree**. Using this approach, one can effectively avoid data extraction by other programs. Due to the parallel nature of GPUs, the GPU RAM can be concurrently overwritten using thousands or ten-thousands of threads.

[9] Additional measures in a DRAM-only scenario are outlined by Gutmann [Gut01]

As the goal is the destruction of confidential data, a synchronization between threads is not necessary. Hence, data in GPU RAM can be overwritten with lower latency than in CPU RAM.

5.2 Law Enforcement – Forensic Methods

Our results also supports forensic investigations. If an application uses the GPU to accelerate computations, it is likely that the object of investigation can be located in the GPU RAM. Hence, research is necessary as to, which methods can be applied to analyze incidents. As we discovered, one method is simply a creation of a GPU RAM dump. As long as the application of interest has freed its used GPU RAM, it is possible to recover its data processed on the GPU. However, this approach is limited, because we are still unable to investigate memory that is still allocated by the application of interest. Hence, we successfully addressed the **Recoverability of Data** challenge from Section 3.2.

Concerning the experiments regarding the GPU subsystem and in reference to the Cyber Forensic Assurance [Dar10] it can be noted that so far, only the component III (b), "Utility/Relevance - Is it useful/is it the right information?" could partly be shown. In order to become an accepted part of the sets of forensic methods, all the other components need to be addressed, outlining the next steps towards the next necessary research.

One scenario for forensic investigation of GPU RAM arises, when the GPU is utilized by an attacker to intrude in a system, e.g., by using a GPU accelerated password cracker such as John the Ripper[10]. To be able to collect all evidence relevant to an investigation, the GPU RAM has to be considered. Our findings on GPU RAM investigation are a first step to include the GPU in forensic investigations.

6 Related Work

One example for software capable of CPU RAM forensics is the Forensic Analysis Toolkit (FAT-Kit) [PWFA06]. FAT-Kit is designed to support a forensic examination and assumes that the phase of data gathering (i.e., the acquisition of raw data) is already finished and a low-level image of the main memory already exists. FAT-kit, amongst others, aims at identifying structures from the C-programming language and the re-assembly of known address spaces and visualizing the informations that are extracted.

To the best of our knowledge, there is no noticeable previous work on forensic investigations on GPU RAM in the context of database coprocessing on GPUs. Hence, no tool exists, which can fulfill the legal requirements for a fully fledged forensic workflow for GPU RAM.

However, approaches exist for accelerating forensic analysis using GPUs. Jacob and Brodley presented an approach that outsources computations of an intrusion detection system

[10] `http://www.openwall.com/john/`

[JB06]. Marziale et al. investigated GPU accelerating of forensic tools in general and found that GPU coprocessing is beneficial for binary string search, an important operation for forensic investigations [MRR07].

7 Future Work

We only considered the global device memory in this paper. Our investigation has to be extended to the recoverability of data in different GPU memories such as local, constant, texture or shared memory, which are frequently utilized to further improve the performance of GPU algorithms [SK10]. Furthermore, we plan to repeat our evaluation using other GPGPU frameworks such as OpenCL, AMD APP SDK or abstractions of CUDA, such as Thrust[11]. Additionally, we investigate the applicability of other forensic methods for CPU RAM on GPU RAM, e.g., complete and forensically sound data gathering [Sch08], automated interpretation of data structures [PWFA06], and address space reconstruction [PWFA06].

8 Conclusion

In this paper, we addressed the problem of secure data processing for General Purpose Computing using Graphics Processing Units (GPGPU) in the context of database systems. We identified challenges for secure data processing as well as forensic investigations if the GPU is used as coprocessor in database systems. We discovered that application data can be easily accessed by other programs as soon as the allocated GPU memory is freed by an application. However, we did not discover a method to investigate application data in GPU RAM that is still allocated by the application. Furthermore, we discuss how our results can be used from a forensic and an anti-forensic point of view. Note that our results are not limited to database systems, but are applicable to all applications using CUDA C.

Acknowledgements

The work in this paper has been funded by the German Federal Ministry of Education and Science (BMBF) through the Research Program under Contract No. FKZ: 13N10817 and 13N10818. We thank Ingolf Geist and Siba Mohammad for helpful feedback and discussions as well as the reviewers for their constructive criticism.

[11] `http://thrust.github.com/`

References

[AW10] Witold Andrzejewski and Robert Wrembel. GPU-WAH: Applying GPUs to Compressing Bitmap Indexes with Word Aligned Hybrid. In *DEXA (2)*, pages 315–329, 2010.

[Bee09] Nicole Beebe. Digital Forensic Research: The Good, the Bad and the Unaddressed. In *Advances in Digital Forensics V*, volume 306 of *IFIP Advances in Information and Communication Technology*, pages 17–36. Springer, 2009.

[Bis02] Matthew Bishop. *The Art and Science of Computer Security*. Addison-Wesley, 2002.

[BS10] Peter Bakkum and Kevin Skadron. Accelerating SQL database operations on a GPU with CUDA. In *GPGPU*, pages 94–103. ACM, 2010.

[CGS09] Michael Cohen, Simson Garfinkel, and Bradley Schatz. Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow. *Digit. Investig.*, 6:S57–S68, 2009.

[Cod82] E. F. Codd. Relational database: a practical foundation for productivity. *Commun. ACM*, 25(2):109–117, 1982.

[Dar10] Glen S. Dardick. Cyber Forensics Assurance. In Andrew Woodward, editor, *Proceedings of the 8th Australian Digital Forensics Conference*, pages 57–64. School of Computer and Information Science, Edith Cowan University, Perth, Western Australia,, 2010.

[DG01] Lloyd Dixon and Brian Gill. *Changes in the Standards for Admitting Expert Evidence in Federal Civil Cases Since the Daubert Decision*. RAND Institute for Civil Justice, 2001. ISBN: 0-8330-3088-4.

[DWL+12] Gregory Diamos, Haicheng Wu, Ashwin Lele, Jin Wang, and Sudhakar Yalamanchili. Efficient Relational Algebra Algorithms and Data Structures for GPU. Technical report, Center for Experimental Research in Computer Systems (CERS), 2012.

[GGKM06] Naga Govindaraju, Jim Gray, Ritesh Kumar, and Dinesh Manocha. GPUTeraSort: High Performance Graphics Coprocessor Sorting for Large Database Management. In *SIGMOD*, pages 325–336. ACM, 2006.

[Gut01] Peter Gutmann. Data remanence in semiconductor devices. In *Proceedings of the 10th conference on USENIX Security Symposium - Volume 10*, SSYM'01, pages 4–4, Berkeley, CA, USA, 2001. USENIX Association.

[HLY+09] Bingsheng He, Mian Lu, Ke Yang, Rui Fang, Naga K. Govindaraju, Qiong Luo, and Pedro V. Sander. Relational Query Coprocessing on Graphics Processors. *ACM Trans. Database Syst.*, 34:21:1–21:39, 2009.

[HM12] Max Heimel and Volker Markl. A First Step Towards GPU-assisted Query Optimization. In *ADMS*, 2012.

[HY11] Bingsheng He and Jeffrey Xu Yu. High-Throughput Transaction Executions on Graphics Processors. *PVLDB*, 4(5):314–325, 2011.

[HYF+08] Bingsheng He, Ke Yang, Rui Fang, Mian Lu, Naga Govindaraju, Qiong Luo, and Pedro Sander. Relational Joins on Graphics Processors. In *SIGMOD*, pages 511–524. ACM, 2008.

[JB06] Nigel Jacob and Carla Brodley. Offloading IDS Computation to the GPU. In *ACSAC*, pages 371–380. IEEE, 2006.

[KHDV09] Stefan Kiltz, Tobias Hoppe, Jana Dittmann, and Claus Vielhauer. Video surveillance: A new forensic model for the forensically sound retrival of picture content off a memory dump. In Stefan Fischer, Erik Maehle, and Ruediger Reischuk, editors, *Proceedings of Informatik2009 - Digitale Multimedia-Forensik*, pages 1619–1633, 2009.

[KLMV12] Tim Kaldewey, Guy Lohman, Rene Mueller, and Peter Volk. GPU Join Processing Revisited. In *DaMoN*, pages 55–62. ACM, 2012.

[LDKA10] Tobias Lauer, Amitava Datta, Zurab Khadikov, and Christoffer Anselm. Exploring Graphics Processing Units as Parallel Coprocessors for Online Aggregation. In *DOLAP*, pages 77–84. ACM, 2010.

[MHS+11] Roger Moussalli, Robert Halstead, Mariam Salloum, Walid Najjar, and Vassilis J. Tsotras. Efficient XML Path Filtering Using GPUs. In *ADMS*, 2011.

[MRR07] Lodovico Marziale, Golden G. Richard, III, and Vassil Roussev. Massive threading: Using GPUs to increase the performance of digital forensics tools. *Digit. Investig.*, 4:73–81, 2007.

[NVI12a] NVIDIA. NVIDIA CUDA C Programming Guide. `http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf`, 2012. pp. 30–34, Version 4.0, [Online; accessed 1-May-2012].

[NVI12b] NVIDIA. NVIDIA CUDA Library Documentation. `http://www.clear.rice.edu/comp422/resources/cuda/html/index.html`, 2012. Version 4.0, [Online; accessed 30-October-2012].

[OLG+07] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krger, Aaron E. Lefohn, and Timothy J. Purcell. A Survey of General-Purpose Computation on Graphics Hardware. *Computer Graphics Forum*, 26(1):80–113, 2007.

[Pir12] Holger Pirk. Efficient Cross-Device Query Processing. *Proceedings of the VLDB Endowment*, 2012.

[PMK11] Holger Pirk, Stefan Manegold, and Martin Kersten. Accelerating Foreign-Key Joins using Asymmetric Memory Channels. In *ADMS*, pages 585–597. VLDB Endowment, 2011.

[Pol07] Mark M. Pollitt. An Ad Hoc Review of Digital Forensic Models. In *Systematic Approaches to Digital Forensic Engineering, 2007. SADFE 2007. Second International Workshop on*, pages 43–54, 2007.

[PSMK12] Holger Pirk, Thibault Sellam, Stefan Manegold, and Martin Kersten. X-Device Query Processing by Bitwise Distribution. In *DaMoN*, pages 48–54. ACM, 2012.

[PWFA06] Nick L. Petroni, Jr., Aaron Walters, Timothy Fraser, and William A. Arbaugh. FATKit: A framework for the extraction and analysis of digital forensic data from volatile system memory. *Digit. Investig.*, 3(4):197–210, December 2006.

[Sch08] Andreas Schuster. The impact of Microsoft Windows pool allocation strategies on memory forensics. *Digital Investigation*, 5, Supplement(0):58–64, 2008. The Proceedings of the Eighth Annual DFRWS Conference.

[SK10] Jason Sanders and Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley, 1st edition, 2010.

[WWN+10] Slawomir Walkowiak, Konrad Wawruch, Marita Nowotka, Lukasz Ligowski, and Witold Rudnicki. Exploring Utilisation of GPU for Database Applications. *Procedia Computer Science*, 1(1):505–513, 2010.

The Monetary Value of Information: A Leakage-Resistant Data Valuation

Stefan Barthel, Eike Schallehn
Institute of Technical and Business Information Systems,
Otto-von-Guericke-University Magdeburg, Germany,
{stefan.barthel, eike.schallehn}@ovgu.de

Abstract: The importance of information as a main asset of a company or organization is widely acknowledged nowadays. The loss of or the unauthorized access to sensitive information are critical and can possibly send a company into bankruptcy. Furthermore, the risk of information larceny is most often not caused by a direct attack of unauthorized outsiders, but by authorized extractions by malicious or unaware insiders passing data to unauthorized outsiders. Unfortunately, this problem cannot be solved by the typically used role-based authentication. The detection of malicious accesses based on typical access characteristics, which has inspired some research, is limited in its potential. Therefore, we present a conceptual approach based on the valuation of information, i.e., using a description of the actual worth of data items within database systems. This allows to rate potential losses on the fly as well as preventing valuable extractions done by insiders. In detail, we describe a mechanism called leakage-resistant data valuation that calculates a monetary value for every query and takes according action if the cumulated monetary value exceeds a threshold (per query or per time span).

1 Introduction

In current, worldwide distributed information systems, sensitive data needs to be stored securely and well protected to prevent malicious use [VBF+04]. One example is the misuse of bank account information, which we will focus on to illustrate our proposal. In the area of financial institutes, highly sensitive information are stored compactly. Authorized users are querying sensitive data every day, while they enjoy an enormous trust from the management. Other fields with highly sensitive data include health care, law, government, industrial research, and military applications. Because sensitive data are so valuable, several organizations, individuals, and even governments are interested in data extracts, in some cases regardless whether they are obtained with legal or illegal methods. For example, countries that assume tax evasion by their inhabitants in countries with high bank secrecy, are interested in getting additional evidence of tax evaders, because it can lead to millions of additional tax income. Therefore, buying stolen data is feasible, even it is prohibited in the country of origin. In the case of Germany, which bought several extracts of tax evaders in the recent past, the stolen data was almost always sold on a CD, probably extracted and

copied by a member of staff or someone who has access to the system. This scenario is generally referred to as the *insider threat*. It is generally accepted that the cost of damages caused by insider threats exceeds those of outsider threats [HSRE10]. That means, even if several security restrictions and authorization based access controls are active, at least one person has the ability to access all the data – in most cases the system administrator. Therefore, from our point of view, the conventional way of access authorization has to be thoroughly rethought for sensitive data.

In this paper, we introduce our approach based on an actual monetary value of the data any user is allowed to work with. We choose the mapping of monetary value and information, because it is intuitively accessible when speaking about a certain valuation of information. By specifying two thresholds and a monetary value for every attribute, the system is enabled to easily react in two ways:

- alerting and logging, if single queries or cumulated accesses of one user exceed a certain threshold, or
- truncating the result set, i.e., limiting the output of sensitive data, if a further and more rigid threshold is exceeded.

For example, a bank clerk will be able to access personal and bank information of individual customers, but as there is no need to extract many records at once or in a short period of time, this can be restricted by our approach. Nevertheless, the database management system will be able to backup data and have the ability to read and write data according to fixed application purposes.

2 Preliminary Considerations

Security mechanisms for conventional databases are divided into three layers: physical security, operating system security, and database management system security [BS05]. Because the database management system (DBMS) is the primary line of defense in constraining access to the data stored in a database, we focus on access control models of the DBMS. Established access control models can be grouped into three classes: Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role Based Access Control (RBAC) [SS94]. They are widely used, well investigated, and have been proven to be useful, as long as the data is accessed using mechanisms of the DBMS. Especially RBAC is an effective way to scale access control in large systems [PG06]. However, access control is useless if an attacker gains direct access to the database or at least the hard disk drive. An intruder could mine a database footprint on disk and simply copy the plain text. Encryption can solve those issues and protect the data against unauthorized, physical reads and writes [BG09]. In addition, encryption can solve the problem of data backup theft [SVEG10].

Nevertheless, access control as well as encryption are not restrictive and protective enough to guarantee that sensitive data can not be queried if an insider or an outsider,

who gained access rights, extracts data. This threat is most often caused by current or former contractors, employees, or even business partners who have or had authorized access to an organization's network and want to take advantage of it [GSB+04]. To ensure a maximum of protection – even if a total protection will never exist – this insider threat should be reduced to a minimum. We will illustrate that the protection level can be significantly improved by introducing the relatively light-weight concept of a leakage-resistant data valuation.

3 Leakage-Resistant Data Valuation

Our proposed approach is a valuation-based extension of the conventional access control for the relational database model, which calculates for every query the total monetary value depending on queried tables and attributes as well as used operations. In detail, every attribute $A_i \in R$ of a base relation schema R is valuated by a certain monetary value ($mval(A_i) \in \mathbb{R}$). With these attribute valuations, we derive a monetary value for one tuple $t \in r(R)$ given by Equation (1), as well as the total monetary value of the relation $r(R)$ given by Equation (2), if data is extracted by a query.

$$mval(t \in r(R)) = \sum_{A_i \in R} mval(A_i) \tag{1}$$

$$mval(r(R)) = \sum_{t \in r(R)} mval(t) = |r(R)| * mval(t \in r(R)) \tag{2}$$

We are considering (1) the quality of information (monetary value of attributes) as well as (2) quantity of information (number of tuples). Based on this, query results derived by operations need to be considered and valuated depending on type and granularity. For these purposes we decide, that operations which enhance the information content should increase (e.g., joins), whereas coarsening operations (e.g., aggregation), which reduce the number of rows or columns, should decrease the total monetary value of the resulting data. However, the calculated monetary value of one tuple is less than of a tuple with the same number of attributes, but at least as of one aggregated value, because the information content increases due to aggregation. For the sake of simplicity, we propose to multiply the monetary value of an aggregated attribute by a certain factor (e.g., 2), regardless of the type of aggregation (e.g., AVG, SUM, MAX, etc.). In case of a joined tuple of several tables, all monetary values of included attributes are summarized. The cumulated total monetary value of a query is afterwards compared to two thresholds.

Suspicious threshold: $thr_{susp} \in \mathbb{R}$ is used to identify queries with a monetary value which is suspicious, i.e., could indicate malicious behavior, and these are written to an alert log and handled accordingly.

Truncate threshold: $thr_{trun} \in \mathbb{R}$ is used to identify malicious queries directly. Queries with a higher total monetary value are truncated when exceeding this boundary. In addition, truncated queries also need to be logged, therefore the truncate threshold is set to a higher value than the suspicious threshold ($thr_{susp} < thr_{trun}$).

Alert log: The *alert log* accordingly captures every exceedance and is written continuously to allow a tracing of violations over time. A maintenance job periodically analyses the captured discrepancies and informs responsible security guards – a number of persons that are delegated (at least four-eye-principle).

To prevent valuable extractions by one query and several queries within a short period of time, we do not only (1) truncate or log a single query that extracts to much information, but also (2) prevent a sequence of queries with the same intention. Hence, we cumulate the monetary values of previous non-suspicious queries ($mval(r(R_i)) < thr_{susp}$) per user, to measure how much information was extracted over a period of time. Because we are just interested in a certain time interval, the user valuation is reset periodically (e.g., hourly, daily, weekly, etc.). The mechanism of logging and truncating when reaching thresholds is also used for the time restricted user valuation. With this determination we are facing line by line extractions, e.g., by a batch script. Additionally, we recommend that the calculation of the total monetary value of a query is done before physically querying the result set. Due to the fact that monetary values of attributes and tuples as well as both thresholds are known, by pre-calculating, there is no overhead of joining tuples, which will not be displayed anyway. To do so, we calculate the monetary value of one result tuple including every attribute (shown in Eq. (1)) and we derive the maximum releasable tuples ($tpl_{max} \in \mathbb{N}$) for a certain result relation $r(R)$:

$$tpl_{max}(r(R)) = \left\lceil \frac{thr_{trun}}{mval(t \in r(R))} \right\rceil \tag{3}$$

To inform the requester of a query, that the result set was shortened because the truncate threshold was exceeded, an information is displayed. The truncation only applies to commands which create displayed results. Others, like backup processes, data modifications, etc., could for instance be privileged by a whitelist. We see our approach as an extended access control mechanism and classify it as active access control regarding to [PG06], because it provides continuous and dynamic access control beyond initial access control decisions. For implementation purposes, monetary values for distinct attribute may differ and need to be set manually while defining the table schema. Moreover, monetary values of tuples are derived from attributes and do not need to be set at all. The suspicious threshold as well as the truncate threshold are disabled and set by default ($thr_{susp} = thr_{trun} = \infty$), but can be easily activated.

To be able to define monetary values for attributes and thresholds, we propose an extension of SQL by adding thresholds as system variables and monetary values of attributes as extra options within the table definition. An example of syntax to define several monetary values for attributes while creating a table and enabling both thresholds are shown below:

```
suspicious_valuation=2000
truncate_valuation=4000

CREATE TABLE table_1
(
  attribute_1 INT PRIMARY KEY MVAL 0.1,
  attribute_2 UNIQUE COMMENT 'important' MVAL 10,
  attribute_3 DATE
);
```

Within this code example attribute 1 and 2 are directly set by definition, while attribute number 3 is set to the default value ($mval(A_3) = 0$) by the system. With this determination, we can ensure, that a DBMS with an implemented leakage-resistant data valuation, but without any set valuations, will exactly work like a conventional, non-modified DBMS. Consequently, it is also possible to modify those valuations by using an ALTER-command of SQL. All data valuations are stored in the data dictionary, because it is a centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format [IBM93]. However, at runtime all valuations should be held in main memory to ensure a maximum of performance.

4 Related Work

Conventional database management systems mostly use access control models to face unauthorized access on data. However, these are insufficient when an authorized individual extracts data regardless whether she is the owner or has stolen that account. Several methods were conceived to eliminate those weaknesses. We refer to Park and Giordano [PG06], who give an overview of requirements needed to address the insider threat.

Authorization views partially achieve those crucial goals of an extended access control and have been proposed several times. For example, Rizvi et al. [RMSR04] as well as Rosenthal et al. [RS00] use authorization-transparent views. In detail, incoming user queries are only admitted, if they can be answered using information contained in authorization views. Contrary to this, we do not prohibit a query in its entirety. Another approach based on views was introduced by Motro [Mot89]. Motro handles only conjunctive queries and answers a query only with a part of the result set, but without any indication why it is partial. We do handle information enhancing (e.g., joins), as well as coarsening operations (e.g., aggregation) and we do display a user notification. All authorization view approaches require an explicit definition of a view for each possible access need, which also imposes the burden of knowing and directly querying these views. In contrast, the monetary values of attributes are set while defining the tables and the user can query the tables or views she is used to. Comparison methods are also not addressing the problem of how to keep up to date,

because new operators and constraint constructs are repeatedly evolving. Moreover, the equivalence test of general relational queries is undecidable and equivalence for conjunctive queries is known to be NP complete [CM77]. Therefore, the leakage-resistant data valuation is more applicable, because it does not have to face those challenges.

However, none of these methods does consider the sensitivity level of data that is extracted by an authorized user. In the field of *privacy-preserving data publishing* (PPDP), on the contrary, several methods are provided for publishing useful information while preserving data privacy. In detail, multiple security-related measures (e.g., k-anonymity [Swe02], l-Diversity [MKGV07]) have been proposed, which aggregate information within a data extract in a way that they can not lead to an identification of a single individual. We refer to Fung et al. [FWCY10], who give a detailed overview of recent developments in methods and tools of PPDP. However, these mechanisms are mainly used for privacy-preserving tasks and are not in use when an insider accesses data. Moreover, privacy preserving mechanisms are not applicable for our scenario, because they do not consider a line by line extraction over time as well as the information loss by aggregating attributes of the result set, that the user is allowed to see.

To the best of our knowledge, there is only the approach of Harel et al. ([HSRE10], [HSRE11], [HSRE12]) that is comparable to our data valuation to prevent suspicious, authorized data extractions. Harel et al. introduce the *Misuseability Weight (M-score)* that describes the sensitivity level of the data exposed to the user. Hence, Harel et al. focus on the protection of the quality of information, whereas our approach predominantly preserves the extraction of a collection of data (quantity of information). Harel et al. also do not consider extractions over time, logging of malicious requester and the backup process. In addition, mapping attributes to a certain monetary value is much more applicable and intuitional, than mapping to a artificial M-score. We also suppose a better performance for our approach, because we can calculate the maximal size of a result set before physically querying.

Our extended authorization control does not limit the system to a simple query-authorization control without any protection against the insider threat, rather we allow a query to be executed whenever the information carried by the query is legitimate according to the specified authorizations and thresholds.

5 Conclusion and Future Work

We presented the leakage-resistant data valuation as an additional layer in a security defense model for raw data. Within this security defense model each layer functions as a separate firewall and all of the layers are ordered like onion skins with the raw data as base (see Fig. 1). This principle is called defense in depth, because attackers must get through layer after layer of defense to reach the base – the raw data. The data valuation layer is inserted below the conventional access control

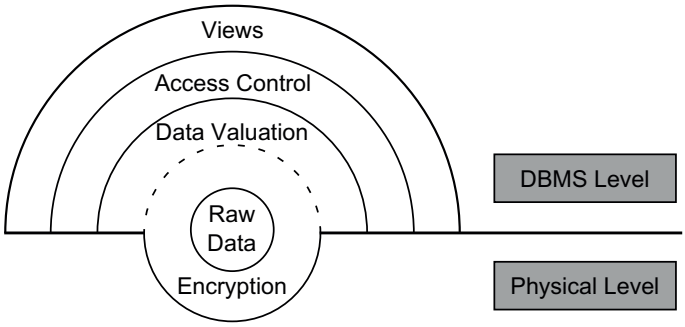


Figure 1: Security defense model on DBMS and physical level

layer and therefore more powerful and restrictive. On the top of access control methods, it is common practice to define views which give users access to a specific portion of data without having direct access rights. Encryption on the contrary, is situated below our data valuation layer, because it uses DBMS-, file-, application- or client side encryption to protect the raw data [SVEG10]. In future work we will implement our leakage-resistant data valuation approach and evaluate performance as well as usability issues. There will also be research on modifying our quantity-cutting mechanism to a quality-cutting one, where data will be masked instead of truncated. Furthermore, we will publish a more detailed description of how to treat various operations (e.g., different joins, set operation, etc.), procedures, and functions. It needs to be investigated how to set the monetary values and thresholds effectively, regarding to various usage scenarios. Another field of application is a self-protecting, leakage-resistant data valuation, where the system will set various parameters by itself, collect user profiles, and compare them to a malicious behavior that was identified earlier. With this extension, we will also be able to establish an anomaly detection or at least enhance an existing anomaly detection system.

Acknowledgments: This research has been funded in part by the German Federal Ministry of Education and Science (BMBF) through the Research Program under Contract FKZ: 13N10817. We thank Ingolf Geist for his support.

References

[BG09] Luc Bouganim and Yanli Guo. Database Encryption. In *Encyclopedia of Cryptography and Security*, pages 1–9. Springer, 2009.

[BS05] Elisa Bertino and Ravi Sandhu. Database Security - Concepts, Approaches, and Challenges. *IEEE Dependable and Secure Comp.*, 2(1):2 –19, March 2005.

[CM77] Ashok K. Chandra and Philip M. Merlin. Optimal Implementation of Conjunctive Queries in Relational Data Bases. In *Proceedings of the ninth annual ACM symposium on Theory of computing*, STOC'77, pages 77–90. ACM, 1977.

[FWCY10] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. Privacy-Preserving Data Publishing: A Survey of Recent Developments. *ACM Comput. Surv.*, 42(4):14:1–14:53, June 2010.

[GSB+04] Richard Guida, Robert Stahl, Thomas Bunt, Gary Secrest, and Joseph Moorcones. Deploying and Using Public Key Technology: Lessons Learned in Real Life. *IEEE Security Privacy*, 2(4):67 – 71, August 2004.

[HSRE10] Amir Harel, Asaf Shabtai, Lior Rokach, and Yuval Elovici. M-score: Estimating the Potential Damage of Data Leakage Incident by Assigning Misuseability Weight. In *Proceedings of the 2010 ACM workshop on Insider threats*, Insider Threats'10, pages 13–20. ACM, 2010.

[HSRE11] Amir Harel, Asaf Shabtai, Lior Rokach, and Yuval Elovici. Eliciting Domain Expert Misuseability Conceptions. In *Proceedings of the sixth international conference on Knowledge capture*, K-CAP'11, pages 193–194. ACM, 2011.

[HSRE12] Amir Harel, Asaf Shabtai, Lior Rokach, and Yuval Elovici. M-Score: A Misuseability Weight Measure. *IEEE Trans. Dependable Secur. Comput.*, 9(3):414–428, May 2012.

[IBM93] Corporation IBM. *IBM Dictionary of Computing*. McGraw-Hill, Inc., New York, NY, USA, 10th edition, 1993.

[MKGV07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-Diversity: Privacy Beyond k-Anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):1–50, March 2007.

[Mot89] Amihai Motro. An Access Authorization Model for Relational Databases Based on Algebraic Manipulation of View Definitions. In *Proceedings of the Fifth International Conference on Data Engineering*, pages 339–347. IEEE Computer Society, 1989.

[PG06] Joon S. Park and Joseph Giordano. Access Control Requirements for Preventing Insider Threats. In *Proceedings of the 4th IEEE international conference on Intelligence and Security Informatics*, ISI'06, pages 529–534. Springer, 2006.

[RMSR04] Shariq Rizvi, Alberto Mendelzon, S. Sudarshan, and Prasan Roy. Extending Query Rewriting Techniques for Fine-Grained Access Control. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, SIGMOD'04, pages 551–562. ACM, 2004.

[RS00] Arnon Rosenthal and Edward Sciore. View Security as the Basis for Data Warehouse Security. In *CAiSE Workshop on Design and Management of Data Warehouses*, DMDW'2000, pages 5–6. CEUR-WS, 2000.

[SS94] Ravi S. Sandhu and Pierangela Samarati. Access Control: Principle and Practice. *IEEE Communications Magazine*, 32(9):40–48, September 1994.

[SVEG10] Erez Shmueli, Ronen Vaisenberg, Yuval Elovici, and Chanan Glezer. Database Encryption: An Overview of Contemporary Challenges and Design Considerations. *SIGMOD Rec.*, 38(3):29–34, December 2010.

[Swe02] Latanya Sweeney. k-Anonymity: A Model For Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[VBF+04] Vassilios S. Verykios, Elisa Bertino, Igor N. Fovino, Loredana P. Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-Art in Privacy Preserving Data Mining. *SIGMOD Rec.*, 33(1):50–57, March 2004.

Workshop on Information Systems in Digital Engineering (ISDE)

Maik Mory, Veit Köppen, Gunter Saake

Center for Digital Engineering (CDE)
Otto-von-Guericke-Universität Magdeburg
Universitätsplatz 2
D-39106 Magdeburg

firstname.lastname@ovgu.de

Message from the Chairs

The Workshop on Information Systems in Digital Engineering (ISDE) is held the first time in conjunction with the 15th Conference on Database Systems, Technology, and Web (BTW 2013) at the Otto-von-Guericke-University Magdeburg on March the 12th, 2013. The ISDE brings together two vibrant communities: the database and information systems domain, and the digital engineering domain. We expect to establish cooperation and collaboration in theory and practice.

Today, most engineers are supported by a software landscape. Information systems are a natural fit. Within the supporting software landscape, more and more data management technologies are embedded. Digital engineering is an emerging trend that aims at bringing together traditional engineering (e.g., electrical engineering and mechanical engineering) with modern approaches in software and systems engineering. Today's extension of data management to the domain of distributed realtime and embedded systems raises new questions and challenges. Fortunately, research and development of databases and information systems have a long and rich history as well as new technologies in hard- and software.

The contributions to this workshop are related to automotive electrical engineering, virtual reality in engineering, early stages in product development, and safety and hazard defense engineering. Anybody comparing the papers will notice that they altogether connect distributed clients by means of information system technology. Consequently, they consider various aspects of interoperability.

Tiedeken, Herbst, and Reichert [THR13] examine the software landscape in product data management of electrical and electronic, automotive components (E/E-PDM). They focus on partial information integration about products' dependencies from

heterogeneous data sources. The identified challenges account for software evolution and variable organizational constraints. Their main contribution is a set of requirements for information integration in E/E-PDM.

Broneske, Schäler, and Grebhahn [BSG13] tackle a long-running problem in database operation with virtual reality. Inadequate selection or implementation of index structures may cause unresponsive behavior of databases and thus become a showstopper when embedded in real-time systems. Therefore, they instrument the database system minimally invasive and visualize the gained data in a virtual space. The presented software prototype is a sound foundation for more experiments.

Oellrich and Mantwill [OM13] consider collaborative engineering and open innovation with web based technologies. On the one hand, today's popular web based services show that near-to-interactive collaboration is feasible. On the other hand, methodological development approaches define their very own notations and processes. Thus, the authors put a web based frame around methodological product development. Their most recognizable innovation is a closer look at a web based implementation of the morphological box.

Mann, Gurath, Stahn, and Gülde [MGSG13] introduce web based information systems to a domain, which is dominated by unstructured, office-oriented information and paper till this day. Determining the minimum required amount of civil defense resources is a common municipal task. A web based information system that implements the empirical mathematical risk analysis method is proposed. Challenges and benefits are discussed for firefighters, the municipal administration, and citizens.

We are deeply grateful to everyone who made this workshop possible – the authors, the reviewers, the BTW team, and all participants.

References

[THR13] Tiedeken, J.; Herbst, J.; Reichert, M.: Managing Complex Data for Electrical/Electronic Components: Challenges and Requirements. In (G. Saake et al. (Hrsg.)): Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband, Magdeburg 2013.

[BSG13] Broneske, D.; Schäler, M.; Grebhahn, A.: Extending an Index-Benchmarking Framework with Non-Invasive Visualization Capability. In (G. Saake et al. (Hrsg.)): Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband, Magdeburg 2013.

[OM13] Oellrich, M.; Mantwill, F.: Concept for a web based Support of the Product Development Process. In (G. Saake et al. (Hrsg.)): Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband, Magdeburg 2013.

[MGSG13] Mann, S.; Gurath, C.; Stahn, S.; Gülde, R.: EMRAgis – Analytical Risk Assessment Tool for Community Risk Reduction. In (G. Saake et al. (Hrsg.)): Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband, Magdeburg 2013.

Managing Complex Data for Electrical/Electronic Components: Challenges and Requirements

Julian Tiedeken[1], Joachim Herbst[2], and Manfred Reichert[1]

[1]Institute of Databases and Information Systems, Ulm University, Germany
{julian.tiedeken, manfred.reichert}@uni-ulm.de
[2]Daimler AG, Germany
joachim.j.herbst@daimler.com

Abstract: In the automotive domain, innovation is driven by the introduction and continuous improvement of electrical and electronic (E/E) components (e.g. sensors, actuators, and electronic control units). This trend is accompanied by increasing complexity of interdependencies between these E/E components. In addition, external impact factors (e.g. changes of regulations) demand for more sophisticated E/E product data management (E/E-PDM). Since E/E product data is scattered over distributed heterogeneous IT systems, application-spanning use cases (e.g. consistency of artifacts, plausibility of logical connections between electronic control units) are difficult to realize. Consequently, the partial integration of the corresponding application data models becomes necessary. Evolution of application data models is common in the context of E/E-PDM, but not considered by existing application integration approaches. Furthermore, no methodology for realizing application integration models exists. This paper elaborates the challenges to be tackled when integrating E/E product data from different applications. It further presents properties of the IT landscape involved in E/E-PDM and reveals occurring problems. Finally, requirements for E/E-PDM are discussed.

1 Introduction

A modern car consists of up to 70 *electronic control units* (ECUs) [MHHR06]. New functionality, product innovation, and customer requirements continuously increase the complexity as well as the number of ECUs in the car. Due to country-, market-, and customer-specific requirements, in addition, a large number of variants of electrical/electronic (E/E) product data exists [HBR08]. In this context, simultaneous engineering, manufacturer-supplier-relationships, and development processes show complex interdependencies (cf. Fig. 1). In particular, E/E product data refers to digital development artifacts, e.g. requirements, circuit diagrams, wiring diagrams, bootloader, flashware, and software. The latter are created and managed by autonomous information systems satisfying different requirements. Legal requirements (e.g., product liability (ISO 26262 [fS11]) and envi-

ronmental regulations (RoHS[1], WEEE[2], EuP[3]) demand for traceability and verification of development decisions. Hence, any change of E/E product data must be documented. For this purpose, E/E product data management (E/E-PDM) systems maintain configurations that bundle interrelated E/E products occuring in all development phases. For example, basic information include the developer or supplier of an E/E product (e.g., ECU, sensor, or actuator), change requestor, and change reason.

Each E/E product has a lifecycle that comprises the steps required to create and distribute it, e.g. planning, development, production, and after sales. In particular, product development constitutes an integral part of the product lifecycle. In this context, emerging market trends (i.e. green-mobility, e-mobility), increasing product liability, environmental regulations, tighter integration of suppliers, and shorter development cycles demand for high data quality as well as integrated E/E product development processes. To cope with these challenges, an IT landscape must adequately cope with product changes. In current practice, a multitude of heterogeneous applications are used for accomplishing different development tasks. In turn, this results in numerous artifacts that partially overlap. In particular, ensuring data consistency and data quality (e.g. accuracy, actuality) constitutes a challenging task in such an environment. Especially, use cases requiring data from different sources are difficult to support, due to missing mappings between interrelated artifacts. To realize these use cases, a partial integration of heterogeneous and distributed data models from different applications is required. Usually, these data models apply different concepts for supporting versioning, variability, and aggregation of E/E product data. Furthermore, semantic dependencies between artifacts from different application data models are undocumented. Another challenge in this context is the independent evolution of the data models maintained in different autonomous applications.

Usually, an IT landscape fostering E/E product development has evolved over years and is based on engineering expertise. To create a common integration model in such an environment, a methodology is needed that explicitly considers existing application data models and technologies. Furthermore, standards fostering the exchange and interoperability of product data (e.g. MSR[4], AUTOSAR[5] [AUT11b, AUT11a], ISO 10303) must be supported by the methodology as well.

This paper elaborates problems that occur when partially integrating heterogeneous applications and their data models in the context of E/E-PDM. Based on this, we identify fundamental requirements an IT support for E/E data management must meet in such a setting.

The remainder of this paper is organized as follows: Section 2 discusses problems and requirements for effective E/E-PDM. Section 3 discusses related work along the identified requirements. The paper concludes with a summary in Section 4.

[1] Restriction of Hazardous Substances Directive
[2] Waste Electrical and Electronic Equipment Directive
[3] Energy-using Products Directive
[4] Manufacturer Supplier Relationship
[5] AUTomotive Open System ARchitecture

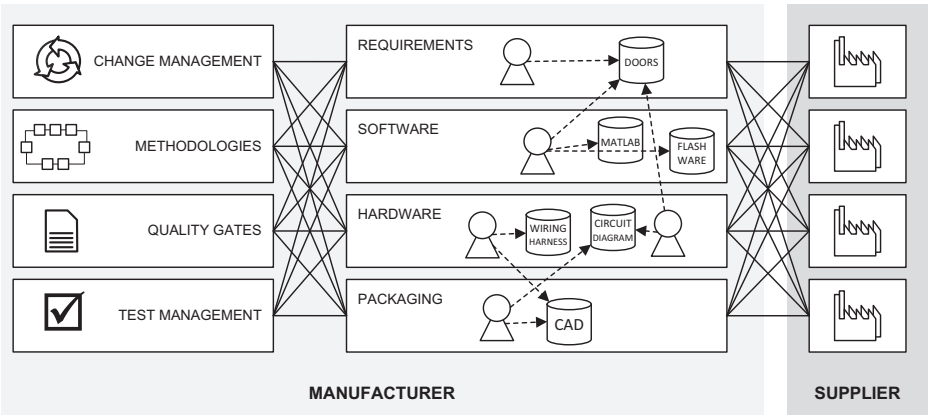


Figure 1: Complexity of E/E product data.

2 Findings, Challenges, Requirements

We analyzed applications involved in E/E development processes and considered several use cases enabling application-spanning consistency checks of E/E product data (e.g. plausibility of logical connections between electronic control units and networks). This section summarizes results of these analyses. It further identifies fundamental challenges emerging in the context of E/E development processes and their IT support. Finally, requirements for holistic and effective E/E-PDM are elicited.

2.1 Challenge 1: Common ontology for data integration

Findings. A main characteristic of E/E development processes is imposed by manufacturer-supplier relationships: Systems and components are developed in close collaboration with suppliers. Systems and sub-systems describe parts of a car characterized by the functions provided. Usually, systems aggregate features perceived by the customer (e.g. heat, ventilation, or air conditioning). Furthermore, they are technically realized based on interacting components (i.e., ECUs, actuators, and sensors). Due to the networked enterprises in a supply chain, for different development stages, deadlines become necessary to guarantee product delivery times. To cope with such a scenario, engineers execute development tasks concurrently (simultaneous engineering) using different applications (cf. Fig. 2). In addition, these applications provide different concepts for enabling versioning, variability, and aggregation of product data. Typically, engineers use numerous applications, hosted and maintained by different departments, to create and maintain E/E product data (e.g. applications for software function modeling, requirements engineering, and computer-aided design). These applications were designed and developed

independently from each other, and their data models reflect long-term experiences of engineers. Furthermore, there are important use cases not covered by the data models of these applications at all. For example, a car usually has ECUs to control functions of installed seats (e.g. position, heat and ventilation). If these ECUs do not differ in their functions, they are, represented by a single model object. To enable these use cases, a number of workarounds are applied, which constitute tacit knowledge of the engineers and are not documented at all. Other use cases, which utilize artifacts from different heterogeneous data sources (e.g. checking plausibility of references between ECU pins and logical signals) are performed manually by responsible actors. The latter collect and aggregate development artifacts from different departments. In particular, this constitutes a cumbersome and time-consuming task during which some of the relevant artifacts may have to be replaced by new versions.

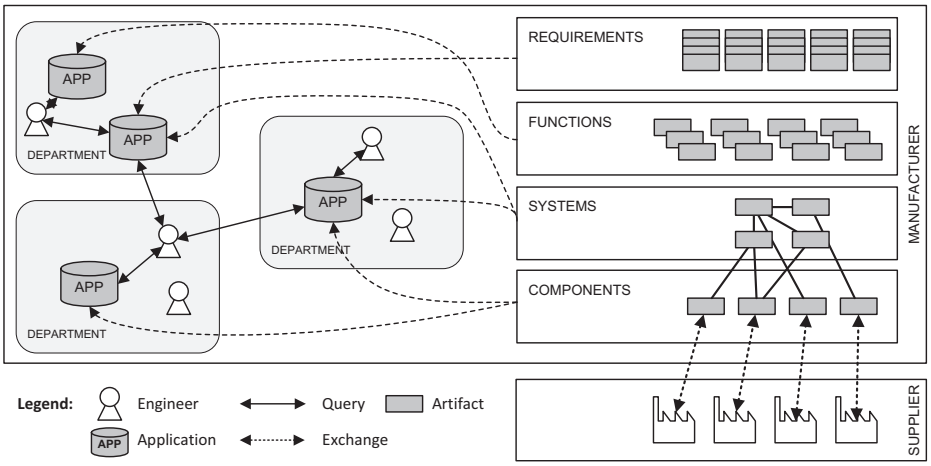


Figure 2: Logical structure of E/E product data.

Challenges. Usually, development tasks are separated into system- and component-specific parts. Both terms are ambiguously defined and used in application data models. For example, while a component in a wiring diagram refers to a variant of an E/E component at a specific geometric position, in the context of requirements documentation, the same term describes all variants of an E/E product in general.

Requirements. For use cases requiring read access to E/E data from different application, a common integration ontology is needed (*Requirement 1*). Such an ontology constitutes the core of any application involved in E/E development processes (cf. Fig. 3). To integrate E/E product data in a consistent and transparent manner, bidirectional mappings between related concepts are needed (*Requirement 2*). As an advantage of such mappings, semantic inconsistencies between application data models can be eliminated. In particular, through mappings changes are made transparent and reproducible. Moreover, integration of development artifacts can be accomplished semi-automatically and, thus, contribute to save

time and reduce errors (as introduced in the context of manual integration). To prevent ambiguities, any common integration ontology must allow modeling artifacts semantics. In addition, application-specific concepts for versioning, variability, and aggregation of E/E product data, as well as documentation methodologies must be considered in the context of a common integration ontology (*Requirement 3*).

2.2 Challenge 2: Consistency of E/E product data

Findings. E/E product data are created, maintained, and used in all stages of the automotive lifecycle (development, production, after sales). Usually, different stakeholders (e.g. E/E architects, system and component responsible, engineers) participate in these lifecycle phases using different applications [MHHR06]. In turn, this results in redundant artifacts and models (e.g. wiring harness, logical connection). In this context, many of the artifacts created in a particular lifecycle phase are required in subsequent development phases. We denote such interdependent applications as tool chains. Generally, development artifacts are reused along these chains to reduce development time. For example, if a new car model series is developed, artifacts of the previous model series serve as basis for product improvements and new ideas. To coordinate development steps, E/E systems and components are developed using the Vee Model [FMC05], which defines necessary steps and responsibilities of all involved parties for the distributed development and testing of artifacts. To control and test required functionality of an entire system, integration tests for the used components become necessary. Hence, quality gates constitute integral parts of any development process [Coo90]. In particular, they describe fixed development stages with predefined quality requirements to be fulfilled at the specified point in time. Generally, development processes involve different stakeholders, departments, and applications. Although global process overviews exist to some degree (e.g. Fig. 3), these have not realized based on workflow technology [RW12]. Changes of an E/E product are common and may impact other E/E products. For example, if the software interface of an ECU changes, logically connected ECUs must be identified and analyzed. For this purpose, global change management processes exist that coordinate and document change requests.

Challenges. Although development is performed simultaneously, artifacts from heterogeneous data sources must be continuously integrated at specific points during development (e.g. design review or prototype analysis). In this context, complex transformations between heterogeneous data models must be defined and maintained. In particular, explicit mappings between artifacts from different data models become necessary. Currently, these mappings are defined manually. Hence, communication between different stakeholders and development departments become necessary, which is error prone and time consuming. Overall, consistency of related artifacts from heterogeneous, distributed data models is a costly, manual task. Another problem is the lack of any technical processes implementation in E/E development, turning the monitoring of E/E product data changes a challenging and costly task; i.e., requests by different stakeholders and departments must be handled manually.

Requirements. The tight integration of OEM and suppliers requires data of high qual-

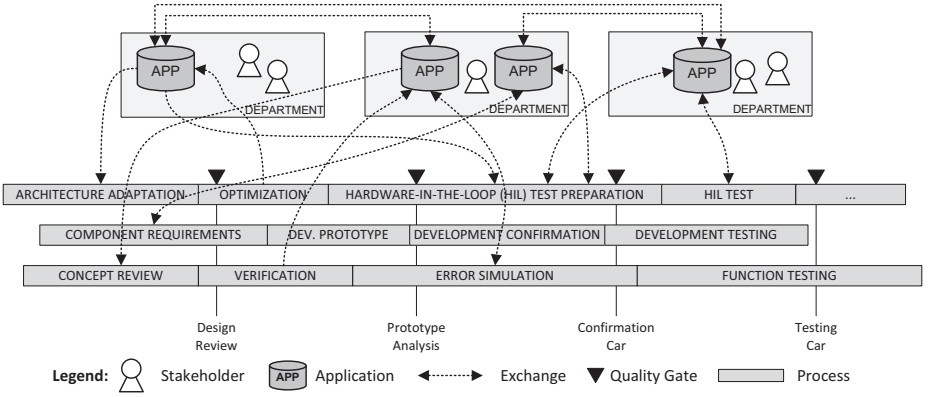


Figure 3: Overview on E/E development processes.

ity as well as standardized processes. For example, during the development of an ECU, functional models and communication information (software ports, frames, and signals) are concurrently created and maintained in different applications. Further, ECU development processes are separated into manufacturer- and vendor-specific parts; e.g vendors realize software for ECUs based on interfaces provided by the manufacturer. Hence, the consistency of artifacts must be guaranteed. Note that any error of a signal, frame, or bit might lead to a malfunctioned product. Hence, consistency management allowing for the identification of inconsistent E/E product data is crucial (*Requirement 4*).

2.3 Challenge 3: Data Model Changes

Findings. In general, changes of application data models are performed autonomously based on department-specific schedules. Although these schedules are distributed across the different E/E development departments, the coordination of data model changes remains a challenging task.

Challenges. Changes of application data models might affect other application data models which are difficult to identify due to undocumented interdependencies between artifacts from different data models. If a change is realized without notifying affected applications, the operation of existing tool chains might be impaired. Hence, before a change of an application data model can be realized, additional data models of other applications must be analysed.

Requirements. To handle data model changes in a more systematic manner, the dependencies between artifacts from different data models must be explicitly modeled and maintained (*Requirement 5*). Usually, data models comprise numerous artifacts. Hence, the manual comparison of all artifact combinations across different data models would be too costly. To reduce this complexity, matching algorithms based on well defined criteria

(e.g. name, data type), are needed. Additionally, data model interdependencies should be analyzed to evaluate the impact of data model changes in advance of their execution (*Requirement 6*). To enable flexible change management, transformations between application data models should be derivable from the dependencies maintained (*Requirement 7*). Hence, adaptations to data model changes can be semi-automated, which contributes to reduce development costs as well as errors.

2.4 Challenge 4: Methodology for documenting E/E product data

Findings. In the automotive domain, AUTOSAR is the de facto standard for developing embedded systems. In this context, the definition of standardized protocols for software development allows exchanging software running on ECUs from different vendors. In addition to the software architecture, methodologies for describing and defining reusable and scalable software components constitute an integral part of AUTOSAR. Besides standardizing ECU software, processes for developing safety related E/E products (e.g. airbag, electronic stability control) are an important component of contemporary E/E-PDM: First, *Failure Mode and Effects Analysis* (FMEA) [Sta03] is used for failure prevention and security management. In particular, FMEA focuses on reducing change costs of artifacts, i.e., on reducing late changes and hence reducing implementation costs. Second, ISO 26262 as emerging standard for functional safety in the automotive domain, needs to be considered. It defines a security lifecycle covering all aspects of development processes. To classify different levels of security requirements, so-called *Automotive Safety Integrity Levels* (ASILs) must be assigned to safety-relevant E/E products. Besides these standards, existing applications define methodologies for repeating tasks and product data documentation.

Challenges. Even though AUTOSAR's integrated methodologies support soft- and hardware, other aspects like wiring harness and connectors of E/E product data are not taken into account. Additionally, tacit knowledge makes the integration of data models a difficult task to accomplish, since semantic inconsistencies might remain unresolved. Finally, an explicit monitoring and control of existing documentation methodologies and guidelines are missing. As a consequence, ambiguities in E/E product data occur.

Requirements. A comprehensive methodology for documenting E/E product data is needed, taking existing standards and methodologies (e.g. AUTOSAR, ISO26262, Automotive SPICE [MHDZ12]) into account (*Requirement 8*). To create a common integration ontology allowing for the integration of existing application data models, a modeling methodology is needed. Note that applications creating and maintaining E/E product data have resulted from long-term experiences. Hence, a modeling methodology must support the creation of a common integration ontology from scratch (top-down) as well as from existing data models (bottom-up) (*Requirement 9*).

Table 1 summarizes the discussed requirements, and associates them along the common integration ontology in Figure 4.

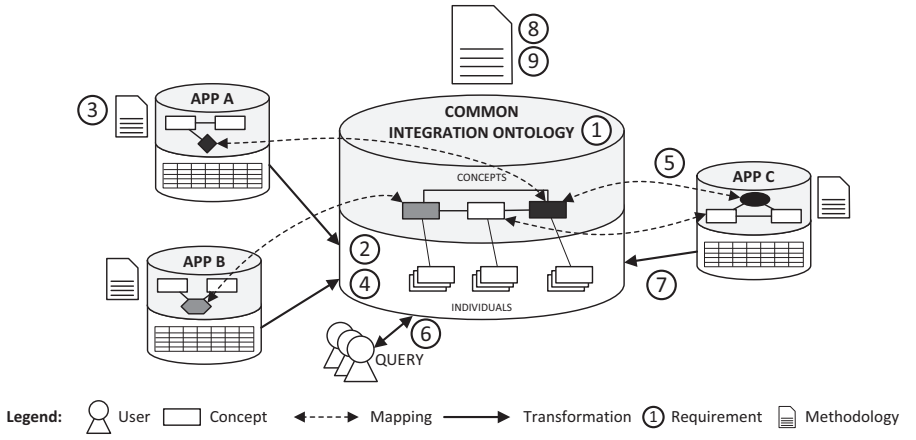


Figure 4: Common integration ontology.

3 Related Work

A common meta-model for tool integration based on EAST-ADL2 and HRC is presented in [Bau10]. This approach focuses on embedded systems engineering and derives concepts for a common meta-model from HRC and EAST-ADL2 meta-models. Although the authors take concepts, like component, part, and port into account, other ones (e.g. requirement, system) are missing. [CCWC09] uses an ontology for integrating distributed product knowledge and proposes a product knowledge service model realized through web services. In addition, a meta-knowledge schema for knowledge modeling is described. Finally, similarity measures for labels and relationships are introduced. The presented approach focus on the integration of heterogeneous, distributed knowledge. Integrating underlying application data schemes is not considered.

[BGN+04] is a model-based software (re-)engineering approach. It integrates tools at the meta-model level and proposes two design patterns for data integration. Further, consistency rules and integration constraints are provided. For this, simple graph grammar rules are defined restricting a meta-model to enable interoperability. Note that this allows for consistency checking as well. Finally, a triple graph grammar is used to model semantic relationships between different meta models. Other approaches using triple graph grammar are presented in [KS06]. ToolNet [FK03] focuses on tool integration and data consistency. For this purpose, consistency relations between reference objects are modeled manually. The approach focuses on requirements specifications, function models, and geometric product data. As a major drawback, changes of application data models are not considered. [NK04] compares the evolution of database schemes with the one of ontologies evolution. In particular, it highlights the differences of both research areas and gives insights into challenges for ontology evolution. These challenges for ontology evolution constitute a starting point for future research on change management of the

No.	Description
1	Common integration ontology.
2	Bidirectional mappings between artifacts.
3	Transparency of data model semantics and documentation methodologies.
4	Consistency management for E/E product data.
5	Maintenance of data model interdependencies.
6	Support of impact analysis.
7	Transformations between application data models must be derivable from the maintained dependencies.
8	Comprehensive documentation methodology, taking current standards into account.
9	Modeling methodology for the common integration ontology.

Table 1: Requirements for E/E product data management.

required common integration ontology. In summary, there exists no holistic approach covering the aforementioned challenges for E/E product data management in an integrated and comprehensive manner. Although many approaches focus on tool integration, change management of partly integrated application data models is not properly considered. In addition, methodologies for E/E product data management are not covered.

4 Summary and Outlook

This paper has identified challenges emerging in the context of E/E-PDM. Based on a comprehensive analysis of applications involved in E/E development processes as well as characteristic use cases for application-spanning consistency checks, we have elaborated requirements for effective E/E-PDM. To enable application-spanning uses cases (e.g. consistency), a common integration ontology is required allowing for the integration of relevant concepts from existing application data models. In this context, ensuring the consistency of artifacts is a challenging task. Another challenge concerns the evolution of application data models. Note that such changes of application data models are common and hence must be supported. A possible approach is to explicitly document data model interdependencies. The latter are crucial for automatically deviating data model transformations. Finally, a documentation methodology is needed, that takes existing applications data models, technologies as well as standards into account.

In future work we will provide detailed insights into our integration framework and methodology. Finally, we will focus on changes of application data models in context of E/E-PDM.

Acknowledgements. This work has been conducted within the PROCEED[6] project funded by Daimler.

[6] **PRO**active **C**onsistency for **EE** product **D**ata management

References

[AUT11a] AUTOSAR. AUTOSAR Methodology, Technical Report Version 1.2.2, Release 3.2, Rev 0001, 2011.

[AUT11b] AUTOSAR. Technical Overview, Technical Report Version 2.2.2, Release 3.2, Rev 0001, 2011.

[Bau10] A. Baumgart. A common meta-model for the interoperation of tools with heterogeneous data models. In *In 3rd Workshop on Model-Driven Tool and Process Integration (MDTPI 2010)*, Paris, France, June 2010.

[BGN+04] S. Burmester, H. Giese, J. Niere, M. Tichy, J.P. Wadsack, R. Wagner, L. Wendehals, and A. Zündorf. Tool integration at the meta-model level: the Fujaba approach. *Int J Softw Tools Technol Transfer*, 6:203–218, 2004.

[CCWC09] Y.M. Chen, Y.J. Chen, C.C. Wen, and H.C. Chu. Ontology-Based Knowledge Integration for Distributed Product Knowledge Service. In *Proc. World Congress on Engineering and Computer Science*, 2009.

[Coo90] R.G. Cooper. Stage-gate systems: a new tool for managing new products. *Business Horizons*, 33(3):44–54, 1990.

[FK03] R. Freude and A. Königs. Tool integration with consistency relations and their visualization. In *Proc. Workshop on Tool-Integration in System Development (TIS 2003)*, pages 6–10, 2003.

[FMC05] K. Forsberg, H. Mooz, and H. Cotterman. *Visualizing project management: Models and frameworks for mastering complex systems*. Wiley, 2005.

[fS11] International Organization for Standardization. ISO/FDIS ISO 26262-2:2011: Road vehicles – Functional Safety – Part 2: Management of functional safety, 2011.

[HBR08] A. Hallerbach, T. Bauer, and M. Reichert. Managing Process Variants in the Process Lifecycle. In *10th Int'l Conf. on Enterprise Information Systems (ICEIS'08)*, pages 154–161, June 2008.

[KS06] A. Königs and A. Schürr. Tool integration with triple graph grammars-a survey. *Electronic Notes in Theoretical Computer Science*, 148(1):113–150, 2006.

[MHDZ12] M. Mueller, K. Hoermann, L. Dittmann, and J. Zimmer. *Automotive SPICE in Practice: Surviving Implementation and Assessment*. Rocky Nook, 2012.

[MHHR06] D. Müller, J. Herbst, M. Hammori, and M. Reichert. IT Support for Release Management Processes in the Automotive Industry. In *4th Int'l Conf. on Business Process Management (BPM'06)*, number 4102 in LNCS, pages 368–377. Springer, September 2006.

[NK04] N.F. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and information systems*, 6(4):428–440, 2004.

[RW12] M. Reichert and B. Weber. *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Springer, Berlin-Heidelberg, 2012.

[Sta03] D.H. Stamatis. *Failure mode and effect analysis: FMEA from theory to execution*. ASQ Press, 2003.

Extending an Index-Benchmarking Framework with Non-Invasive Visualization Capability

David Broneske[1], Martin Schäler[1], Alexander Grebhahn[2]

[1] Department of Technical and Business Information Systems
Otto-von-Guericke University Magdeburg
david.broneske@st.ovgu.de, schaeler@iti.cs.uni-magdeburg.de

[2] Brandenburg University of Applied Sciences
grebhahn@fh-brandenburg.de

Abstract: Finding a suitable multi-dimensional index structure for a data-intensive system is not a trivial task. The QuEval framework supports users in finding the best index structure from a list of candidates. Nevertheless, if an index structure shows itself superior to other index structures most oft the times, but fails for one data set, we want to know the reason for this phenomenon. To support an understanding of deficits, a visualization of the partitioning scheme is helpful. Consequently, we propose a visualization component which interacts with QuEval without affecting the performance evaluation. Thus, we use a modern software-engineering approach based on AspectJ to support Digital Engineering of complex solutions.

1 Introduction

Modern forensic investigation makes increasing use of digitally stored evidences. For this purpose, on the crime scene latent fingerprint scans are executed [KFV11]. These fingerprints are stored in a database to find possible suspects. Querying a fingerprint database bears challenges, e.g. given workload [GG98], data dimensionality and used query types [GBS+12]. These challenges are addressed by multi-dimensional index structures. Numerous multi-dimensional index structures are introduced [GG98, Sam05], but the most suitable index structure depends on the current scenario.

To find the most suitable index structure for a given use case, the QuEval framework[1] can be used. QuEval offers an expendable list of index structures, customizable data sets and different query types, too. Working with QuEval, we observed performance issues of index structures with some data sets. Consequently, we have to figure out partitioning deficits under these circumstances. As a consequence, the need for a visual representation arises to examine the partitioning scheme and to derive deficits [GBS+12]. Furthermore, the sequence of operations (e.g. inserts and updates) influence the partitioning of index structures [NT01]. Thus, two different sequences performing the same operations lead to different structures. In summary, such a visualization component shall, for instance help engineers to decide whether identified performance problems are due to limitations of the index structure or may be due to erroneous data gathering procedures.

[1] http://wwwiti.cs.uni-magdeburg.de/iti_db/research/iJudge/index_en.php

The research question, we address is whether we can generalize the visualization of index structures and whether we can automatically extract necessary information. To implement such a visualization component, we have to face two tasks:

(a) a suitable visualization of indexed regions for multi-dimensional data and
(b) the information extraction from QuEval and integration into the framework without remodeling of already implemented index structures.

To summarize, we need to integrate a non-invasive visualization that does not affect non-functional properties of the index structures, such as response time and memory consumption. Consequently, we need a modern software-engineering approach that allows seamless integration and unintegration of the additional functionality.

The paper is structured as follows: In Section 2, we show relevant properties of index structures for our visualization followed by our visualization and software requirements. We show their implementation in Section 4 and 5 and evaluate the implementation of our requirements in Section 6. We finish with related work, conclusion and future work.

2 Impact of Index Structure Categories on the Visualization

In literature, many different index structures are proposed. To give an overview of existing index structures, we categorize them either as tree techniques, optimized sequential search, dimensionality reduction or hashing methods [GG98]. The following paragraphs will give a brief overview of these categories, which will help us to figure out specific requirements on the visualization. For more detailed information, we refer to the given references.

Tree Techniques. Tree techniques partition the space hierarchically to improve the query performance. They partition the space so that one superordinate regions encloses several subordinate regions. This builds a hierarchical tree structure. With this, the complexity should decrease from $\mathcal{O}(n)$ to $\mathcal{O}(log(n))$ for point search. An important multi-dimensional tree technique is the R-Tree [Gut84] as well as its derivates R^+-Tree [SRF87] and R^*-Tree [BKSS90]. In Figure 1, we visualize the SS-Tree [WJ96] using multi-dimensional spheres and the SR-Tree [KS97] whose regions are the intersection between a multi-dimensional sphere and rectangle. However, tree techniques suffer from the curse of dimensionality, which says that at a certain dimensionality (for tree techniques ca. 26), the nearest neighbor query performance of an index structure will be worse than a sequential search [WSB98]. Thus, a visualization of the query execution would be helpful to show these deficits. When visualizing the partitioning scheme, it is important to have a good representation for the hierarchy. Since regions on the same level may overlap, the belonging of subordinate to superordinate have to be encoded in the visualization strategy.

Optimized Sequential Search. Since tree-based index structures are affected by the curse of dimensionality, methods using an optimized sequential search are introduced which compare every data point with the query. The optimization is often an approximation of the actual data points to reduce comparison costs. For example, the VA-File [WB97] splits the whole space into cells

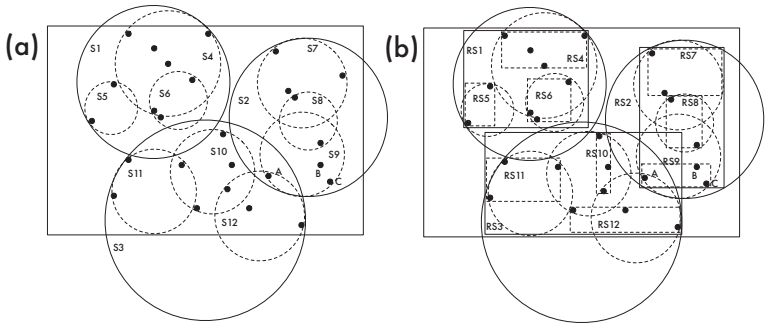


Figure 1: (a) SS-Tree, (b) SR-Tree

and assigns a unique bit string for each. For each data point the bit string of its cell will be stored. As a result, many bit strings of points fit into main memory and disk accesses are reduced. The partitioning should result in almost equally filled regions, nevertheless the data distribution may interfere with this goal. Consequently, visualizing the partitioning is helpful.

Dimensionality Reduction. Another option to avoid the curse of dimensionality is to decrease the dimensionality of the space. This can be done for example using space-filling curves. A space-filling curve is a mathematical function, which is constructed iteratively assigning a one dimensional value to each indexed region. Typical space-filling curves are the Z-Curve [OM84], used in the UB-Tree [Bay97], and the Hilbert-Curve [FR89]. Considering nearest neighbor queries, it is helpful to see neighboring regions in the space-filling curve and whether all of them are evaluated.

Hashing Methods. As a hashing method for multi-dimensional data, Locality Sensitive Hashing (LSH) [IM98] is most often used, because they support nearest neighbor queries. LSH uses locality sensitive functions to set up several hash tables with the constraint that all points in one bucket should, with a high probability, be locally nearer to each other than to any point in another bucket. With this constraint, neighborship relations are preserved. Promising LSH methods are p-stable LSH [DIIM04] whose partitioning looks similar to stairs and the permutation approach [CGFN08]. Since hashing methods use several hash tables, a visualization of several partitionings is necessary.

3 Requirements for a Visualization Component

For our visualization component, we divide the requirements into (a) visual and (b) software requirements. The visualization requirements define properties that help to show the functionality of an index structure, whereas software requirements demand that no runtime or memory consumption overhead occurs from additional visualization implementation.

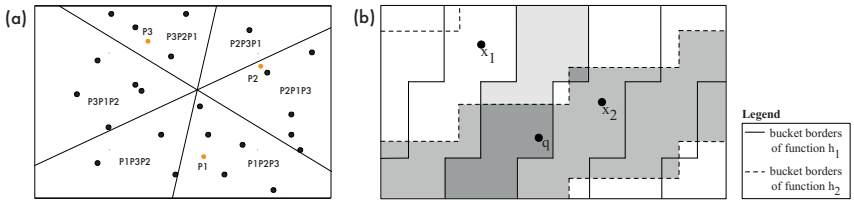


Figure 2: (a) permutation Approach of LSH, (b) p-stable LSH

3.1 Visual Requirements

The visual requirements include visualizing the (a.1) partitioning scheme, (a.2) hierarchies, (a.3) queries and it should be able to (a.4) support different dimensions of data.

(a.1) Giving the user an understandable picture of an index structure, we want to visualize the partitioning schema step by step when constructing the index structure. With this, users should be able to figure out the functionality of the index structure and understand important algorithms, for example different split algorithms of the R-Tree. Even a comparison of the same index structure with different algorithms can be done with a visual representation. Another advantage is that optimizations can be estimated, when phases are seen where the partitioning results in overfull regions.

(a.2) Another challenge is to represent hierarchies. Visualizing a hierarchical index structure without visual support for different hierarchies, it may be hard to differentiate between different hierarchy levels.

(a.3) Additionally, we want to highlight regions that are accessed when a query is performed. With this visualization, users can identify regions that are frequently evaluated, because they are too large or overfull. Considering hierarchical index structures, we can additionally identify regions that are too near or even overlapping so that multiple paths have to be taken while evaluating a query.

(a.4) An important role is to support different dimensions. Since multi-dimensional data implies a dimensionality of up to 100 [VCPF08], we also have to be able to visualize all of these dimensions and dependencies between them.

3.2 Software Requirements

Considering the implementation, we define the following requirements: (b.1) no adjustments to index interface given and used by QuEval, (b2.) no performance deficits while evaluation, (b.3) a common interface for the extracted visual information.

(b.1) Our visualization component will be integrated into QuEval. QuEval is a Java framework for finding the most suitable index structure for a given use case. To have a comprehensive collection of index structures, an extension of the collection can be done very easily. Users

do only have to implement a few functions to conform to the simple interface. For more information visit our website[2]. When integrating a visualization component, the interface should not be change to assure an easy implementation of index structures.

(b.2) Since QuEval is a tool for evaluating the query performance of an index structure, our visualization should not affect the performance of the index structure. Consequently, we must not produce a computational overhead for our visualization when an evaluation is in progress. Thus, necessary code for extracting visual information should only be present, if we want to visualize an index structure and no performance benchmark between several index structures is running.

(b.3) Another requirement refers to the extracted information, which will be used to visualize the partitioning scheme. To have a suitable representation of the extracted information, we have to find a common interface to describe index structures.

We visualize a summary of our requirements in Table 1. In the following both sections, we will present first solutions for the given problems and primary concepts of our implementation.

	Visual requirement		**Software requirement**
(a.1)	visualizing partitioning scheme	(b.1)	no adjustment to index interface
(a.2)	support hierarchies	(b.2)	no performance deficits
(a.3)	visualizing query evaluation	(b.3)	common interface
(a.4)	visualizing different data dimensions		

Table 1: Visual and software requirements

4 Visual Representation

In this section, we discuss implementation of the visual requirements, we defined before. We visualize a screenshot of our tool visualizing the R-Tree in Figure 3.

(a.4). To support a multi-dimensional representation of the data and partitioning scheme, we use a scatterplot matrix. A scatterplot is the visual representation of a Cartesian space where the data points are arranged. Although 3D-Scatterplots can be extended using color, shape, and size [EDF08], a single scatterplot does not suffice our requirements for visualizing multi-dimensional data. To solve this problem, we use a scatterplot matrix having the dimensions as rows and columns. Each entry represents an own 2D-scatterplot visualizing the data distribution in the corresponding dimensions. We only visualize the upper triangular matrix (cf. 3), because the lower scatterplots are mirrored duplicates. Nevertheless, with increasing dimensionality, we face the problem of an overcrowded display. Nevertheless, we cannot use an approximation of the data, because the partitioning schema relies on a multi-dimensional vector space. Changing the visualization paradigm (a collection can be found here [Kei02]) would destroy the relation between data and the corresponding partitioning scheme of the index structure.

[2] http://wwwiti.cs.uni-magdeburg.de/iti_db/research/iJudge/index_en.php

(a.1), (a.3). To visualize the construction and query execution, we identify *two-dimensional* geometrical shapes the index structures are based on. Most of the index structures use simple geometrical shapes (here: line, rectangle, sphere):

- Space-filling curves and permutation approach of LSH use lines.
- The R-Tree and its derivates as well as the VA-File are based on rectangles.
- Spheres are used in the SS-Tree.

These geometrical shapes can easily be drawn, if we extract the necessary information. For instance, we only need the end points of the lines to draw it or the end points of the diagonal for a rectangle. To draw a sphere, the center point and the radius have to be known.

Nevertheless, there are some exceptions that are not based on these simple shapes. An index structure that is using multiple geometrical shapes is the SR-Tree. One region is the intersection of a minimum bounding rectangle and a minimum bounding sphere. The visualization of such a complex region is hard to implement without further computations (cf. Figure 1.b), so we decided to draw both shapes. Complex regions are also created when using p-stable LSH, as the region of the hash buckets are similar to stairs and one region cannot be constructed using one simple geometries (cf. Figure 2.(b)).

(a.2). When visualizing hierarchical index structures, we have to support the user in distinguishing between regions of different levels. For this, we provide the individual tree structure, where the user can filter the visualized regions. By hovering over or selecting a node in the tree in the right half of the tool, only the node and its subtree is visualized in the scatterplot matrix.

5 Integration into QuEval

This section addresses our implementation of software requirements, we defined in Section 3.

(b.2). The necessary code for our visualization component is only integrated, when a visualization is performed. An overhead should not be present, when a performance evaluation is done. For this, we use AspectJ to integrate our code into QuEval.

AspectJ is an extension to Java which introduces a set of language constructs to permit aspect-oriented programming in Java [KHH+01]. Aspect-oriented programming aims at implementing crosscutting problems in one programming unit, namely an *aspect*. The code of an aspect is weaved into code, when the aspect is activated. Otherwise, the code will not be taken into account and no computational or storage overhead occurs. The points where aspects add their code will be defined by pointcuts which can extend methods, fields, or classes. A detailed overview of AspectJ can be found in [KHH+01].

(b.1). With AspectJ, changes in the classes are weaved into code when the corresponding module is activated. Thus, the interface stays as simple as it was. Nevertheless, users have to implement suitable AspectJ modules for their index structures, if they want to use the visualization component.

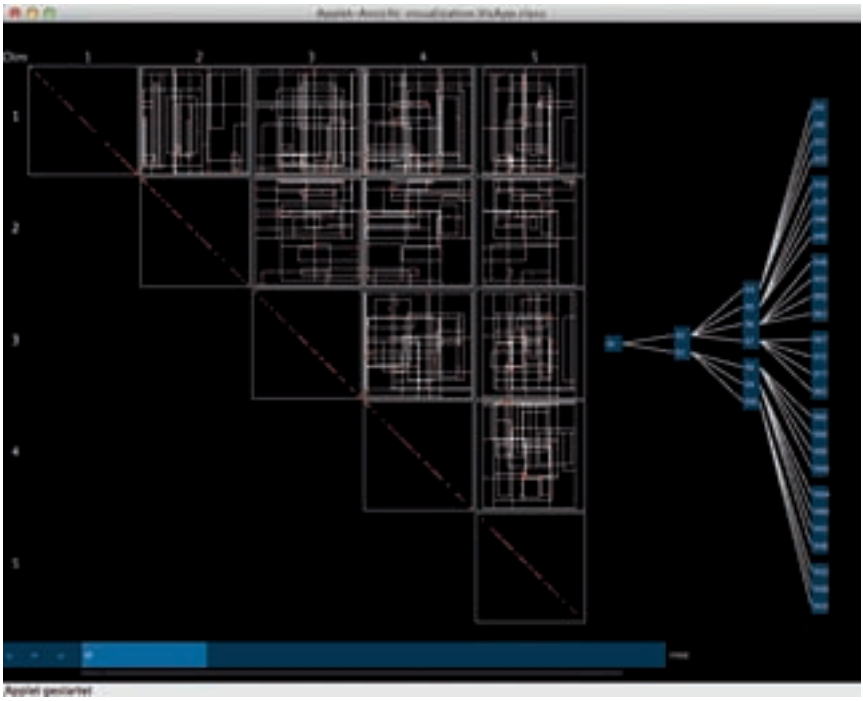


Figure 3: Screenshot of our visualization tool

(b.3). The AspectJ modules should hook in at points, where the partitioning scheme changes or a query is executed and extracts the necessary information. Per stage (state after an insertion or deletion) of the index structure, an array of regions is constructed. We describe a region as a pair of a geometrical object and the parent-ID which encodes the hierarchy. If the index structure is non-hierarchical, the parent-IDs are `NULL`. Consequently, we describe a region with the following interface:

$$(GeoObj, parentID)$$

A geometrical object has an ID, a type, and a location. The simple geometrical shapes (cf. Section 4) can be encoded by two arrays, because lines and rectangles can be described by two two-dimensional points. A sphere consists of one two-dimensional point and the radius of the sphere, so the first entry of the second array denotes the radius while the rest is zero. Consequently, a geometrical object can be encoded as:

$$(ID, Type, MinPoint, MaxPoint)$$

The extracted information can be used to visualize the index structure. To have a reconstructable image of the index structure, the stages are stored and visualized after a full run of evaluation done by QuEval. Then, users can stepwise switch through the stages and see what is happening.

6 Evaluation of Requirements

In this section, we summarize the implementation of our requirements, which is represented in Table 2.

Our visualization component supports hierarchies and queries can be visualized on the partitioning scheme. Nevertheless, there are index structures that are hard to visualize at the current stage, for example p-stable LSH. Furthermore, we visualize multi-dimensional data using a scatterplot matrix, but an increasing number of dimension the user loses the overview.

Considering the software requirements, AspectJ helps us to extend QuEval without performance impact or extension of the index interface. Furthermore, we have presented an interface describing our simple geometrical objects.

	Visual requirement			**Software requirement**	
(a.1)	visualizing partitioning scheme	≈	(b.1)	no adjustment to index interface	✓
(a.2)	support hierarchies	✓	(b.2)	no performance deficits	✓
(a.3)	visualizing query evaluation	✓	(b.3)	common interface	✓
(a.4)	visualizing different data dimensions	≈			

Legend: ≈ - partially implemented; ✓ - completely implemented

Table 2: Implementation of visual and software requirements

We have shown, that it is possible to visualize many index structures. For the presented index structures, it is possible to have a generalized representation which can be used for a visualization. Nevertheless, the programmer of the index structure has to provide the necessary AspectJ modules for each index structure that has to be visualized.

7 Related Work

Related work has already been done by Keim and Kriegel [KK94]. With VisDB, they introduce a system for an exploration of databases. They present their own visualization paradigm using grouping and transformation of the data into a two-dimensional screen to have a compressed view on the data. Furthermore, they visualize queries on the data which can be redefined by users to get a more meaningful query result. In fact, the aim of VisDB is related to our topic. Nevertheless, we cannot transform the data, because the relation between data and the corresponding partitioning scheme would be destroyed.

Another important visualization framework is the scalable framework [KLS00]. Since there is a limited number of objects that can be visualized, the authors propose several techniques, such as Self-Organizing Maps or Magic Eye View to represent high amounts of data points. Such techniques may be helpful to give an overview of scatterplots in the matrix, because the two-dimensional space is displayed in a small pane.

8 Conclusion and Future Work

Visualizing index structures is a promising task to show their functionality and identify disadvantageous partitioning. We have shown, that there are many opportunities to support user's perception considering the partitioning scheme, hierarchies and query execution. The partitioning scheme can be visualized using simple geometric shapes. To extract shapes that have to be visualized, we use AspectJ to weave the necessary code into the index structure. We presented an encoding for the information that will be used in our visualization component.

Further challenges refer to the dimensionality. At the moment, we use scatterplot matrices to visualize the multi-dimensional space. However, with increasing dimensionality, the matrix gets huge and a single scatterplots is barely visualizable. Furthermore, the user is overflooded by numerous scatterplots which makes it hard to make important findings. Another task would be to help the user in comparing the same index structure with different partitioning algorithms or inserting the same data in different orders. Displaying two evaluations side by side would support an understanding of differences in algorithms or insertion orders.

Acknowlegements

The work in this paper has been funded in part by the German Federal Ministry of Education and Science (BMBF) through the Research Program under Contract No. FKZ: 13N10817 and FKZ: 13N10816.

References

[Bay97] Rudolf Bayer. The universal B-Tree for multidimensional indexing: General concepts. In *Proc. Int'l Conf. on Worldwide Computing and its Application (WWCA)*, pages 198–209. Springer-Verlag, 1997.

[BKSS90] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R*-Tree: An efficient and robust access method for points and rectangles. In *Proc. Int'l Conf. on Management of Data (SIGMOD)*, pages 322–331. ACM, 1990.

[CGFN08] Edgar Chavez Gonzalez, Karina Figueroa, and Gonzalo Navarro. Effective proximity retrieval by ordering permutations. *IEEE Trans. Patt. Anal. and Machine Intell. (TPAMI)*, 30(9):1647–1658, 2008.

[DIIM04] Mayur Datar, Piotr Indyk, Nicole Immorlica, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. Symp. on Comput. Geometry (SCG)*, number 3, pages 253–262. ACM, 2004.

[EDF08] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 14:1141–1148, 2008.

[FR89] Christos Faloutsos and Shari Roseman. Fractals for secondary key retrieval. In *Proc. Symp. Principles of Database Systems (PODS)*, pages 247–252. ACM, 1989.

[GBS+12] Alexander Grebhahn, David Broneske, Martin Schäler, Reimar Schröter, Veit Köppen, and

Gunter Saake. Challenges in finding an appropriate multi-dimensional index structure with respect to specific use cases. In *Grundlagen von Datenbanken Workshop*, pages 77–82, 2012.

[GG98] Volker Gaede and Oliver Günther. Multidimensional access methods. *ACM Comput. Surv.*, 30(2):170–231, 1998.

[Gut84] Antonin Guttman. R-Trees: A dynamic index structure for spatial searching. In *Proc. Int'l. Conf. on Management of Data (SIGMOD)*, pages 47–57. ACM, 1984.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. Symp. on Theory of Computing (STOC)*. ACM, 1998.

[Kei02] Daniel A. Keim. Information visualization and visual data mining. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 8(1):1–8, January 2002.

[KFV11] Tobias Kiertscher, Robert Fischer, and Claus Vielhauer. Latent fingerprint detection using a spectral texture feature. In *Proc. ACM workshop on Multimedia and Security (MMSec)*, pages 27–32. ACM, 2011.

[KHH+01] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and William G. Griswold. An overview of AspectJ. In *Proc. Europ. Conf. on Object-Oriented Programming (ECOOP)*, pages 327–353. Springer-Verlag, 2001.

[KK94] Daniel A. Keim and Hans-Peter Kriegel. VisDB: Database exploration using multidimensional visualization. *IEEE Comput. Graph. Appl.*, 14(5):40–49, September 1994.

[KLS00] Matthias Kreuseler, Norma Lopez, and Heidrun Schumann. A scalable framework for information visualization. In *Proc. Symp. on Information Vizualization (INFOVIS)*, pages 27–37, 2000.

[KS97] Norio Katayama and Shin'ichi Satoh. The SR-Tree: An index structure for high-dimensional nearest neighbor queries. In *Proc. Int'l. Conf. on Management of Data (SIGMOD)*, pages 369–380. ACM, 1997.

[NT01] Moni Naor and Vanessa Teague. Anti-persistence: History independent data structures. In *Proc. ACM Symp. on Theory of computing (STOC)*, pages 492–501. ACM Press, 2001.

[OM84] Jack A. Orenstein and Tim H. Merrett. A class of data structures for associative searching. In *Proc. Symp. on Principles of Database Systems (PODS)*, pages 181–190. ACM, 1984.

[Sam05] Hanan Samet. *Foundations of multidimensional and metric data structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[SRF87] Timos K. Sellis, Nick Roussopoulos, and Christos Faloutsos. The R+-Tree: A dynamic index for multi-dimensional objects. In *Proc. Int'l Conf. on Very Large Data Bases (VLDB)*, pages 507–518. Morgan Kaufmann Publishers Inc., 1987.

[VCPF08] Eduardo Valle, Matthieu Cord, and Sylvie Philipp-Foliguet. High-dimensional descriptor indexing for large multimedia databases. In *Proc. Int'l Conf. on Information and Knowledge Management (CIKM)*, pages 739–748. ACM, 2008.

[WB97] Roger Weber and Stephen Blott. An approximation-based data structure for similarity search. Technical report, ETH Zürich, 1997.

[WJ96] David A. White and Ramesh Jain. Similarity indexing with the SS-Tree. In *Proc. Int'l Conf. on Data Engineering (ICDE)*, pages 516–523. IEEE Computer Society, 1996.

[WSB98] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proc. Int'l. Conf. on Very Large Data Bases (VLDB)*, pages 194–205. Morgan Kaufmann Publishers Inc., 1998.

Concept for a web based Support of the Product Development Process

M.Sc. Marc Oellrich, Prof. Dr.-Ing. Frank Mantwill

Machine Parts and Computer Aided Development
Helmut-Schmidt-University
Holstenhofweg 85
22043 Hamburg
marc.oellrich@hsu-hh.de
frank.mantwill@hsu-hh.de

Abstract: During the last years there have been some developments in the internet which might support the product development process. Some ideas at the beginning of the millennium have shown that web based systems can raise the efficiency, but the possibilities are nowadays much higher. While at that time representations have been only in a static state, they can now be handled much more user-friendly and get accepted like shown with Wikipedia or Facebook. Interesting further opportunities are given by Open Innovation, where problems are solved by a big amount of online users.

This concept will show principal components of an integrated web based system, which supports the development methodological approach, reduces the workload to collect and enter redundant data, allows collaborative and partially asynchronous cooperation and will contribute to determine and map the knowledge and experience of the employees, what should lead to a higher ability to compete and gives a clear competitive advantage.

1 Introduction

The competitive pressure on enterprises of almost every sector is on a high level and will probably grow further, driven by the Emerging Markets, while the customers on the other hand are requesting a more and more specific diversity of products. The enterprises also have to face problems of losing know-how, caused i.e. by the demographic change. The resulting requirements for the Development Process with abridged time-to-market cycles, improved meeting of the customer needs and a less amount of know-how carriers are enormous.

Some enterprises already got to know the advantages of a development methodological approach. Which includes shorter development times, better design solutions by using established best-practice ones and comparison of different solution variants based on lots of ideas. Further advantages are given by less material- and manufacturing costs,

reduced needed documentation and most important at all, a higher customer satisfaction. But still most of the enterprises have a non-methodological approach or work even intuitively [Eh03] [PBF05].

First IT-supporting tools for the Development Process have been developed in the last century. While those tools often supported project management or process management, they often were not flexible enough for companies to use them efficiently or they were not supporting any methods of the development methodology. At the beginning of the millennium some projects have shown the positive effects of web supported development methodology like reduced development costs by 20% and reduced development time by 25% (projects in plant engineering, production of domestic appliances and railway vehicle manufacturing) [ESV01]. But this support assisted the user/ team member only by giving textual instructions for each process step during the development process and maybe offered standardised documents to download. Other IT-supporting tools usually addressed the product data management or CAD-systems what are different themes and should not be considered in this paper.

Nowadays, there are possibilities to compensate the mentioned disadvantages and to support the development methodology much more, by using easy extensible web technologies. Especially young employees are already familiar with those technologies and could assist older employees to overcome possible inhibitions.

The following sections will sketch an integrated web based system, starting with the basis functionalities of the total system in section 2, the basis project management functionalities in section 3, the special functionalities in section 4 and ends with the conclusion in section 5.

2 Basis Functionalities

Most of the following seven functionalities are common standard or are part of additional software in the companies which can normally be accessed via interfaces (APIs). But for showing the possible benefit, it is easier to think of an all integrated system.

For having a personal system which allows secure file access and know-how access as well as to communicate with other members and to be informed about all relevant activities the basis functionalities should include

- A Login,
  so the system can generate dynamic user and role specific views.
- User profiles,
  so other members can have an idea with whom they are working with when they see a picture of the other user.
- A Document-Management-System,
  so files can be handled by checking them out and in.
- A Wiki,

so information can be entered and accessed central.

- A Blog-system,
  which works as a collaboration-platform like a forum and can be used as well as a comment-system or for user-to-user communication as a message-system.
- An Activity stream,
  so every user gets clearly informed about what happened globally (inside of the company/ enterprise) and in the projects he is part of. It allows also the direct entry by using the links in the activity stream.
- A Versioning-system,
  so every entered data or uploaded file can be restored.

The system should be based on a server with a database, where the information are stored.

3 Basis Project Management Functionalities

After successful login the user sees the activity stream, a list of the activities or at least the count of activities that happened since his last login. Those activities are ordered and grouped by global activities, those who belong to everything out of the projects and by project specific activities, so that the user is able to get an idea how much has happened in each project (s. Figure 1).

Furthermore the user can access each project by the links of the activity stream or he can create a new project, provided he has the necessary rights.

Figure 1: Exemplary project Activity stream

Each new project can be chosen from a predefined type, like e.g. the four development phases of the VDI 2221[1] [VDI93] or the ten phases of the Value Management[2] [VDI10].

[1] Clarify the tasks; plan; design; work out

[2] Prepare project; define project; plan project organisation; collect comprehensive data on the object; analyse functions and costs, formulate detailed targets; collect and develop solution ideas; evaluate solution ideas;

It is also possible to enter new or adapt existing project types and phases. This allows the company to build master types of their own usual development process(es). These phases will lead through the projects. Components (tools) which assist the methodological approach can be assigned to specific phases, so that it is easier to give the user a clear view of expedient tools. Using master types helps also to live a continuous improvement process.

Checklists can be developed and assigned to a total master project as also for each phase to support the project(phase). These checklists as well are derived from adoptable master copies to live the continuous improvement process.

This basis functionalities should enable a project team to do their usual work only in a different system. So the conventional way of working is still possible. The positive effects of this system can be achieved, when the components described in the next section are used. In that case, the tools can start to work together and generate an added value.

4 Special Functionalities

4.1 Data Access

The common systems already have a rights and role management. But those systems can normally only check if someone has access rights, denying restrictions cannot be handled. In [Ha12] the author points out that enterprises in the industry are interested in using Wikis, but she found out that the users are afraid of using Wikis to share confidential data, if the Wiki can be accessed by everyone. The usual solution of the enterprises ends in Wikis which can be accessed only from some departments or they renounce to use Wikis.

Another idea would be to use a rights and role management based on the mathematical theory of sets. This would make it possible to give a group of people the right to access the data and also to give other groups or persons the non-access right what reduces the set (s. Figure 2). Especially for large enterprises with lots of different user groups and therewith many roles this should simplify the administrative work, because it will not be necessary to define for each reduced set an own role.

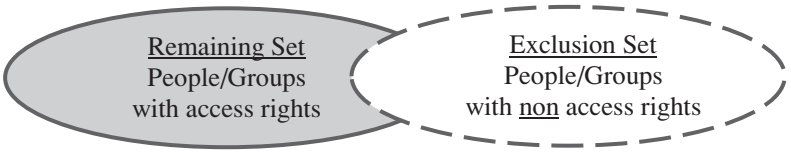


Figure 2: Rights and role management based on the theory of sets

develop holistic proposals, select solutions; present proposals, bring about decisions; implement decisions, realise proposals

A possible implementation would be to allow each tool to register their necessary rights in the system. Therefor it would be helpful to register the ‘dominant-right’ and also the ‘anti-right’ and map both with each other. For each access request the system can then check which rights exist and if there is a dominant one.

4.2 Communication

For collaboration on the platform it is necessary to be able to communicate. While audio and video support is difficult to implement and also needs a lots of bandwidth, the implementation of text based communication like messages, chats or blogs is easy. As shown in section 2, the communication should take place with a blog-system. This blog-component is planned to be configurable so that it can be used for different tasks on the

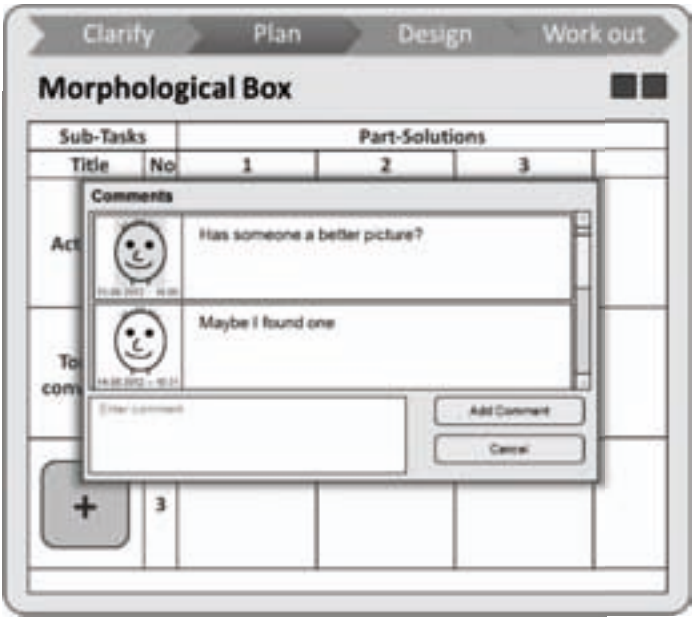


Figure 3: Principle schematic of the direct blog communication

platform, e.g. as a communication user-to-user, as a real blog (like a forum) and as a chat. The blog functionality can further be implemented on its own, but also integrated in several tools like a comment-system shown in Figure 3. This enables to communicate e.g. about one entry of a Morphological Box, directly in this view and concentrates the flow of communication what accelerates the access to specific information.

4.3 Knowledge Management

Knowledge management is one of the most import things enterprises should take care of. The demographic change, with less young employees and older and older getting employees, makes it necessary to save know-how of older employees, because it is not

possible anymore to transport know-how from only one person to another one. Especially older engineers have a lot of expert knowledge and experience. It is probably not possible to prepare all knowledge in a way to make it accessible for the company. But if only some main parts of the knowledge, like concepts and ideas would be available and requirements and failure-know-how would be documented, assigned and comfortable findable, the enterprise would be much more ready to face the future.

As shown in section 3 several parts of this integrated system follow a continuous improvement process. Therefor the used elements to work with are only derived from master copies and will be updateable if the master copy has changed.

One main part of the planed knowledge management is a Wiki with a rights and role management described in sub-section 4.1 and the mentioned advantages.

As first step of the development process of the VDI 2221 [VDI93], it is necessary to clarify the task. The main result of this process is the Requirements List, which will be one tool of the system.

Another very helpful element is the Morphological Box, which is (most often) a tabular tool to collect part-solutions for sub tasks. Normally this is a big sheet of paper which is generated in some creativity meetings. By using a web based system, it is now possible to think of different ways which are similar to some aspects of the ‘open innovation’ idea (s. Figure 4). Exercises like splitting tasks in sub tasks as also generating part-solutions can now be done and entered from everywhere and asynchronous. So the Morphological Box can now grow and get more detailed over the time. The rights and role management mentioned above helps to secure this know-how. By linking part-solutions directly to the wiki it is possible to generate primitive first basic versions of new wiki articles automatically after entering new solutions. Also backward linking to the Morphological Box is possible and seems helpful. Another supported output are Development Catalogues, where a user can find different part-solutions for only one task. Those Catalogues have normally more detailed information so that it is easier to compare and work with the different solutions. Conventional Catalogues needed all the information on one large Sheet for quick access but webpages can present the same information more clearly on different pages. The quick access is guaranteed with links. Thinkable is also to enable the users to add columns to a catalogue. Like the Wiki article could those catalogues be generated in a first version out of the Morphological Box. A linking to the Box and also to the Wiki should exist. Also an update functionality between mapped Boxes and Catalogues would be very helpful to combine the benefit. But one main benefit would lie in the possibility of supporting the build-up process of a new Morphological Box, when word suggestions would be offered for entering the sub-tasks and part-solutions and also to offer a complete list of solutions based on already existing Boxes and Catalogues.

If the Morphological Box is filled complete enough, users can in the next step develop solution variants for the total task. For each solution they have to combine one part-solution for each sub-task by starting in the first row and ending in the last one. The

different solution variants should be then rated (s. sub-section 4.4) and analysed so that maybe some other and better variants can be developed.

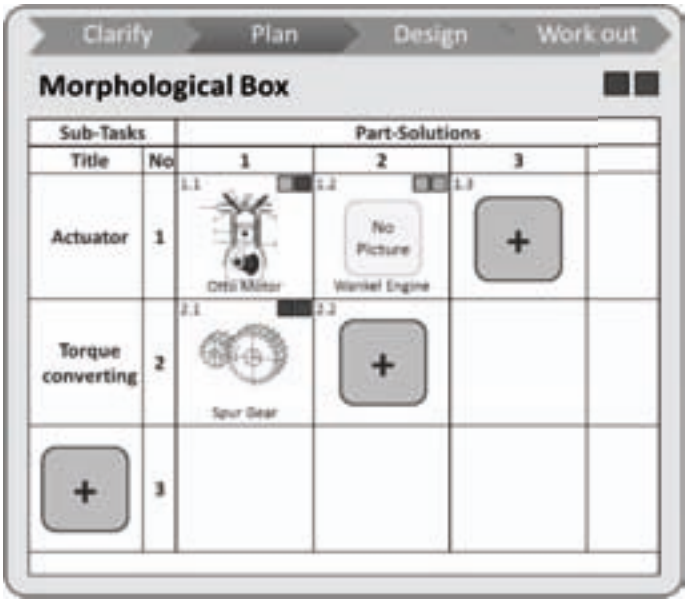


Figure 4: Exemplary Morphological Box

The solutions and their ratings are further important parts of knowledge which now can be accessed from anyone and everywhere provided the required rights. If the future brings any new technologies / innovations or one user has another idea, the Morphological Box can be extended. It is then only necessary to check potential new solutions and rate them as well to actualise the information.

4.4 Decision support

Decision support means components which help to decide and to differentiate between opportunities. For differentiation of criteria of comparability the Pairwise Comparison[3] is one possibility. Because it is in this case implemented in a web based system the comparison process can also be accelerated by using the improved pairwise comparison version [Oe12], which reduces the amount of decisions to make, dramatically.

For rating different solutions the system will have a benefit analysis component. For each solution and every criterion users have to rate the solutions on a defined scale about how much them fulfil each criterion. These values multiplied with the percentage weight

[3] Criterions to compare are listed rowwise and colwise in a table. Each row gets then compared and rated (+1 better , 0 equal, -1 worse) with the crossing column. The rowsum divided by the total sum results in a percentual weight for each criterion.

of the pairwise comparison will end in a percentage rate for each solution and allows a quantified comparison.

4.5 Creativity support

For innovative products creativity is the key. There are several different famous techniques to support the creativity like e.g. Brainstorming, Brainwriting and the Delphi-Method [Eh03]. Because it is not planned to implement audio or video supporting tools, Brainstorming is ruled out. Brainwriting however is predestined for an implementation. In the conventional way people would meet and write their ideas on small cards. The cards are then sticked up on a wall and can be afterwards grouped and extended. All of this is usually organised by a moderator.

For supporting the creativity, the planned tool works similar to a chat-system. Ideas can be entered and will be seen of every logged in Member directly in the chat-box. One main difference is that it is not visible who and when an idea was posted, what should increase the willingness to post own ideas.

Also one person should fulfil the moderating part. This person is able to post comments like “We need more.” or “What would it be if you describe it in a negated way?”. Those comments should be saved in the database as well, so that the comments can be entered and posted directly or be chosen from a list of existing comments. Another idea in this topic is to automate this process. After the Brainwriting session is started, the server could measure how many ideas are posted in which distance of time. If the distance hits a threshold value, a comment could be automatically, randomly selected and posted to push the creativity again.

The moderator also ends the first creativity part. Afterwards the users are enabled to group the ideas and to give them meaningful names/titles. If necessary the moderator can start another round to extend the amount of ideas.

The described Brainwriting tool works synchronously. It is also conceivable to use the same tool in an asynchronous way. Therefor it should be configurable so that the component supports both ways. The asynchronous part would probably make a moderator needless, so functions like grouping and renaming should be accessible right from the session start. For an asynchronous Brainwriting a maybe helpful property would be to define a time stop when the session ends automatically like e.g. ‘in four days’.

For both ways it is possible to use the idea of ‘open innovation’, so that not only the project team is invited but also maybe the total enterprise[4].

The possibility of using this data directly in other components like e.g. the Morphological Box should be clear.

[4] Open Innovation means normally a free access also from outside of the company. Most Enterprises of Mechanical Engineering are normally very suspicious and conservative so that an ‘inhouse open innovation’ variant might be more accepted.

Later on, the Delphi-method could be implemented as well. Possible is to think of inviting (extern) experts temporary to such a creativity session with very less rights. The standard procedure of a Delphi session is to ask experts for specific solutions, to review them and to deliver the reviewed solutions afterwards to all of them again, to get a better expertise. An implementation should be very easy.

4.6 CE – Compliance support

With the CE declaration of conformity every enterprise guarantees the conformity with all necessary regulations and laws. It is obliged to put the CE mark on products that are sold inside the European Union. An FMEA (Failure Mode and Effect Analysis) helps to fulfil here for necessary requirements.

To support also this step it is planned to implement a component which lets the users accomplish an FMEA inside the web system. It works also in a master copy way like described before so that further problems or difficulties are collected centrally.

FMEAs are tabular views that include for each possible failure / problem / difficulty a description of the problem, a description how to handle it, three values on a scale from 1 to 9 for the probability of occurrence, the probability of detection and the value of importance and the risk priority value which is the mathematical product of those three values.

The FMEA component should also have an integrated blog system. Comments would maybe help, but blogs for improvement ideas, mapped directly to one FMEA or even one entry, analogue to the requirements list, could accelerate the improvement process much more.

5 Conclusion

The presented system includes a project management functionality extended with key elements of the methodological approach. These elements allow the reuse of data e.g. entered in a Requirements List in a Pairwise Comparison and afterwards the extended data in a benefit analysis to enable decisions based on quantified values. Entries in Morphological Boxes are mapped to Development Catalogues and to a Wiki and can be extended from everywhere and anytime what should push the amount of, and know-how about, different possible solutions.

The different right and role management guarantees an easy administration, where every user can control the data access for data he is authorised to. This enables the total enterprise to participate of ideas made somewhere else but also allows to keep information secure.

The channelled and mapped comments in blogs enable the users to access information where they belong, what should accelerate the access and enables the enterprise later to analyse and improve their development process. The Activity stream further guarantees

every user to be informed about everything necessary in the company and in his projects but also protects him from unnecessary information.

Creativity is supported by a Brainwriting component and the Morphological Box. The Brainwriting component allows integrated synchronous and asynchronous creativity sessions, which can be restricted so only team members can deliver their input, but also be open for the whole enterprise to participate. This ‘open innovation’ part opens a much bigger pool of creative people what raises the probability to get real innovative ideas and pushes the technology leadership.

References

[Eh03] Ehrlenspiel, K.: Integrierte Produktentwicklung, Hanser-Verlag, 2003

[ESV01] Eversheim, W.; Schröder, J.; Voigtländer, C.: Intranetbasiertes Entwicklungshandbuch – der schnelle Weg zu Neuprodukten: Konstruktion 5-2001, Springer-VDI Verlag, 2001

[Ha12] Hackermeier, I: Wikis im Wissensmanagement: Determinanten der Akzeptanz eines Web 2.0 Projektes innerhalb eines internationalen Zulieferers der Automobilindustrie: Dissertation, München, 2012

[Oe12] Oellrich, M: Improved Pairwise Comparison, WASET, Amsterdam, 2012

[PBF05] Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.H.: Konstruktionslehre, Springer-Verlag, 2005

[VDI93] Verein Deutscher Ingenieure: VDI 2221 - Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte: VDI-Richtlinien, Mai 1993

[VDI10] Verein Deutscher Ingenieure: VDI 2800 –Value Analysis: VDI-Richtlinien, August 2010

EMRAgis – Analytical Risk Assessment Tool for Community Risk Reduction

Stephan Mann, Christoph Gurath, Sebastian Stahn, Richard Gülde

University of Applied Science Magdeburg-Stendal,
Otto-von-Guericke University Magdeburg
Safety and Hazard Defense

39106 Magdeburg
info@emragis.de

Abstract: EMRAgis is a modern risk assessment tool. It is developed as an innovative fire prevention strategy for "Community risk reduction" (CCR) in Germany. The primary objective is to deal the assessment of personnel and material requirements of the local fire services based on reproducible empirical mathematical algorithm. These algorithms quantify the risk in a municipal area of discretion and deriving the demand of fire services. Therefore empirical approaches and efficiency investigations describe necessary measures. Simulations with geo-information systems of the improved measure allow quantified evidences of their efficiency for the local fire protection value. This new level of risk assessment needs a wide data collection. For an effective data handling a flexible and high-performance web-application had to be developed.

1 Introduction

1.1 Need of new kind of risk assessment

In the next few years German cities and communities will have more difficulties to solve problems with their assigned tasks. The last published report about the capability of the common fire services shows that the quantity of volunteer members decreased in the last ten years about 14%. Furthermore 96% of all fire services in the state of Saxony-Anhalt are not ready for action on workdays [LSA13]. As the local fire prevention authorities they are obliged by law so set up and retain a functional and public fire service corresponding with the fire protection requirement plan [LSN12]. With the increasing scarcity of funds and the simultaneous demographic change it is an almost insoluble task to gain the socially required security level within the community. In the previous system for creating a fire protection requirement plan functionaries of the fire services were able to have some influence on the plan with their general knowledge and years of

experience. The purpose was to clarify the structural alignment of the fire defense to the political representatives. By the ongoing realignment within the municipal administration and the growing pressure for financial savings potential more and more fire services are confronted with the demand for profitability and a more functional deployment. Furthermore the delivery of engines for the fire services in different German states is based on a fire risk analysis. If the analysis would not be accepted of the administration no delivery is possible [LSA11].

In addition to volunteer fire services and those with full-time fire fighters are affected and professional fire services who are in regions with falling population and a lack of economic growth. For those cities and municipalities it is not just a problem to maintain the necessary infrastructure for firefighting but also to handle the apparently immoderate number of staff and connected with that the labor cost. Fire services see themselves now compulsorily to justify their plans and their deployment, which are hard to implement because of a lack of methodical rudiments while planning a fire protection requirements plan.

With the constraint for savings the municipality has to pay attention to the functionality of fire prevention measures. That raises the question, if fire services are the financial ruin of a municipality or if they are a chance for economic growth because of an existing security level demanded by society.

In this predicted financial area of tension between fire services and politics started 2010 the special service for fire brigades Pirna as a leading role for Saxony. Together with students with the major „Safety and Hazard Defense“ from the Otto-von-Guericke university Magdeburg and the University of Applied Science Magdeburg-Stendal arose a pilot scheme for developing a forward-looking method for fire protection requirement plans. In the 2 years duration of this scheme and cooperation with the Institute of Fire Services Saxony-Anhalt (IdF LSA) and the Fraunhofer Institute for traffic and infrastructure systems (IVI) was the model EMRAgis developed. The abbreviation EMRAgis stands for empirical mathematical risk analysis for cities and municipalities – geographic information system. Due to this empirical mathematical method in a community based risk analysis and a replicable logic it is possible to make a statement about the necessary demand for operational technology and personnel. Together with the density ratio of the risk within the municipality, that was ascertained based on analytical gathering, it is possible to optimize the vehicle deployment and the necessary personnel. Thus the best safety level in a municipality is achieved with the lowest cost.

Furthermore, there are important intercepts who allow a regional assistance of operational technology for bordering municipalities. As a result there is a regional capability of fire service appropriate plannable taking into account personnel and economic aspects. At the same time does the travel time analysis has the same importance as the systematic integration of the latest scientific perceptions in the field of fire prevention and demographic change.

1.2 Project group

The project group consists of four students, studying safety and hazard defense in the last bachelor term. The course of studies aims to train executive managers for the civil protection as well as safety engineers for industry. Thus, all members of the project group are active firefighters and research on the theme of risk assessment and fire prevention requirements since three years. During this time, a large network of collaborations has been created, for instance with the fire research center in the town of Heyrothsberge or with the Fraunhofer institute of traffic- and infrastructure systems in the town of Dresden.

2 Demands on the software support

2.1 Preliminary considerations

The new method for risk analysis, the so called EMRA model, was developed for the fire protection requirement plan Pirna (Saxony, district Saxon Switzerland-East Erzgebirge). With the ambition to raise the capability of fire protection in Pirna and the problem of unavailable personnel during weekdays or bad accessibility of certain districts within response time [PIR05]. Therefore the project group determined requirements for the new fire protection requirements plan.After a critical literature research the project group was certain that the risk analysis by Prof. R. Grabski from the Institute of Fire Services Saxony-Anhalt [GP07] fits best for determining the equipment for fire services in Pirna.

This risk analysis of the Institute of Fire Services Saxony-Anhalt is based on the classification of buildings. That means that all buildings have to be known by address and some additional information. The emerging amount of data sets was the first sign for the project group to use a software based acquisition.To solve the second problem of Pirna, the unavailability of personnel, was no usable model known. Therefore the project group developed her own personnel analysis. This asks personal data of the staff of fire services from Pirna. Among details about place of residence, job location and the fire service training was the query of availability for different times of day given priority to.With reference to a total strength of the fire department Pirna of 140 fire fighters was it unavailable to use a software based acquisition and evaluation as well. The following figure shows the necessity of a software assisted method for a risk analysis for Pirna.

Figure 1: Necissity of software assistance

For an estimation of the amount of data were this assumption made:

- 42 information per staff member
- 15 information per object
- 5 basic data per fire department

With this estimation was certain, that not just a software assisted acquisition was needed but also a software assisted data evaluation.

2.2 Definition of requirements

The main demand at the data acquistion was the unlimited practicability of the data input by the personnel on site in Pirna. In other words the use by users without profound knowlegde in using computer and without knowlegde of databases. The data input had to be clear and lucid for the user. For the acquisition of the analysis was also a requirement that the underlying algorithm is allready carried out in the gathering of the programm in order to ensure the accuraccy of the analysis. Even a secure performance of the software was provided.

Finally a software for data processing had to be used. Either one that allready existed in the software pool of the fire department Pirna or one that could be used with the existing computersystems without additional costs.

The figure shows the requirements on the data- management- system.

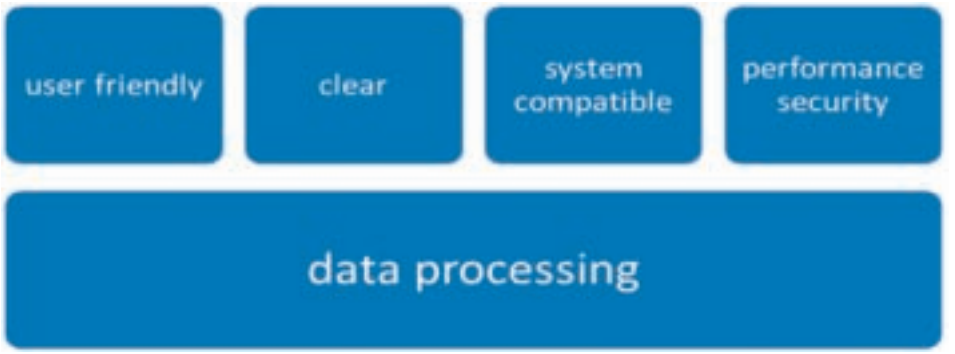


Figure 2: data management system

With the completion of the development of the EMRA model by the end of 2012 the requirements of the data acquisition and analysis had to be extended. The project group was able to verify the EMRA model with other municipalities in Saxony and discussed

improvements especially in data processing. In addition the working title EMRAgis arose.

Especially to perform data acquisition with tablet PCs or smartphones and also to examine the data while acquisition on site and to identify errors, new requirements arose, they are shown in figure 3:

- Functioning independent of operation system
- Functioning as a web application

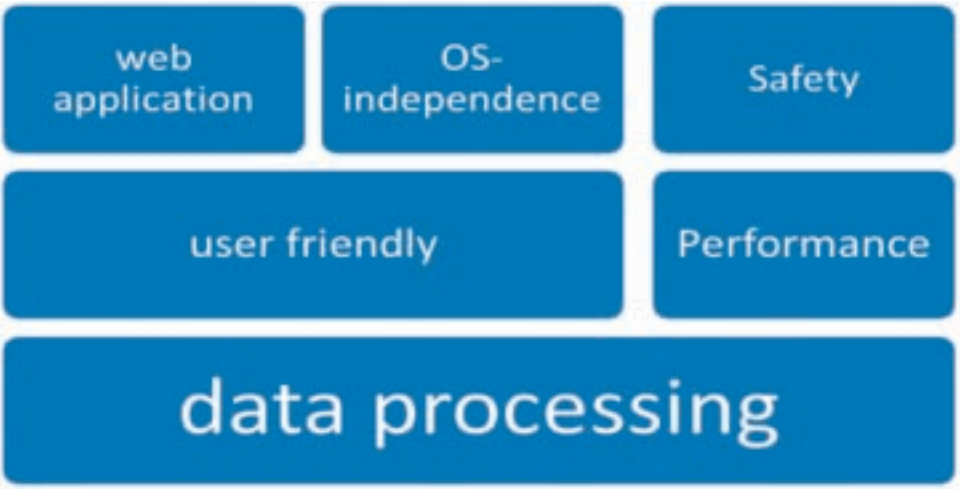


Figure 3: advanced repuirements on the data management system

It should be possible to perform data input with a large number of devices online and offline and there should always be the possebility that the project group can access the data. The requirements allready existing in figure 2 must apply. In summary figure 3 represents the increased demands on the software support for the EMRA model.

3. Implementation of the data processing software

3.1 First step: MS Access

The members of the project group EMRAgis have received a thorough and broad engineering education during their studies, but an in-depth training in computer science is not part of the studies. Thus it was clear to them that they need external help for the software implementation.

The first database application was implemented by a student from Pirna using the software MS Access. Both the personnel acquisition and the object acquisition have been implemented with the database management system. The personnel acquisition was a simple query of personal information without logic operation. The resulting query form included either a text or a yes/no decission.

For the form of the object acquisition was the use of different logic operations between the data input necessary to be able to perform calculations. Depending on the selection of the kind of object and the assignment of a risk level the database application calculates an object specific risk. The necessary data such as kind of object and the degree of danger were determinded by the Institute of Fire Services Saxony-Anhalt and can be imagined as a catalogue [GP07]. To this catalogue of objects will the building on site then be assigned.

The first step in the data base application with MS Acces made the implementation of the EMRA modul in Pirna possible and was a vital basis for the large acceptance of this method in the fire protection requirements plan. However the data analysis was not possible with this simple form and had to be carried out manually.

This major disadvantage and the lack of knowledge of the project group in working with MS Access were the reasons to find other database applications, especially with relation to the requirements in figure3.

3.2 Second step: Python-Tool django

With dedicated support of the Center of Digital Engineering (CDE) of the Otto-von-Guericke university Magdeburg the project group implemented their demands for a database since october 2011 with the Python web framework Django. Both the requirements for the database application (especially the possibility to use it online and the independence of an operations system) and the appropriate performance parameters (object-relational mapper, automatic-admin interface, DRY-principle...) from Django were decisive for this kind of implementation. Based on experience with MS Access it was important for the project group to learn and understand the programming itself (none of the project group members knew a relevant programming language).

The implementation of the personnel acquisition and the object acquisition could be realised with the in Django existing object-relational mapper. The production ready automatic admin interface allows an immediate representation and self-control of programming performance. The stringent realisation of the DRY-principle promotes a traceable source. The existing admin interface and the authentification system enabled the project group to concentrate on the permutation of the context from the beginning without having to create a user interface.After an intensive course in programming of Python, in auxilliary service by the CDE, was the project group able to realise the

personnel acquisition and the object acquisition in Django within one year and achieved the standard of the MS Access forms. In additon the evaluation of the object data could, corresponding with the algorithm of the Institue of Fire Services Saxony-Anhalt, be realised and parts of the personnel evaluation. Within a print ready template the results of the risk analysis from the EMRA model can now be displayed directly without manual evaluation.With the Python web framework the demands of the project group for a database application were fulfilled very well. Especially the operational system independence by the function of the frameworks in any internet browser and associated with that with tablet PCs and smartphones are very crucial for further developments for EMRAgis.

The aim of the project group to create an all-embracing software based application for fire protection requirements plans is possible with Django and in basic approaches realised. In the future the application will be provided online with Microsoft Azure (easy realisation with MS Visual Studio), to make the on site collected data immediately available for the project group. Thus and on site support for the user and a quality control of the gathering can be achieved. Furthermore a geographic information system will be integrated. For this Django provides ready modules.

To sum it up, the Python web framework implements the demand of the project group for a fire protection requirement plan very well and this is the reason for a large positive response to the EMRAgis model of expert groups from fire services and local and state administration.

4 Results

“Community risk reduction“ (CRR) as a fire prevention strategy has existed in Germany for a long time. But there´s one aspect of CRR that we continue to struggle with, and it affects all the other aspects of this strategy: Prospective we must be able to accurately identify what´s at risk before developing or purchasing prevention programs intended to mitigate that risk. These processes require suitable and forward-looking solutions, which consider economic trends and demographic changes. Past practices are characterized by qualitative evidences and subjective grades with justifiability statements in the face of politicians.

Performing a risk assessment tool like “EMRAgis” can be as simple as asking firefighters where they most frequently respond, because what we´ve faced in the past is what we´ll likely face in the future – and so we should be doing something proactive to prevent it. Based on calculus estimates, “EMRAgis” can objectively and conclusively differentiate between the risks in various places in your community. One area may respond frequently to assisted-living centers, while another may have industrial property to protect. Transferring this system in a web-based information system enables “EMRAgis” to safeguard fire prevention with respect to structurally weak areas. In this way the efficiency of fire prevention will be improved and with scientific statements

underpinned. Furthermore supply crunches can be identified and pointedly supported with necessary resources.

All in one EMRAgis allows us to better target the most at-risk citizens and reaching a high level of quality in CRR. In addition it helps to establish locational advantages for a seismic shift in fire prevention. That's why we should investigate EMRAgis further.

References

[LSA13] *Abschlussbericht „Feuerwehr 2020“*, Ministerium des Inneren Sachsen-Anhalt, S. 46, Magdeburg, 2013.

[LSA11] *Zuwendungsrichtlinine Brandschutz*, Ministerium des Inneren Sachsen-Anhalt, MBI . LSA 2011, S. 244, Magdeburg, 2011.

[PIR05] *Brandschutzbedarfsplan der Feuerwehr 2005*, Stadtverwaltung Pirna – Amt 37, Pirna, 2005.

[LSN12] *Sächsisches Gesetz über den Brandschutz, Rettungsdienst und Katastophenschutz*, Sächsisches Ministerium des Innern, SächsGVBI. S. 647, Dresden, 2012.

[GP07] R. Grabski und W. Präger, *Erarbeitung einer Risikoanalyse für die Ausrüstung sowie die Anzahl der zu besetzenden Funktionen einer Gemeindefeuerwehr*, Institut der Feuerwehr Sachsen-Anhalt, Heyrothsberge, 2007 (korrigiert 2011).

Translation: Ulrike Arsand, student of safety and hazard defense

Crowd-enabled Data and Information Management (CDIM)

Wolf-Tilo Balke[1], Andreas Nürnberger[2]

[1]Institut für Informationssyteme (IFIS),
TU Braunschweig
38106 Braunschweig
balke@ifis.cs.tu-bs.de

[2]Fakultät für Informatik,
Otto-von-Guericke-Universität Magdeburg
39106 Magdeburg
http://www.dke.ovgu.de

Abstract: The first CDIM workshop on crowd-enabled data and information management was held in conjunction with 15th GI-Fachtagung Datenbanksysteme für Business, Technologie und Web (BTW), Magdeburg, Germany, 2013.

1 Message from the Chairs

In the recent years algorithms for data and information management, organization, and access have grown quite powerful even over huge and largely unstructured information repositories like the Web. Applications are almost limitless ranging from basic information extraction over knowledge management to complex business intelligence. However, with more complex information processing, retrieval, or mining capabilities also the algorithms' complexity, susceptibility for errors and danger of overspecialization increases.

Since most failings can be traced back to limited cognitive abilities, missing contextual knowledge or heuristics gone wrong, the idea of direct human feedback, supervision and intervention at processing time is currently pursued in many domains. In fact, the need for human assistance in bridging the final quality gap for today's information processing has already given rise to information systems that rely on hybrid architectures. Such hybrid architectures transparently combine the efficiency of current algorithms with the cognitive power and flexibility of humans. Here, generally two design directions are popular:

- Using human input for improving the steps performed by information processing algorithms by providing training samples, answering questions about ambiguous results, or by providing relevance feedback.

- Involving humans directly into the information processing process, explicitly outsourcing some of the required tasks or operators within the process.

The CDIM workshop focuses on raising awareness of new possibilities provided by crowdsourcing techniques and can benefit the community in getting a better grasp on possible applications and future research directions. Bringing together researchers from the database, information retrieval and the data mining communities and offering a platform for discussions within the BTW conference series thus promises to create synergies with mutual value.

Enhancing Named Entity Extraction by Effectively Incorporating the Crowd

Katrin Braunschweig, Maik Thiele, Julian Eberius, Wolfgang Lehner

Technische Universität Dresden
Faculty of Computer Science, Database Technology Group
01062 Dresden, Germany
{firstname.lastname}@tu-dresden.de

Abstract: Named entity extraction is an established research area in the field of information extraction. When tailored to a specific domain and with sufficient pre-labeled training data, state-of-the-art extraction algorithms have achieved near human performance. However, when presented with semi-structured data, informal text or unknown domains where training data is not available, extraction results can deteriorate significantly. Recent research has focused on crowdsourcing as an alternative to automatic named entity extraction or as a tool to generate the required training data. While humans easily adapt to semi-structured data and informal style, a crowd-based approach also introduces new issues due to monetary costs or spamming. We address these issues by combining automatic named entity extraction algorithms with crowdsourcing into a hybrid approach. We have conducted a wide range of experiments on real world data to identify a set of subtasks or operators, that can be performed either by the crowd or automatically. Results show that a meaningful combination of these operators into complex processing pipelines can significantly enhance the quality of named entity extraction in challenging scenarios, while at the same time reducing the monetary costs of crowdsourcing and the risk of misuse.

1 Introduction

Extracting named entities (NE) from unstructured data is an important NLP task in order to provide meaningful semantic search over the data. Such a search functionality is desirable not only for unstructured documents, but for a wide range of semi-structured documents, as well. However, many state-of-the-art NE extraction algorithms are specifically designed for unstructured text, relying on correct grammar and sentence structures. While these algorithms achieve near human performance on corpora that are similar to the training corpus regarding structure and content, they often show a much weaker performance on corpora with unfamiliar document and sentence structures. To improve the quality of NE extraction on these documents, better suited training data is required. However, obtaining a large enough training corpus labeled by experts is not only time

consuming but also expensive.
In contrast, recent research focused on NE crowdsourcing techniques as an alternative to automatic extraction or expert labeling. Therefore, crowdsourcing platforms, such as Amazon Mechanical Turk[1], or service providers, such as CrowdFlower[2], provide a set of tools to distribute tasks to a large number of people in exchange for monetary compensation. Offering a certain amount of flexibility regarding the individual task design, these platforms enable a wide spectrum of crowdsourcing tasks, ranging from simple tagging to more complex jobs such as text translation. However, not all tasks are equally suited for crowdsourcing. On the one hand, technical limitations of the platform itself can have a negative impact on the result quality. On the other hand, crowdsourcing platforms can be affected by spamming or misuse by crowd workers, leading to flawed results.
Previous results indicate that crowdcourcing NE extraction can achieve good results. However, while these works mostly focus on replacing automatic NE extraction by the crowd, irrespective of the costs, we focus on the combination of both approaches in order to enhance the overall extraction performance while keeping the costs minimal. Therefore, the main goal of this paper is to investigate different options for hybrid NE extraction with respect to their costs and effectiveness. With an experimental study on a corpus of semi-structured documents, we show how NE extraction can benefit most from the processing power of the crowd

The rest of this paper is organized as follows: In section 2, we analyze related work on both, automatic as well as crowd-based NE extraction, in order to identify the strengths and weaknesses of both alternatives. Based on these characteristics, we establish general options for hybrid NE extraction approaches in section 3. These options form the basis of our case study, which is presented in section 4. Finally, we summarize our findings and point out directions for future work.

2 Named Entity Extraction

Named entity extraction is an established part of many information extraction systems seeking to identify and label subsets of a text with categories such as *Person*, *Organization* or *Location*. In the following sections we distinguish between automatic approaches that have been studied extensively and the crowd-based extraction which emerged during the last three years.

2.1 Automatic Extraction

Many years of research have lead to great variety of automatic named entity extraction techniques. A common classification discriminates between rule-based

[1] www.mturk.com
[2] www.crowdflower.com

and learning-based approaches. Rule-based techniques, such as ANNIE [CMBT02] or SystemT [CKL+10], require the manual construction of grammar rules in order to extract NEs. In contrast, learning based approaches usually require a large annotated corpus in order to train a model based on discriminative features of entities in the corpus. In [NS07], Nadeau et al. present an extensive list of such learning-based approaches, including supervised approaches such as Conditional Random Fields or Support Vector Machines. While state-of-the-art algorithms tailored to a specific task achieve very good results, they are significantly less effective when presented with a new domain or text genre. Adaption to a new setting often requires new rules or new training data, which can be very expensive and time-consuming.

2.2 Crowd-based Extraction

As an alternative to automatic NE extraction, some researchers have focused their attention on crowdsourcing in order to obtain labeled training data from documents where classic extraction approaches fail and expert labeling is too expensive. Lawson et al. [LEPYY10] use Mechanical Turk to annotate emails, while Finin et al. [FMK+10] used both, Mechanical Turk and CrowdFlower to extract named entities from Twitter messages. Both text types differ in structure and style of writing from the classic type of newswire articles. In [VNFC10], Voyer et al. combine expert labeling with non-expert annotations from the crowd to obtain high-quality annotations at a lower expense. All of these examples use different task designs and layouts to present the task to the crowd. This highlights one of the main challenges when using a crowdsourcing approach: identifying the optimal task design. Layout, task granularity and additional parameters such as fee per unit, level of redundancy or batch size may influence the quality of the result, the overall costs as well as the processing time. The task of NE extraction can be seen as a single task or as a combination of two subtasks: 1) locating the span of a NE in the text, and 2) selecting the correct type of the NE. Finin et al. follow the single task approach in [FMK+10], using a form containing radio buttons to label each word in a Twitter message.[LEPYY10] and [YYSXH10] both use custom layouts which enable span selection to first locate potential NEs. Afterwards, they use drop-down menus and toggle buttons, respectively, to select the correct entity type. Voyer et al. [VNFC10] only leverage the crowd for the subtask of typing, leaving the task of span selection to experts in order to achieve a higher quality.
To improve the result quality, most crowdsourcing platforms offer some form of control mechanism. Mechanical Turk offers quality assessment tests, which workers have to pass, before being eligible to perform the actual tasks. In contrast, CrowdFlower uses gold data with pre-defined answers, which are mixed with the actual tasks to randomly check the quality of the worker's answers. If a worker fails too many gold questions, all of his answers are rejected. Another common control mechanism is inter-annotator agreement, where the same task is presented to several

(a) Extraction (b) Extension (c) Verification

Figure 1: Basic Operators

workers and an answer is only accepted if a sufficient number of workers agree. The level of redundancy as well as the level of agreement can be controlled by the task designer. Still, not all tasks are equally suited for these control mechanisms. Some tasks, such as span selection, are more likely affected by spam and misuse, since it is difficult to define correct gold data. Additionally, as noted in [LEPYY10], the fixed payment system of most crowdsourcing platforms can have a negative effect on NE extraction, as it does not encourage a high recall. Workers are paid per processed document, not for the number of NEs that have been extracted. For a worker, there is no difference between selecting one entity or ten. Mechanical Turk offers a bonus system, which allows the payment of a small bonus in addition to the regular fee in case a worker has performed very well and, for example, extracted a high number of NEs. However, such a bonus system is not available on all platforms.
Using crowdsourcing as an alternative to automatic processing introduces significent costs. Therefore, researchers have also studied different approaches to limit the costs using hybrid processing techniques. Wang et al. [WKFF12] used a hybrid approach for the task of entity resolution, with machines performing the initial matching and humans verifying only the most likely matches. Similarly, Demartini et al. [DDCM12] first use automatic matching for the task of entity linking and the crowd afterwards to improve the quality of the links.

3 Hybrid Extraction

Based on the different characteristics of automatic and crowd-based NE extraction, our goal is to further enhance the performance by combining both techniques into a hybrid approach. In addition to the quality, measured by precision and recall, we also focus on the effectiveness and quality-cost-ratio. It is important to note that the costs only refer to the monetary costs of the crowd-based tasks. The overall quality of a hybrid approach therefore depends on the three following parameters: the precision and recall as well as the total crowdsourcing costs. The optimization goal is to maximize precision and recall while keeping the costs minimal. Since it is very difficult to optimize all parameters simultaneously, we start with addressing each one individually.
In any case, the baseline for a hybrid extraction approach is formed by a classic NE extraction step, illustrated in Figure 1a. The input is an unlabeled document and

the output is a list of extracted entities or the labeled document. The extraction step itself can be either automatic or crowd-based. In case of a crowd-based approach, the costs can be calculated using the following cost function, where n_r is the redundancy level and n_b is the batch size.

$$C = \left\lceil \frac{U_{input} \times n_r}{n_b} \right\rceil \times c_a \tag{1}$$

The redundancy level determines how many workers are asked to perform a specific task, while the batch size determines how many of these tasks are grouped into a single assignment. Fees, depicted as c_a, are usually paid per assignment. U_{input} is the total number of individual units of work, which are presented to the crowd for processing. In most NE extraction applications, a unit of work is a document that needs to be labeled. If the extraction step is split into several consecutive crowdsourcing steps, the overall costs are determined by the sum of the individual costs.

3.1 Increasing Recall

Starting from an unlabeled document, an extraction step should always increase precision and recall. Obviously, the results of several extraction steps can be combined. However, it is hard to make any assumptions about the overall precision and recall. Instead, we focus on another type of processing step, which extends the list of named entities. Based on a list of entities extracted by a previous processing step, the extension step aims to extract additional entities that have not been identified before. As illustrated in Figure 1b, this type of processing step adds additional labels to a partially labeled document. Unless the respective extension algorithm is ineffective, we can expect overall recall to increase. Unfortunately, we can make no certain assumptions about its effecton precision. Again, the extension approach can be either automatic or crowd-based. The cost function remains the same as for the extraction step.

3.2 Increasing Precision

We also want to increase precision, which basically requires the removal of false positives from the set of extracted NEs. This can be achieved by adding a verification step, which is illustrated in Figure 1c. This step removes all labels from the document, which are identified as incorrect or conversely, which can not be identified as correct. Similar to the other processing steps, verification can be performed automatically or by the crowd. Given the applied technique is effective, this step increases the precision. Recall is either constant, if only false positives are removed, or decreases, if the approach is imprecise and removes not only false positives but also some true positives. Again, the cost of a crowd-based verification step can be calculated using

Equation 1, this time considering a single NE as a unit of work.

3.3 Reducing Costs

It is clear that the cost of a processing step based on crowdsourcing mainly depends on the number of units of work U_{input} that need to be processed. In some cases, U_{input} itself depends on the dataset (i.e. number of named entities contained in the dataset) as well as the performance of previous processing steps. While extraction and extension steps potentially increase the number of labeled NEs in the document, the opposite holds for verification steps. Verification reduces the number of labeled NEs and therefore also reduces the costs of subsequent crowdsourcing steps. Since every crowdsourcing step adds to the costs, it is important to only add them if they are effective and justify the investment.

We have identified three different generic classes of processing steps, each with different effects on precision, recall and costs. While extraction steps can be performed individually, the other two obviously only make sense in combination with anextraction step. We argue that combining these different processing steps into complex workflows can significantly improve the performance of NE extraction in challenging scenarios or domains, without necessarily increasing the total costs. To study the effect of the different classes in more detail and to confirm our theoretic assumptions, we conducted an extensive case study on real world data. We present the results of this study in the following sections.

4 Experimental Evaluation

4.1 Dataset

For our study we used a dataset containing 50 sets of metadata collected from the Open Data platform *data.gov.uk*. For each dataset a set of metadata is stored to support the retrieval of the dataset. An example is shown in Figure 2. Each set of metadata contains a list of properties, which we limited to the seven properties depicted in the example. The 50 documents have been selected from the platform with respect to their topic or publisher. We selected four topics with 10 related documents for each topic. A fifth group of 10 documents was added, containing randomly selected documents. To enable performance evaluation, all documents have been labeled manually. The gold data contains a total of 427 (162 distinct) named entities, including hierarchical NEs and abbreviations. Of these NEs 11 (11) are of type *Person*, 162 (47) of type *Location*, and 254 (104) of type *Organization*. Occurrence of a named entity is only counted once per property section. Organization is the most common type in the dataset. Person names are very rare. We found this dataset to be suitable for our study because it combines several features which cause

Figure 2: Example Dataset

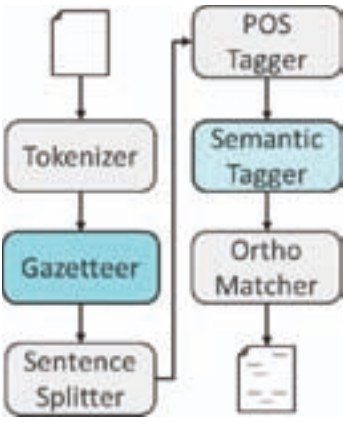


Figure 3: ANNIE Processing Pipeline

automatic extraction approaches to fail. First of all, it contains sections of different text genres. As shown in Figure 2, the section *Notes* contains classic unstructured text, while others, such as *Title* or *Description* contain only sentence fragments. The section *Tags* consist of a list of words or hyphenated phrases. Second, capitalization rules for proper names are not observed consistently. While some names appear in lowercase, other nouns are capitalized for emphasis.
Due to the described characteristics the named entity corpus is very ambitious. The dataset contains many difficult organization names, including hierarchical NEs. For example, the organization name "Taunton and Somerset NHS Foundation Trust" also contains the location names "Taunton" and "Somerset" as well as the organization name "NHS". Additionally, the corpus contains a large number of abbreviations of both location and organization names, such as "DEFRA". In some cases, nonstandard shortened versions of organization names are used, which are very difficult to detect. Due to the high percentage of hierarchical NEs and abbreviations in the corpus, it is rather unlikely to achieve a very high recall.

4.2 Crowdsourcing Environment and Setup

All crowdsourcing tasks have been created using the CrowdFlower Builder and executed on Amazon Mechanical Turk via CrowdFlower. We tried to keep the external conditions of our experiments as stable as possible, posting new tasks at the same time of day (2-3 pm, GMT) to the same platform (Amazon Mechanical Turk). The list of countries, which specified which workers were eligible to perform our tasks, was also kept unchanged throughout our study. We focused on a list of predominantly English speaking countries to reduce the risk of misinterpreting the task description.

We also kept the internal setting constant for all experiments. If not stated otherwise, we set the redundancy level n_r to 3 and batch size n_b to 5. The minimum inter-annotator agreement was set to 2, meaning that at least 2 out of the 3 workers performing the same task would have to agree on the answer, otherwise the result was discarded as *undecided*. Due to the costs involved, all crowdsourcing tasks were performed only once per experiment. Therefore, the results do not represent the statistic mean.

4.3 Extraction Algorithms

As the baseline of our experiments, we tested both automatic as well as crowd-based extraction individually.

4.3.1 Automatic Extraction

As an example of an automatic NE extraction algorithm, we used the rule-based approach ANNIE [CMBT02]. We decided on a rule-based technique, because the interpretation of the algorithm's behavior is easier for rules than for statistic approaches. It is important to note that statistic approaches, which were not specifically trained for our corpus, did not achieve significantly better results than the rule-based approach. ANNIE extracts entities from documents using a pipeline of different processing steps, as illustrated in Figure 3. Both, gazetteer lists and regular expression-like rules, are used to identify NEs of various types. The performance results for ANNIE are presented in Table 4.

4.3.2 Crowd-based Extraction

As mentioned in Section 4.3.2, there is a wide range of options, how to present the overall task of named entity extraction to the crowd. The task design has a direct impact on the monetary costs as well as potential impact on the result quality. We studied different layout designs and settled on splitting the task into two subtasks: 1) the location of NEs, and 2) the type selection.
For the first subtask, we asked workers to mark the span of potential NEs in each document, without the need to decide on a type. Due to the size of each document, we did not batch several task into a single assignment for this step. The layout of this task is shown in Figure 6a. In the second task, we asked workers to select the correct type for each NE from a drop-down menu. To provide some context, we display the section of the document, which contained the NE. The layout for this tasks is shown in Figure 6b. Due to its smaller size, we batched the second task, with each assignment containing 5 NEs. For each task we used a redundancy level of 3. The type selection required an inter-annotator agreement of at least 2. However, for the complex task of marking the span of an NE in the text, we did not take agreement between annotators into account and accepted all labels to increase

	ANNIE		Crowd	
Topic	P	R	P	R
CO	**0.67**	**0.59**	0.56	0.17
DEFRA	0.57	0.27	**0.76**	**0.48**
MISC	**0.57**	**0.55**	0.52	0.34
NERC	0.72	0.33	**0.77**	**0.42**
NHS	0.65	**0.42**	**0.7**	0.29
All	0.64	**0.42**	**0.68**	0.34

Figure 4: Precision (P) and Recall (R) for NE extraction using ANNIE and the crowd, respectively.

Experiment	P	R	C
Abbreviations	0.67	0.5	0
Crowd (Tags)	0.65	0.48	70
Crowd (All)	0.63	0.61	285

Figure 5: Precision (P), Recall (R) and Cost (C) of Extension Steps in Combination with ANNIE.

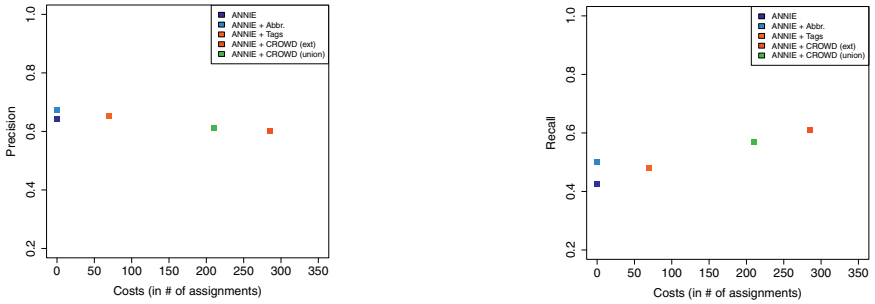
recall.
The span selection task proved very difficult with respect to quality control and spamming. Due to access constraints on Mechanical Turk, we could not make use of the bonus system to increase recall. It was also difficult to incorporate the control mechanisms provided by CrowdFlower. Therefore, we had to perform the labeling step without any quality control mechanisms, which resulted in a high number of empty documents in the result set. For the classification step, we added a set of gold data to enable quality control. The performance of this experiment is also presented in Table 4. The crowd-based extraction requires the processing of 210 assignments by the workers on Mechanical Turk. In our study, we paid workers \$0.01 or \$0.02 per assignment, depending on the difficulty of the task.
Overall, the crowd-based approach achieved a slightly higher precision than ANNIE, but at a lower recall. Still, the results are very similar. However, analyzing the performance of each approach separately for each of the 5 groups of documents described in Section 4.1 reveals significant performance differences. Especially the performances for groups *CO* and *DEFRA* indicate that the different extraction approaches do not behave similarly for all document. Based on this observation, we expect a hybrid approache to perform better than each individual approach. Even a naive union of both result sets achieves a precision of 0.61 and a recall of 0.57.

4.4 Extension Approaches

To increase the recall of our NE extraction task, we studied three extension approaches, an algorithmic approach and two crowd-based approaches. For the algorithmic approach, we used the high number of abbreviations in the corpus as motivation. Both extraction approaches described in the previous section, identified only a small percentage of these abbreviations. Therefore, we used a simple algorithm to identify additional abbreviations. The algorithm takes previously labeled entities of types *Location* and *Organization*, which contain at least 2 words, as input

(a) Span Selection (b) Type Selection (c) Type Verification

Figure 6: CrowdFlower Tasks



(a) Cost and Precision (b) Cost and Recall

Figure 7: Extension Steps

and creates several potential abbreviations following different patterns. For the name "Department for Environment, Food and Rural Affairs", for example, the following five patterns were created: "DEFRA", "DFEFARA", "defra", "dfefara", and "Defra". Afterwards the algorithm searches for occurences of these abbreviations within the document where the original NE was found and assigns the correct type to each occurence. The performance of this extension step (in combination with ANNIE as the primary extraction algorithm) is presented in Table 5.

The first crowd-based extension approach is similar to the crowd-based extraction approach described in Section 4.3.2. However, it is not performed for all documents, only for sections of documents, where a previous extraction step failed to find any NEs. We noticed that ANNIE could not deal with the format of the *Tags* section in the documents, so we asked the crowd to label only these sections. Again, we used the same processing pipeline consisting of two separate crowd tasks as before. Due to the smaller size of the sections, we used a batch size of 5 for both steps. Performance and costs of this approach are also presented in Table 5.

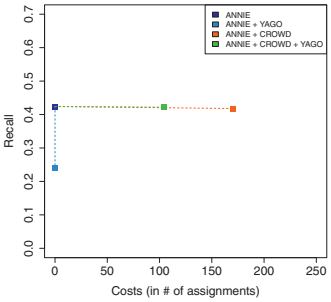
The second crowd-based approach is also very similar to the crowd-based extraction, but with minor differences. This time, previously labeled NEs are marked in the text and the workers are asked to only mark NEs, which have not been labeled

before. The parameters n_r and n_b are the same as in Section 4.3.2, but it is obvious, that this is a more difficult task and likely to produce less precise results. The results in Table 5 show a slight decrease in precision. However, the recall increased significantly compared to the baseline.
Figures 7a and 7b compare the effects of each of the proposed extension techniques on precision and recall, respectively. They also show the costs involved to perform each step. The results confirm the assumption that while each extension step increases recall to some extend, we cannot make any definite assumptions about the effect on precision.

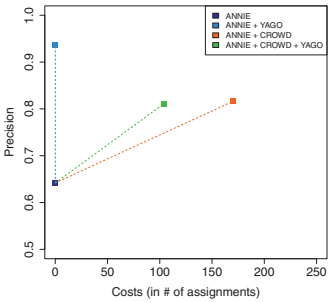
4.5 Verification

As described in Section 3.2, in order to increase precision, we incorporate a verification step which needs to answer two questions: 1) Is the NE in question a proper name, and 2) has the NE been typed correctly? Only if the answers to both questions are yes, the NE is added to the result set. Again, we have tested both, an automatic and a crowd-based approach.
The automatic verification technique is inspired by related approaches in the field of question answering, where evidence from external sources, such as knowledge bases, is collected in order to verify the answer type. In a similar fashion, we collect evidence from the YAGO knowledge base [SKW07] to verify the types of our named entities. YAGO is a structured knowledge base, which consolidates information from various sources, including Wikipedia, WordNet and GeoNames and contains a total of $9,756,178$ entities[3]. The knowledge base can be queried using SPARQL queries. For each type (i.e. Person, Location, Organization), we defined several different queries. We accepted a named entity as correct if at least one query returned a positive match.
For the type *Person*, we searched YAGO for entities of type *person*, whose preferred name matched the NE. Since YAGO contains multiple types with the name *person*, we limited our search to the type extracted from WordNet. Additionally, we tried to extract the first name as well as the last name from the NE and searched YAGO for entities with attributes *hasGivenName* and *hasLastName*.
For type *Location*, we investigated the YAGO WordNet part for entities whose type was *location* or a subclass of *location*, and whose preferred name matched the NE. We also queried YAGO for entities with a *yagoGeoEntity* type or types of the subclasses of *yagoGeoEntity*.
For the type *Organization*, we first looked for entities, whose type was a subclass of YAGO type *organization*. However, for NEs of type *Organization*, this YAGO based technique turned out to be too imprecise, accepting too many false positives. Therefore, we used this approach only for NEs of type *Person* or *Location* in our experiment. Although this approach is expected to significantly increase precision, it also has the disadvantage that named entities from the corpus, which do not appear

[3] www.mpi-inf.mpg.de/yago-naga/yago/statistics.html



(a) Costs and Recall



(b) Costs and Precision

Figure 8: Verification Steps

in any knowledge base, are dismissed, reducing recall significantly. That means that finding no supporting evidence in the knowledge base is no clear indication that the NE is a false positive. The results in Table 10 clearly show the negative impact on recall.

For the crowd-based verification, we presented workers with the NE, its type and its context. We then asked them to verify if the entity in question was of the specified type. The task layout is illustrated in Figure 6c. The task was performed per NE extracted from the documents, using the same parameters as described in Section 4.2. Again, we added gold data for quality control. The results of this verification step (in combination with ANNIE) are shown in Table 10. As expected, the verification step significantly increases the precision compared to the baseline experiment, although to a lesser extend than the YAGO based approach. But, it does not have a negative impact on recall.

As noted in Section 3.3, the costs of a crowdsourcing step depend on the number of units that need to be processed, which in this case is the number of NEs that need to be verified. Our results show that the YAGO based approach can verify some of the NEs with very high precision, but misses some of the correct NEs in the process. We can use this characteristic to our advantage, by combining both verification techniques. As shown in Figure 9, we can use YAGO first to verify all NEs and then use the crowd to check those NEs again, which could not be verified in the first step. Figures 8a and 8b show that this combined approach achieves the same precision and recall as the crowd based verification, but at a significantly lower cost.

4.6 Complex Workflows

After studying the performance of the three classes of processing steps individually, we now take a look at what performance can be achieved when combining multiple

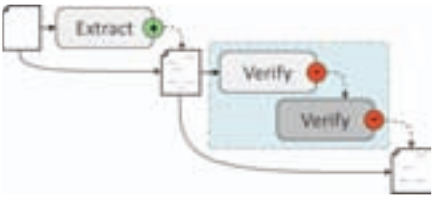
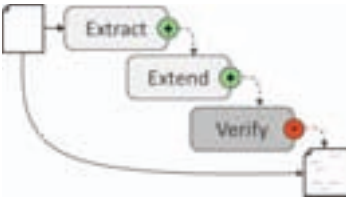


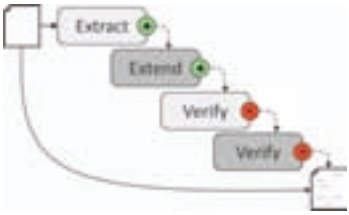
Figure 9: Combination of Automatic (white) and Crowd-based (gray) Verification

Experiment	P	R	C
YAGO	0.94	0.24	0
Crowd	0.82	0.42	170
YAGO + Crowd	0.82	0.42	104

Figure 10: Precision (P), Recall (R) and Cost (C) of Verification Steps in Combination with ANNIE.



(a) Workflow 1



(b) Workflow 2

Figure 11: Complex Workflows with Automatic (white) and Crowd-based (gray) Processing Steps

processing steps into complex workflows. For the first experiment, we combined the ANNIE extraction step with the automatic extension that searches for abbreviations in the document, as well as a crowd-based verification. The pipeline is shown in Figure 11a. Precision, recall and costs for this experiment are presented in Table 12. For the second experiment, we combined ANNIE with a crowd-based extension step as described in Section 4.4. We then used both YAGO and the crowd to verify the results. This workflow contains two crowd-based processing steps, which naturally leads to higher costs. The processing pipeline is shown in Figure 11b and precision, recall and costs for this experiment are also included in Table 12.

The results show that combining several processing steps into complex workflows can significantly improve the overall performance for NE extraction tasks. Figure 13 illustrates the different effects each processing step has on precision and recall. We can see in both experiments, that the extension step increases the recall, while the verification step increases precision. While the first workflow achieves a 30% increase in precision and a 15% increase in recall, compared to the baseline (i.e. ANNIE), the second workflow achieves a 44% increase in recall but only 14% increase in precision. So in a scenario where quality is more important than quantity, the first workflow would be more suitable. If, however, quantity trumps quality, the second workflow would be the better choice. Of course, the results of these experiments do not present the upper limit that can be achieved by combining multiple processing steps.

Compared to the use of crowdsourcing for NE extraction as a simple alternative to

	Workflow 1	Workflow 2
Precision	**0.84**	0.73
Recall	0.49	**0.61**
Costs	191	480

Figure 12: Precision, Recall and Cost of Complex Workflows.

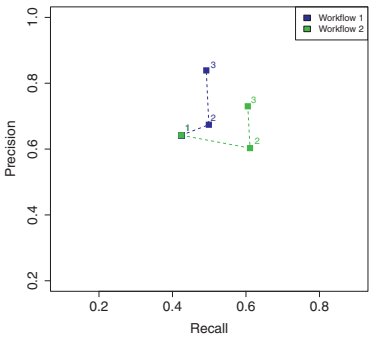


Figure 13: Change in Precision and Recall after Extraction (1), after Extension (2), and after Verification (3)

automatic extraction techniques, we can achieve a much better result for our corpus, using a complex pipeline of processing steps. Although the moderate performance of our crowd-based extraction is partially due to the absence of both, quality control mechanisms and a bonus system, we can still achieve a much better result at lower costs than the original crowd-based extraction. Using workflow 1, we can achieve higher precision and recall at a cost of 191 assignments, compared to the 210 assignments required for the crowd-based extraction.

5 Conclusion and Future Work

Adapting the task of named entity extraction to new genres and domains can be a great challenge, as it often requires sufficient labeled training data for testing and evaluation. Crowdsourcing has been proposed as a cheaper alternative to expert labeling. However, in some cases, the performance of a crowd-based extraction does not meet the expectations, as shown in our example. To address this issue, we identified several additional techniques to enhance the extraction performance. By combining automatic and crowd-based techniques into a complex processing pipeline, we could achieve significantly better results for our test corpus than automatic or crowd-based extraction achieved individually. By investing in crowd-based extension and verification instead, we managed to achieve an increase in both, precision and recall. In some cases, we also reduced the overall costs.
The combination of different processing steps allows for a great variety of different processing pipelines. While some showed a higher impact on precision, others effected recall more. This shows that the extraction pipelines can be tailored to the specific requirements of the application.
Based on our proposed set of operators we would like to extent the approach with a

comprehensive cost and confidence model which allows us to estimate the expected quality-cost-ratios of different workflow alternatives. Given such a model we could choose the best workflow for given requirements on precision, recall and costs, in many ways similar to the hybrid query execution plans proposed by Franklin et al. in [FKK+11].

References

[CKL+10] L. Chiticariu, R. Krishnamurthy, Y. Li, S. Raghavan, F. R. Reiss, and S. Vaithyanathan. SystemT: an algebraic approach to declarative information extraction. In *ACL '10*, pages 128–137, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[CMBT02] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *ACL '02*, 2002.

[DDCM12] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 469–478, New York, NY, USA, 2012. ACM.

[FKK+11] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: answering queries with crowdsourcing. In *SIGMOD '11*, pages 61–72, New York, NY, USA, 2011. ACM.

[FMK+10] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze. Annotating named entities in Twitter data with crowdsourcing. In *CSLDAMT '10*, pages 80–88, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[LEPYY10] N. Lawson, K. Eustice, M. Perkowitz, and M. Yetisgen-Yildiz. Annotating large email datasets for named entity recognition with Mechanical Turk. In *CSLDAMT '10*, pages 71–79, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[NS07] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.

[SKW07] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *WWW '07*, New York, NY, USA, 2007. ACM Press.

[VNFC10] R. Voyer, V. Nygaard, W. Fitzgerald, and H. Copperman. A hybrid model for annotating named entity training corpora. In *LAW IV '10*, pages 243–246, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[WKFF12] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. CrowdER: Crowdsourcing Entity Resolution. In *VLDB*, 2012.

[YYSXH10] M. Yetisgen-Yildiz, I. Solti, F. Xia, and S. R. Halgrim. Preliminary experience with Amazon's Mechanical Turk for annotating medical named entities. In *CSLDAMT '10*, pages 180–183, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Just ask a human? – Controlling Quality in Relational Similarity and Analogy Processing using the Crowd

Christoph Lofi

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
lofi@nii.ac.jp

Abstract: Advancing semantically meaningful and human-centered interaction paradigms for large information systems is one of the central challenges of current information system research. Here, systems which can capture different notions of 'similarity' between entities promise to be particularly interesting. While simple entity similarity has been addresses numerous times, relational similarity between entities and especially the closely related challenge of processing analogies remain hard to approach algorithmically due to the semantic ambiguity often involved in these tasks. In this paper, we will therefore employ human workers via crowd-sourcing to establish a performance baseline. Then, we further improve on this baseline by combining the feedback of multiple workers in a meaningful fashion. Due to the ambiguous nature of analogies and relational similarity, traditional crowd-sourcing quality control techniques are less effective and therefore we develop novel techniques paying respect to the intrinsic consensual nature of the task at hand. These works will further pave the way for building true hybrid systems with human workers and heuristic algorithms combining their individual strength.

1 Introduction

The increasing spread of the Internet and its multitude of information systems call for the development of novel interaction and query paradigms in order to keep up with the ever growing amount of information and users. These paradigms require more sophisticated techniques compared to established declarative SQL-style queries or IR-style keyword queries. Especially natural query paradigms, i.e. those query paradigms which try to mimic parts of natural human communication as for example questions answering [FB10], similarity browsing [Le10], or analogy queries require sophisticated semantic processing. One of these sematic processing steps is determining the *similarity* between two given entities. Broadly, similarity measures can be classified into two major categories: *attributional* similarity and *relational* similarity. Attributional similarity focuses only on the (usually explicitly provided) attributes values associated with entities, and is a well-researched problem with several efficient and high-quality algorithmic implementations, e.g., discovering similar vectors, or computing the similarity between database tuples. This allows for query-by-example interactions or similarity searches; for example, in an e-commerce setting, a user can simply provide an example object (e.g., a mobile phone), and the respective

information system can find other products with similar features and technical specifications [Lo10].

In contrast, *relational similarity* is a significantly more challenging problem. For example, there is a high relational similarity between the entity pairs (ostriches, bird) and (lions, cat) as ostriches are particularly large birds while lions are particularly large cats, i.e. the relations ("*x* is a very large *y*") between both is very similar. This type of similarity measurements is one of the central challenges of *analogy processing*, a core concept of human communication. Being able to reliably process and discover such similarities and analogies allows for a wide variety of new applications, e.g. more effective information extraction, analogy processing,, or certain subsets of question answering. However, actually assessing relational similarity is a difficult and extremely error-prone task for algorithms as well as for humans - current state-of-the-art algorithms achieve an average accuracy of "only" 56%.

Therefore, this paper explores the effectiveness of crowd-sourcing when applied to the challenging problem of relational similarity. The goal of this endeavor is establishing a baseline of human-based performance for further evaluations, and paving the way for later hybrid algorithms which combine algorithmic processing of relational similarity with on-demand crowd-sourcing for improved accuracy and reliability. It turns out that even human crowd-workers struggle with the complexity of assessing relational similarity correctly. This renders many established methods for quality control of crowd-sourcing tasks ineffective, i.e. Gold questions are hard to realize as even the performance of non-malicious workers is quite low (Gold questions: when using Gold questions, the correct answer to some tasks is known upfront. These tasks are transparently mixed into the set of real tasks and all workers which fail to answer the Gold questions correctly are excluded from the overall crowd-sourcing effort). Also, majority voting leads only to mediocre quality. Even more, for relational similarity or analogy, there is no clear "right" and "wrong" as the usefulness of a similarity statement basically depends on consensual agreement. Therefore, the central challenge approached in this paper is the problem of controlling quality of a crowd-sourcing task where individual worker responses are highly unreliable individually and no hard information on a solution's correctness is available during runtime.

The contributions of this paper can therefore be summarized as follows:

- We introduce the challenging problem *analogy processing* and *relational similarity*
- We provide a *baseline* for human performance when confronted with this problem
- We analyze the behavior of crowd-workers with respect to their *reliability* and *performance*
- We extensively discuss several aggregation techniques for improving and *controlling quality* of the final result, suitable for general crowd-sourcing problems relying on multiple choices and a community consensus
- We outline how human-based approaches to solving analogy challenges compare to currently researched techniques for automatically solving the problem performance-wise

2 Evaluation Scenario: Analogy Detection

Most human cognition is based on processing similarities of conceptual representations. During nearly all cognitive everyday tasks like e.g., visual perception, problem solving, or learning, we continuously perform *analogical inference* in order to deal with new information [GM97] in a flexible and cross-domain fashion. It's most striking feature is that analogical reasoning is performed on high-level relational or even perceptional structures and properties. Moreover, in contrast to formal reasoning, deduction, or formal problem solving, the use of analogies (and also analogical inference) appears to be easy and natural to people (contrary to needing a lot of training and experience). As analogical reasoning plays such an important role in many human cognitive abilities, it has been suggested that this ability is the "core of cognition" [Ho01] and the "thing that makes us smart" [Ge03]. Due to its ubiquity and importance, there is long-standing interest in researching the foundations and principles of analogies, mainly in the fields of philosophy, but later on also in linguistics and especially in the cognitive sciences. From here, research slowly spreads to computer science.

In general, an analogy is a cognitive process of transferring some high-level meaning from one particular subject (often called the *analogue* or the *source*) to another subject, usually called the *target.* When using analogies, one usually emphasizes that the "*essence"* of source and target is similar, i.e. their most discriminating and prototypical behaviors are perceived in a similar way. In this paper, we focus on the so-called *4-term analogy model* which is a simpler case of general analogies focusing closely on entities which behave similarly or have a similar role, i.e. those entities which are relational similar. For example, consider the following analogy question: "What is to the sky as is a ship to the ocean?" The obvious answer is 'an airplane', as airplanes are used to travel the sky as ships are used to travel the oceans. The actual differences in their physical properties (like the shape, color, or the material) and their non-prototypical relations to other concepts are ignored, as the ability of ships and airplanes to 'travel a certain domain' is highly similar and dominant. However, the "travels" relation in both term pairs is not the same, as traveling water and traveling air is different in many aspects. Still, by focusing on the important aspects of the relation and ignoring the others, the analogy can be drawn.

This simple type of analogy is also actively researched in linguistics, because many aspects of language evolution and the development of new words via neologisms are based on such simple analogies (think of the time when the word 'spaceship' appeared for the first time: while the word was new, many aspects of its meaning are immediately clear even to people who did not know what a spaceship actually is or how it exactly works). While the example of $ship : ocean :: airplace : sky$ (using Aristotele's notation) is quite illustrative, analogical reasoning can quickly become ambiguous when relational similarity between terms is either not strong enough or too many candidate pairs with similar relational strength exist (e.g. consider "What is to land as is a ship to the ocean?"; here, a correct answer is hard to determine without further information, should the answer be a car, a bus, a truck, an ox cart?). Although being just a simple subcase, the 4-term analogies play an important role in many real life analogies, and are thus very intriguing for computer science research.

2.1 Analogy and Similarity

The relation between analogy and similarity is by many people considered to be a confusing one. This is due to the fact that while analogy is not the same as similarity, analogical reasoning heavily relies on various 'flavors' of similarity. Therefore, sometimes it is claimed that there is the concept of *generic similarity* [Br89], for which the commonly used *property similarity* (what people usually mean when referring to 'similarity'), is just a special case. Other special cases of generic similarity are analogy in terms of relational similarity (e.g. 4-term analogy used in this paper) and structural similarity (e.g. complex analogy).

Like analogy, property similarity also establishes a certain relation between a source concept and a target concept. In this sense, when assuming knowledge provided in form of prepositional networks as in the structure mapping theory, 'normal' (property) similarity can be defined when most of the attributes/properties and the relations of the source are similar to those of the target [Ge83], e.g. if claiming that the "Kepler-30 star system is like our solar system[1]", this is a property similarity statement because both are star systems, both have similar suns, and both show similar planetary trajectories (albeit Kepler-30 has less planets with different properties). In contrast, claiming that "Atoms are like our solar system" is indeed an analogy, as atoms and the solar systems have no similar attributes, but do have similar relations between their related concepts. This difference between analogy and property similarity is quite significant, as similarity is a well-researched problem in information systems, with many efficient and mature implementations already published. Furthermore, similarity can be computed much easier as it mostly relies on attribute values, which are usually readily available. In contrast, computing relational similarity is a little researched problem, and also requires vast and diverse semantic knowledge bases which are difficult to obtain. Therefore, relational similarity is an entirely different challenge.

2.1.1 SAT Analogy Challenges

While not being very complex analogies, the SAT analogy challenges deserve some special attention due to their importance with respect to previous research in computer science. In the study in later sections of this paper, we will therefore also use the SAT dataset for our experiments. The SAT test is standardized test for general college admissions in the United States. It features major sections on analogy challenges to assess the prospective student's general analytical skills. These challenges loosely follow Kant's notion of analogy as relational similarity between two ordered pairs is required, and the challenges are therefore expressed as 4-term analogy problems (but with multiple choices answers): out of a choice of five word pairs, one pair has to be found which is analogous to a given word pair.

As an example, consider this challenge:
legend is to *map* as is
a) *subtitle* to *translation* b) *bar* to *graph* c) *figure* to *blueprint*
d) *key* to *chart* e) *footnote* to *information*

[1] http://web.mit.edu/newsoffice/2012/far-off-solar-system-0725.html

Here, the correct answer is d) as a key helps to interpret the symbols in a chart as does the legend with the symbols of a map. While it is easy to see that this answer is correct when the solution is provided, actually solving these challenges seems to be a quite difficult task for aspiring high school students as the correctness rates of the analogy section of SAT tests is usually reported to be around 57%.

Currently, many research works which deal with analogies in computer science aim at solving the SAT challenges, such as [Bo09, Da08]. This is due to the fact that solving these challenges is significantly easier than dealing with general analogies: basically, the relational similarity for each of the answer choices to the source term pair is computed, and the most similar one is picked. But still, this task is very difficult and far from solved in a satisfied fashion. We outline the current progress of these efforts in comparison to human performance in section 3.3.

3 Crowd-Sourcing Analogy Problems

In the following, we will study the performance of crowd-sourcing in the context of analogy and relational similarity processing. A major focus is on result reliability, a major issue in crowd-sourcing due to malicious or simply incompetent workers. In most simple cases, these concerns can be addressed effectively with quality control measures like majority voting or Gold sampling [Lo12]. In previous studies on crowd sourcing it has been shown that within certain bounds, missing values in database tuples can be elicited with reliable efficiency and quality as long as the information is generally available. Especially for factual data that can be looked-up on the Web without requiring expert knowledge (e.g., product specifications, telephone numbers, addresses, etc.), the expected data quality is high with only a moderate amount of quality control (e.g., majority votes). For example, [Lo12] reports that crowd-sourced manual look-ups of movie genres in IMDB.com are correct in ~95% of all cases with costs of $0.03 per tuple (including quality assurance). Unfortunately, solving analogy problems falls into the difficult complexity class of consensual problems requiring competent workers with certain non-ubiquitous skills. This kind of problem is difficult to approach with simple quality control techniques [Se12].

The problem of controlling quality in this case stems from the very nature of analogies (and is comparable to problems faced in other perceptual tasks like measuring property similarity, finding prototypes, identifying essential properties, etc.). In particular, this is because there is no clear “right” or “wrong” for analogies, but the “correctness” of an analogy statement depends on the general perception and consensus of the people using it and may change over time or between different groups of people.

For example consider: *lions* is to *mammals* as is
a) *crocodile* to *reptiles* b) *shark* to *fish* c) *Queen Elisabeth* to *the English*

Every one of these answers could be argued for, and none is inherently more correct than the others. Therefore, when crowd-sourcing analogy challenges one does not look for correct answerers (which usually do not exist), but for those which people perceive to be the better answer. This unfortunately renders Gold questions, one of the most powerful quality

management tools for factual information unusable as this would wrongfully punish people to voice their opinion on an ambiguous problem. Furthermore, this also means that all other quality control techniques developed for crowd-sourcing relying on the existence of clear truth values as for example probabilistic error models [Li12,Wh09] are inherently unsuitable for analogies, similarity and other consensus-based problems. Therefore, the techniques developed in this paper are heavily based on inter-worker agreements in order to filter out malicious or incompetent workers, and therefore leading to high-quality judgments.

However, as measuring performance and quality of any heuristic for an ambiguous and consensus-based task is difficult to realize, the following study is based on the previously mentioned SAT dataset. Besides containing only simple and easier to handle 4-term analogy problems, this dataset has another unique and exceptional property: as the dataset is used in college admission test, every task is specifically and carefully hand-tailored in such a way that there is only one single non-ambiguous correct answer to every challenge. Therefore, if a worker understands the tasks correctly and knows all involved entities and their relations, he then can indeed derive a correct and undisputable answer. This property is not representative for real-life analogies, but allows us in this study to observe worker performance in a meaningful way.

3.1 Crowd Worker Baseline Performance

The following study is based on a crowd-sourcing experiment on CrowdFlower.com encompassing all 353 challenges from the SAT dataset. For each challenge, 8 judgments from workers which are recruited from the Amazon Mechanical Turk worker pool are elicited. As solving SAT challenges requires a very good understanding of language and vocabulary, we restricted the worker pool to only native speakers from the English-speaking countries Great Britain, United States (from where the SAT test originates, i.e. there

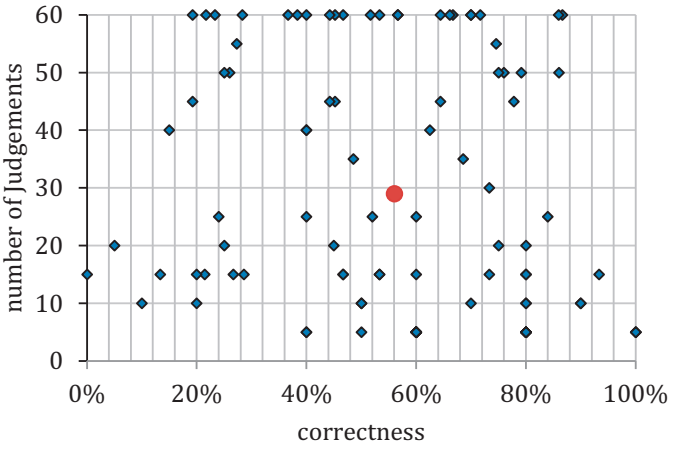


Figure 1. Worker analysis: Number of judgments vs. average correctness for each worker (99 workers overall, marked dot is the average:56% and 29 judgments)

is a high chance that some workers specifically trained solving these challenges), and Australia. Each HIT (smallest work unit) issued to workers consists of 5 analogy challenges for which we paid $0.15 overall (this amount is rather high compared to other crowdsourcing tasks on Amazon Mechanical Turk). This results in an overall cost of $84.72 paid directly to the workers plus additional platform overhead fees. In order to enforce a higher worker churn and discouraging a single worker solving all challenges alone (and therefore also limiting the effects a malicious worker can have on the results), we restricted each worker to solving only 60 challenges (12 HITS). In accordance to the above observations that generally analogies are consensual, no Gold questions are used.

This resulted in 99 workers participating in this study. In average, each worker solved ~29 SAT challenges, i.e. most workers decided not continuing to work on our task after slightly less than 6 HITs in average. This can be attributed to the fact that in comparison to many common crowd-sourcing tasks, solving analogies is rather difficult and requires some deeper though in order to solve the tasks reliably. When observing the correctness of the answers of each individual worker (which can be measured due to the special nature of the SAT dataset), then the average correctness of all judgments is 56% - a number closely matching the reported performance of college applicants performing the SAT test for real. There is also no significant correlation between performance and number of judgments (Pearson correlation of 0.18 towards that workers providing many judgments have lower accuracy). This means that in general, workers tried to solve the tasks and did not simply and blatantly cheat by providing 60 random judgments. A visualization of general worker behavior is given in Figure 1.

Next, we observe the individual judgments themselves, especially with respect to the ambiguousness or unequivocalness of the crowd workers. As it turns out, workers are rarely in agreement on which answer choice is the correct one. This is shown in Figure 3. Out of 5 possible answers choices, most challenges (>67%) received 3 or 4 different answers overall by just 8 crowd-workers; for 9% of all challenges even every single answer choice was selected at least once. Only 3.9% off all challenges are unequivocal, and one of those is even unequivocally wrong. Just 19.5% of all challenges received rather focused answers with just two different result choices.

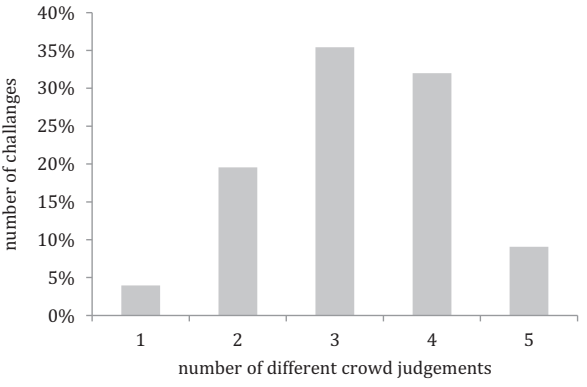


Figure 2. Different answer choices (out of 5) provided by 8 crowd workers for all SAT challenges

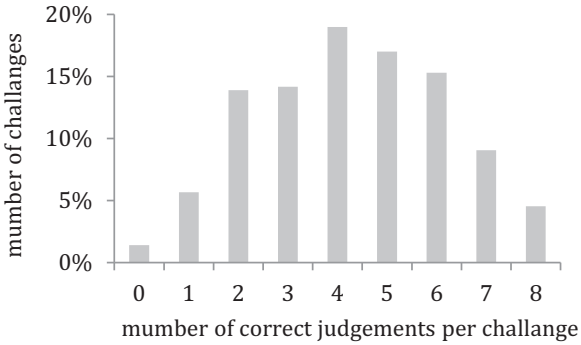


Figure 3. Analogy challenges: out of 8 judgments per challenge, how many judgments are correct? (353 challenges)

As each SAT challenge does indeed have a single carefully engineered correct answer, we can in the following focus on how reliably the workers choose this correct choice. In average only 4.2 workers out of 8 judge vote for the correct answer for each analogy challenge. A full overview of correct judgements is shown in Figure 3. Only very few challenges have an unequivocal and correct result (3.6%) while most challenges can only obtain 4 correct votes. Therefore, the base performance of our crowd-workers is rather low showing many disagreements between the workers themselves. Still, when simply performing a majority vote, this leads to an overall accuracy of 69%, which is clearly better than the performance of a single worker but not satisfying from a general perspective

3.2 Improving Quality and Capturing the Crowd Consensus

In this section, we will introduce heuristics modifying the basic majority vote in such a way that the overall result quality achieved by combining the feedback of the crowd workers further increases. Here, we aim at minimizing the impact of malicious and or non-competent workers while giving more power to workers which have proven themselves trustworthy and valuable. In contrast to the probabilistic approaches as e.g. in [Li12, Wh09], which rely on strict truth values for calibrating the heuristics, our heuristic aims at capturing the overall worker *consensus* and are therefore better suited for tasks like solving analogy problems or establishing similarity.

Our basic technique for deriving the best answer to an analogy challenge is a weighted majority vote, with the weight representing the credibility and reputation of a given worker. We will visit different definitions for the reputation weighting factor r_i later in this section. The generic definition for our basic voting heuristic is then given by:

Definition 1 (Weighted Majority Vote): Given a multiple-choice crowd sourcing task t with possible choices $C_t = \{c_{t,1}, \dots, c_{t,N_c}\}$, a set of workers $W = \{w_1, \dots, w_{N_W}\}$, and a set of corresponding worker judgements $J \subset (W \times C_t)$. Then the aggregated judgement c_v can be derived by:

$$v = \max_{j=1}^{N_C} \sum_{i=1}^{N_W} r_i \delta_{ij}$$

with r_i being the reputation of worker w_i and

$$\delta_{ij} = \begin{cases} 1 & \text{if } w_i \text{ votes for choice } c_{t,j}; i.e. \;\; if \; (w_i, c_{t,j}) \in J \\ 0 & \text{otherwise} \end{cases}$$

3.2.1 Agreement-based Reputation Weighting

In this first version of our reputation-weighting heuristic, worker reputation is based on their overall agreement with the predominant opinion of the work force. The rationale behind this technique is that while the predominant opinion (which can be derived by a simple majority vote) might be adulterated by malicious or incompetent workers, it is still a good indicator hinting at the best solution. Therefore, any worker who regularly agrees with the community opinion can be considered more valuable than a worker who is frequently disagreeing (and is most likely malicious, incompetent, or just has exotic and therefore generally unhelpful opinions). When combined with definition 1, this results in a 2-stage algorithm in which first all judgments are aggregated with simple majority votes in order to derive the worker reputation, and then are again aggregated using weighted majority votes incorporation said reputation to compute the final crowd judgment.

These considerations lead to following definition of worker reputation:

Definition 2 (Agreement-based Reputation): Given a set of crowd-sourcing tasks $T = \{t_1, \dots, t_{N_T}\}$ and according choices C_t and workers W as in definition 1, and worker judgements $J \subset \left(W \times \bigcup_{t \in T} C_t\right)$. Then, agreement-based reputation can be computed as follows:

$$r_i = \left(\frac{1}{N_T}\right) \sum_{t \in T} \alpha_{t,i}$$

with

$$\alpha_{t,i} = \begin{cases} 1 & \text{if } \left(w_{i,}\, c_{t,v_{majority}}\right) \in J \\ 0 & \text{otherwise} \end{cases}$$

whereas $c_{t,v_{majority}}$ is the result of a simple majority vote of all judgements of task t

3.2.2 PageRank-based Reputation Weighting

During this study, we also tried to apply established reputation management algorithms from other domains. Especially, the challenge of reputation management for crowd-sourcing can also be modeled as a graph: here, each worker is a node, and every time one worker decides for the same solution choice as a given second worker for a crowd-sourcing task, this can be seen as a positive vote of the first worker towards the second one (as each worker believes his choice is correct, he also vouches for other workers sharing his beliefs). Each of these votes then results in a directed edge in the graph. This setup allows

us to employ common link analysis algorithms as for example PageRank [BP98] to determine the importance and reputation coefficient r_i for all workers .

Definition 3 (PageRank-based Reputation): Given tasks T, according choices C_t, workers W, and respective judgements J as in definition 2, then an agreement reputation graph can be defined as follows:

Let RG be a directed graph with $RG = (W, E)$ with E being the set of agreement edges given by:

$$E = \left\{ (w_i, w_j) \in W \times W \mid \exists c_{t,x} \in J : \left((w_i, c_{t,x}) \in J \wedge (w_j, c_{t,x}) \in J \right) \right.$$

The final reputation weights for all workers W can be obtained by applying the PageRank algorithm on RG.

3.2.3 Hard Reputation Limits

The previous two techniques for capturing worker reputation can be further modified to exclude all workers whose reputation is below a certain threshold. When the threshold is well chosen, this tightly limits the impact of malicious workers. The effect of different thresholds is examined in the next section.

3.3 Evaluation

In the following, we will evaluate our aggregation heuristics on the judgments obtained in the study presented in section 3.1, and compare the results to currently established computer-based techniques for solving SAT challenges. This comparison allows for a rough assessment of the potential of different approaches and is summarized in Table 1 (page 12).

Regarding human performance, unanimous votes are close to meaningless with correctness of just 3.6%. Average worker performance is at 56%, with 69% being the accuracy achieved by simple majority votes. Our proposed weighted majority votes significantly improve this initial result quality. By using PageRank-based reputation weights, the performance is pushed to 74.8% and can be further improved to 75.4% by excluding the workers whose reputation is below 40% of the maximal reputation. However, the simpler agreement-based heuristics fares significantly better with a baseline of 76.2% and an accuracy of 81.5% when excluding workers with a reputation below 50% of the maximal reputation.

However, finding a suitable limit for excluding low-reputation workers is non-trivial as the optimal cut-off thresholds varies with the chosen heuristic, dataset, and worker pool. The effects of different thresholds are visualized in Figure 4, and show clearly that choosing the threshold too low or too high will yield suboptimal results. Furthermore, it also shows that limiting the majority vote to include only high reputation workers does not necessarily lead to a better result. This is due to the fact

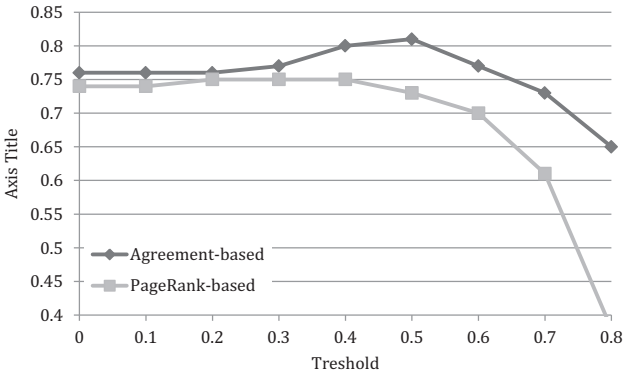


Figure 4. SAT Correctness with increasing reputation thresholds

that reputation for both heuristics is heavily based on agreement among workers, and captures real worker performance only approximately (please note that even though it might be intriguing to design an aggregation heuristics for analogies which is self-tuning by analyzing which workers frequently answer correctly, this approach is not suitable in general for non-SAT analogies or similar problems as usually there are no 'correct' answer which could serve as a Gold standard).

4 Summary and Outlook

In this paper, we examined crowd-sourcing techniques for approaching the challenge of analogy processing and relational similarity. Due to the consensual and ambiguous nature of these tasks, worker's average performance is rather low, rendering traditional quality control techniques like simple majority votes or Gold questions less effective. We approached this issue by weighted majority votes which balance each user judgment with the respective user's reputation. For capturing the notion of reputation, we showcased two heuristics both based on inter-worker agreement. By using these techniques, the measured result quality in the study we performed on the well-known SAT analogy dataset could be raised from 69% for simple majority votes to 81.5%.

In future works, one central challenge will be to further minimize the costs for obtaining crowd judgments. Especially, our approach can be modified in such a way that it respects the ambiguity of a given challenge. Therefore, for each challenge an optimal number of crowd judgments can be elicited (more judgments for ambiguous challenges, less judgments for seemingly easier challenges with more unequivocal results). Finally, this research should lead to dynamically combining the outputs of available machine-based techniques with focused crowd-sourcing in order to build true hybrid systems. Most of the system should rely on automatic algorithms, but in case of doubt, crowd-sourcing can be used to provide additional input.

Algorithm / Approach	Score (%)
Random guessing	20.0
Jiang and Conrath [Tu06]	27.3
Lin [Tu06]	27.3
Leacock and Chodrow [Tu06]	31.3
Hirst and St.-Onge [Tu06]	32.1
Resnik [Tu06]	33.2
PMI-IR [Tu06]	35.0
SVM [Bo08]	40.1
LSA + Predictation [Bo12]	42.0
WordNet [Ve04]	43.0
Bicici and Yuret [BY06]	44.0
VSM [TL05]	47.1
Combined [Bo12]	49.2
Pair-Classifier [Tu08]	52.1
RELSIM [Bo09]	51.1
Pertinence [Tu06-2]	53.5
LRA [Tu06]	56.1
High-school Students (historic SAT data)	57.0
Average Mechanical Turk Worker	56.0
MTurk Unanimous Vote (8 Jugements)	3.6
MTurk Majority Vote (8 Jugements)	69.0
MTurk Weighted Vote (8 Jugements, PageRank)	74.8
MTurk Weighted Vote (8 Jugements, PageRank, Limit 0.4)	75.4
MTurk Weighted Vote (8 Jugements, Agreement)	76.2
MTurk Weighted Vote (8 Jugements, Agreement, Limit 0.5)	81.5

Table 1. Reported correctness scores for computer-based and human-based approaches for solving SAT challenges

Funding Acknowledgements

This work was supported by funds provided by the DAAD FIT program (http://www.daad.de/fit/).

References

[Bo08] Bollegala, D. et al.: Www sits the sat: Measuring relational similarity on the web. Europ. Conf. on Artificial Intelligence (ECAI), Patras, Greece, 2008.

[Bo09] Bollegala, D.T. et al.: Measuring the similarity between implicit semantic relations from the web. 18th Int. Conf. on World Wide Web (WWW), Madrid, Spain, 2009.

[Bo12] Bollegala, D. et al.: Improving Relational Similarity Measurement using Symmetries in Proportional Word Analogies, Information Processing & Management, 2012.

[BP98] Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems. 30, 1-7, 107–117, 1998.

[Br89] Brown, W.R.: Two Traditions of Analogy. Informal Logic. 11, 3, 160–172, 1989.

[BY06] Bicici, E., Yuret, D.: Clustering word pairs to answer analogy questions. TAINN, 2006.

[Da08] Davidov, D.: Unsupervised Discovery of Generic Relationships Using Pattern Clusters and its Evaluation by Automatically Generated SAT Analogy Questions. Ass. for Computational Linguistics: Human Language Technologies (ACL:HLT). , Columbus, Ohio, USA, 2008.

[FB10] Ferrucci, David; Brown, Eric; Chu-Carroll, J. et al.: Building Watson: An Overview of the DeepQA Project. AI Magazine. 31, 3, 59–79, 2010.

[Ge83] Gentner, D.: Structure-mapping: A theoretical framework for analogy. Cognitive science. 7, 155–170, 1983.

[Ge97] Gentner, D., Markman, A.B.: Structure mapping in analogy and similarity. American Psychologist. 52, 45–56, 1997.

[Ge03] Gentner, D.: Why We're So Smart. Language in Mind: Advances in the Study of Language and Thought. pp. 195–235 MIT Press, 2003.

[Ho01] Hofstadter, D.R.: Analogy as the Core of Cognition. The Analogical Mind. pp. 499–538, 2001.

[Le10] Lee, J. et al.: Product EntityCube: A Recommendation and Navigation System for Product Search. 26th IEEE International Conference on Data Engineering (ICDE, Long Beach, California, USA, 2010.

[Li12] Lin, C.H. et al.: Crowdsourcing Control: Moving Beyond Multiple Choice. Human Computation Workshops at AAAI Conference on Artificial Intelligence, Ontario, Canada, 2012.

[Lo10] Lofi, C. et al.: Mobile Product Browsing Using Bayesian Retrieval. 12th IEEE Conference on Commerce and Enterprise Computing (CEC). , Shanghai, China, 2010.

[Lo12] Lofi, C. et al.: Information Extraction Meets Crowdsourcing: A Promising Couple. Datenbank-Spektrum. 12, 1, 2012.

[Se12] Selke, J. et al.: Pushing the Boundaries of Crowd-Enabled Databases with Query-Driven Schema Expansion. 38th Int. Conf. on Very Large Data Bases (VLDB). pp. 538–549 in PVLDB 5(2), Istanbul, Turkey, 2012.

[TL05] Turney, P., Littman, M.: Corpus-based learning of analogies and semantic relations. Machine Learning. 60, 251–278, 2005.

[Tu06] Turney, P.: Expressing Implicit Semantic Relations without Supervision. Int. Conf. on Computational Linguistics (COLING). , Sydney, Australia, 2006.

[Tu06-2] Turney, P.D.: Similarity of semantic relations. Computational Linguistics. 32, 3, 379–416, 2006.
[Tu08] Turney, P.D.: A uniform approach to analogies, synonyms, antonyms, and associations. Int. Conf. on Computational Linguistics (COLING). , Manchester, UK, 2008.
[Ve04] Veale, T.: Wordnet sits the sat: a knowledge-based approach to lexical analogy. Europ. Conf. on Artificial Intelligence (ECAI), Valencia, Spain, 2004.
[Wh09] Whitehill, J. et al.: Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise, Advances in Neural Information Processing Systems. pp. 2035–2043, Vancouver, Canada, 2009.

Studierendenprogramm

Im Rahmen der BTW 2013 in Magdeburg wird wie bei vorangegangenen BTW-Tagungen ein Studierendenprogramm veranstaltet. Es bietet Studierende die Möglichkeit, ihre Arbeiten im Umfeld von Datenbanken und Datenbanksystemen einem Fachpublikum zu präsentieren und sich mit Fachleuten und Gleichgesinnten auszutauschen.

Program Chair

Prof. Dr. Thomas Neumann, Technische Universität München

The story of instant recovery – Extended abstract –

Goetz Graefe
Hewlett-Packard Laboratories

This presentation will summarize the history and the technology of instant recovery from system and media failures. The story starts with modern hardware, e.g., flash storage, and the danger of localized failures due to limited write endurance. Initial research sought methods for detection and recovery of localized, i.e., single-page failures.

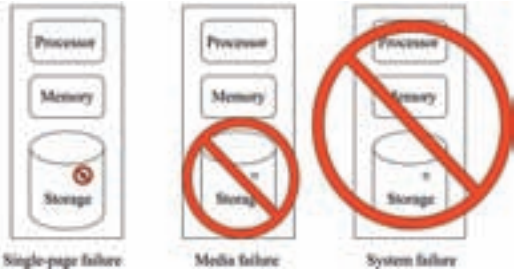


Figure 1: Single-page failures. [GK12]

Figure 1 distinguishes multiple failure classes. If single-page recovery is available, a storage device remains useful even after a localized failure. Otherwise, a single-page failure may imply a device failure, which in turn may imply a node failure - clearly very undesirable if it is avoidable.

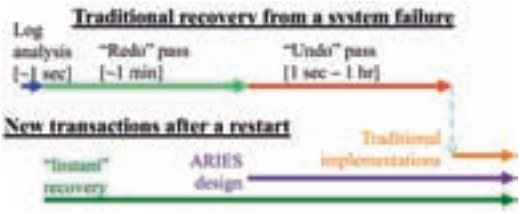


Figure 2: Instant recovery from system failures.

Figure 2 illustrates an application of single-page recovery, specifically incremental recovery from a system failure, e.g., a process crash. Traditional implementations permit new database transactions only after all recovery activities are complete, which may take

minutes or even hours. Instant recovery after a system failure requires only log analysis. Differently than advanced techniques by Mohan and by Speer and Kirchberg, which give early access only to pages not modified for a long time, instant restart permits immediate access to the pre-crash working set. Differently than all previous recovery techniques, new transactions guide the recovery, not scans of the recovery log. Figure 4 illustrates what instant recovery means for media failures. While traditional techniques enable new database transactions only after hours of recovery efforts using database backups and log archives, instant media recovery enables new database transactions almost immediately, i.e., within seconds of an empty replacement device becoming available. The core technique is on-demand incremental recovery enabled by novel log archiving. Due to a single pass over the recovery log, log archiving remains efficient.

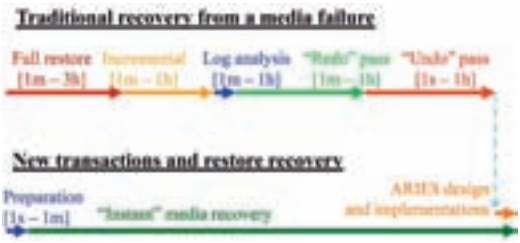


Figure 3: Instant recovery from media failures.

Figure 4 shows how to extend these ideas to index operations without detailed logging, i.e., with allocation-only logging. Instead of detailed ”redo” information in the recovery log, controlled retention of intermediate files enables efficient single-page recovery from failures.

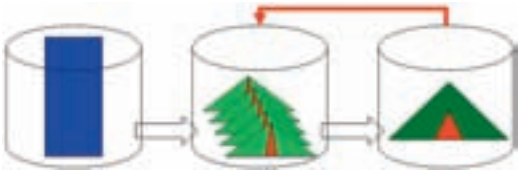


Figure 4: Instant recovery from media failures. [GG13]

The first grand prize for this research will be individual nodes with greatly improved mean time to repair and thus greatly improved availability. The second grand prize will be instant failover to a node that has an out-of-date copy of the database plus log archive and recent recovery log. The third grand prize will be a data center design in which high availability for N working nodes is possible with N+3 nodes rather than with today’s 3N nodes. In other words, we hope to reduce costs for machines, cabling, power, cooling, space, etc. to about a third of today’s standard data center design.

(Figures copied from the author’s cited papers.)

References

[CM92] B. G. Lindsay H. Pirahesh P. M. Schwarz C. Mohan, D. J. Haderle. ARIES, a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. In *ACM TODS 17(1)*, 1992.

[GG13] B. Seeger G. Graefe. Logical recovery from single-page failures. In *BTW*, 2013.

[GK12] Goetz Graefe and Harumi A. Kuno. Definition, Detection, and Recovery of Single-Page Failures, a Fourth Class of Database Failures. *PVLDB*, 5(7):646–655, 2012.

[Gra12] G. Graefe. Instant recovery from system failures. Unpublished manuscript, 2012.

[Gra13] G. Graefe. Instant recovery from media failures. In preparation, 2013.

[JS07] M. Kirchberg J. Speer. C-ARIES, A multi-threaded version of the ARIES recovery algorithm. In *DEXA*, 2007.

[Moh93] C. Mohan. A cost-effective method for providing improved data availability during DBMS restart recovery after a failure. In *VLDB*, 1993.

HyDash: A Dashboard for Real-Time Business Intelligence based on the HyPer Main Memory Database System

Maren Steinkamp Tobias Mühlbauer*

Lehrstuhl für Datenbanksysteme
Technische Universität München · Boltzmannstraße 3 · D-85748 Garching
{steinkam, muehlbau}@in.tum.de

Abstract: Business Intelligence (BI) is a set of techniques that help improve business decision making. From a technical point of view, BI relies on a set of tools which includes performance dashboards: layered services that combine monitoring, analysis, and reporting. However, most dashboard solutions today are based on data warehouses which face a problem of data staleness—a circumstance caused by the separation of transactional and analytical processing. Novel hybrid main memory database systems such as SAP's HANA or HyPer overcome this issue and achieve best-of-breed transaction processing throughput and analytical query response times in one system in parallel on the same database state. In this work, we contribute HyDash, a web-based dashboard for real-time business intelligence. HyDash visualizes key performance indicators (KPIs) based on predefined SQL queries and provides an ad-hoc SQL query interface to dynamically extend the scope of the dashboard. HyDash relies on (i) hybrid main memory database systems and (ii) a push-based query result update mechanism based on the novel HTML5 WebSockets.

1 Motivation

Business Intelligence (BI) systems are data-driven decision support systems, i.e., concepts and methods that improve business decision making by using fact-based support systems [Pow03], which, e.g., analyze an enterprise's transactional data by means of data analysis. Eckerson [Eck10] names an overview of BI tools: reports, performance dashboards, online analytical processing (OLAP), ad-hoc querying, visual analysis, search, as well as statistical models. In his view, performance dashboards represent "the latest incarnation of BI" and provide "a layered information service that combines monitoring, analysis, and reporting" whereby meeting most of the information needs of casual BI users. Commercial dashboard solutions are developed by established companies such as SAP Business Objects or IBM DB2 Alphablox. The advent of big data and an increased need for complex analysis and data visualization recently triggered the launch of several startups such as Platfora (`http://www.platfora.com/`) or Chartio (`http://www.chartio.com`) who now also develop BI systems and dashboard solutions. These dashboards display key performance indicators (KPIs) to analysts in a way that is easy to interpret [PCMP07].

*Tobias Mühlbauer is a recipient of the Google Europe Fellowship in Structured Data Analysis, and this research is supported in part by this Google Fellowship. This work has further been partially sponsored by the German Federal Ministry of Education and Research (BMBF) grant HDBC 01IS12026.

However, the selection of these KPIs is often static and the KPIs are, as most dashboards are based on an underlying data warehouse solution, rarely updated. This issue is often referred to as the problem of *data staleness* which is due to the common separation of transactional database systems and data warehouses. Long-running analytical queries (OLAP), e.g., those determining the KPIs, are running in the data warehouse so to not interfere with the mission-critical online transactional processing (OLTP) in the transactional database. The data warehouse is thereby periodically updated with a fresh transactional view by a so-called extract-transform-load (ETL) phase. As the ETL phase itself could again interfere with OLTP processing, it is often only carried out during times of reduced OLTP load. During busy hours, the data warehouse is not updated and is thus inevitably facing the problem of data staleness. Industry leaders such as Hasso Plattner of SAP however argue that the issue of data staleness is inappropriate for *real-time business analytics*[1] [PZ11]. Recent developments in main memory database systems address this problem: hybrid OLTP&OLAP database systems such as SAP's HANA or HyPer [KN11] achieve best-of-breed transaction processing throughput and OLAP query response times in one system in parallel on the same database state. Being able to access KPIs based on the newest available transactional data generates an enormous benefit for decision makers in industry and science.

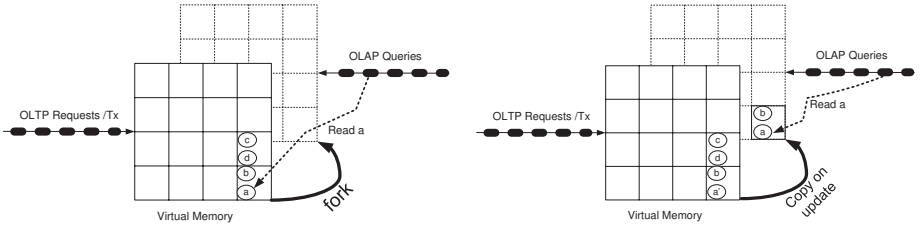
In this work we contribute HyDash, a web-based dashboard for real-time business intelligence. It is the goal of HyDash to visualize key performance indicators (KPIs) based on predefined SQL queries and to further provide an ad-hoc SQL query interface to dynamically extend the scope of the dashboard. Furthermore, updates in query results triggered by changes in transactional data should be propagated to the dashboard in real-time. To achieve this goal, HyDash relies on (i) hybrid main memory database systems, which efficiently separate query processing from mission-critical transactional processing and provide low response times and (ii) a push-based query result update mechanism based on the novel HTML5 WebSockets [Hic12b], which provide a full-duplex communications channel between a web application and its backend. In order to achieve a performant web frontend, several technologies were combined. On the backend side, the HyDash implementation is based on the hybrid main memory database system HyPer and its scale-out ScyPer [MRR+13]. However, any other hybrid main memory database system could have been used. Finally, we evaluate the dashboard by conducting a case study based on the CH-BenCHmark [C+11], a combination of the TPC-C and TPC-H benchmarks.

The remainder of this paper is structured as follows: Section 2 introduces the HyPer hybrid OLTP&OLAP main memory database system. Section 3 outlines the architecture and evaluation of HyDash. Section 4 then provides an overview of work related to HyDash. We conclude by summarizing our results and sketching further work.

2 The HyPer Hybrid OLTP&OLAP Main Memory Database System

HyPer [KN11] is a hybrid OLTP&OLAP main memory database system that achieves best-of-breed transaction processing throughput and OLAP query response times in one

[1] We use real-time in the sense of human real-time, i.e., response times that matter in the human decision making process.



(a) Forking a new snapshot (b) Copy on update to preserve consistency

Figure 1: HyPer's snapshotting mechanism

system in parallel on the same database state. OLAP query processing is separated from the mission-critical OLTP processing — for which the ACID properties are guaranteed — through a virtual memory (VM) snapshotting mechanism. On modern commodity hardware HyPer achieves a throughput exceeding 100,000 TPC-C transactions per second while providing query response times comparable to market-leading OLAP databases.

HyPer's performance is due to the following key characteristics:

1. HyPer relies on in-memory data management and eliminates the ballast caused by DBMS-controlled page structures and buffer management in traditional database systems.

2. OLAP processing is separated from mission-critical OLTP processing by forking transaction (TX)-consistent VM snapshots using the POSIX system call `fork()` (Figure 1(a)). The hardware- and operating system-supported "copy on update" mechanism then preserves the consistency of the snapshot by copying pages only if a page is modified (Figure 1(b)). The creation of each snapshot is scheduled as a system-internal transaction and as such no concurrency control mechanism is needed to separate OLAP and OLTP processing.

3. Transactions and queries are specified in SQL or, alternatively, in the HyPer scripting language which treats SQL as a first class citizen. HyPer essentially acts as a compiler that efficiently compiles these transactions and queries into optimized machine-independent LLVM assembler code [Neu11].

4. Parallel transactions are separated via lock-free admission control. Non-conflicting transactions can thus be processed simultaneously in multiple threads without the need for concurrency control.

5. HyPer uses logical redo logging where the invocation parameters of transaction procedures are logged via a high-speed network.

3 HyDash: A Dashboard for Real-Time Business Intelligence

HyDash is a web-based dashboard for real-time business intelligence which visualizes key performance indicators (KPIs) based on predefined SQL queries and provides an ad-hoc SQL query interface to dynamically extend the scope of the dashboard. Updates in

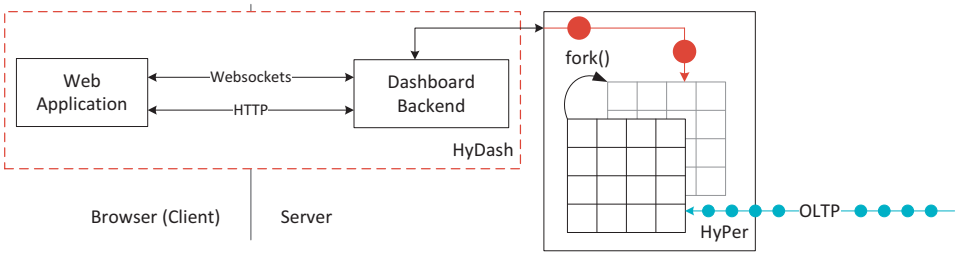


Figure 2: HyDash architecture

query results triggered by changes in transactional data are propagated to the dashboard in real-time through HTML5 WebSockets [Hic12b]. On the backend side, the HyDash implementation is based on the hybrid main memory database system HyPer and its scale-out ScyPer [MRR+13]. However, any other hybrid main memory database system could have been used.

In the following, we first present HyDash's client-server architecture which consists of a web application and a backend application that is directly communicating with the underlying hybrid main memory database system. Then, we evaluate HyDash by conducting a case study based on the CH-BenCHmark [C+11], a combination of the TPC-C and TPC-H benchmarks. We thereby use the 22 analytical TPC-CH queries of the benchmark as the predefined KPIs of the dashboard.

3.1 Architecture

HyDash is implemented using a client-server architecture approach. It consists of a web application and a server-side dashboard backend component. The backend component thereby connects the HyPer main memory database system with the web application. The web application and dashboard backend component communicate through HTML5 WebSockets [Hic12b] and HTTP requests. HTML5 WebSockets provide a low-latency and low-cost full-duplex communication channel between the web application and the dashboard backend component. We use HTML5 WebSockets to implement a scalable push-based approach to propagate changes in query results triggered by updates in transactional data in real-time. HTTP is solely used to deliver the web application's initial page which contains the JavaScript application code to communicate through WebSockets. All other communication related to the visualization of KPIs and ad-hoc query results uses WebSockets. The architecture of HyDash is shown in Figure 2.

Web application. The HyDash web application is designed using the HTML5 markup language to structure the document, CSS3 for the design, and JavaScript to define the application behavior. For each predefined KPI, the dashboard provides a separate view. Each of these views visualizes the current query results in a columnwise sortable tabular view and additionally shows the last execution time of the query as well as statistics such as the amount of returned tuples. If applicable, the result is further visualized as a chart. For the charts and tabular view, we use the Google Chart Tools (`https://developers.`

google.com/chart/). Charts can, e.g., be pie, line, bar, or world map charts. For predefined KPIs, the types of charts that can be used are statically defined. In the future we also plan to implement a chart recommendation engine that proposes suitable chart types based on the returned result set. Additionally, the web application provides an ad-hoc query interface where users can enter arbitrary SQL queries. These queries then define a new KPI and the dashboard visualizes the query results and its updates in real time. Currently only a tabular view is used to display the result. With the aforementioned chart recommendation engine, we plan integrate charts in the ad-hoc query interface.

The web application is communicating with the dashboard backend through HTML5 WebSockets. This allows a push-based communication where the dashboard backend pushes updated query results to connected web applications. The push interval for each query is based on the query's average execution time. On the web application side, we use the Google Closure (https://developers.google.com/closure/) implementation of WebSockets; on the dashboard backend side we use a C++ implementation of the WebSockets protocol. Whenever the user switches the page to see another query's result, the web application checks whether it already cached the query results for that query and loads the data table. Additionally, it requests the most recent query result from the backend and therewith subscribes for updates for this specific query. As soon as an update arrives on the WebSocket, the JSON serialization is parsed and the tabular view's and charts' data are updated automatically. Additionally, in order to reduce the tabular views' rendering time, the table is paged with a page size of 20. This clear arrangement also makes it easier for the user to browse the data. When new data for the table arrives, the table remains on the currently displayed page.

Dashboard backend. The web clients communicate with the dashboard backend through WebSockets and HTTP requests. The dashboard backend is the central instance that handles all open connections, coordinates the execution of the queries with the underlying HyPer main memory database system, and broadcasts updated query results to all connected web clients that subscribed for that query. The dashboard backend is implemented in C++ and is based on a lightweight webserver which provides access to the web application through HTTP. It also implements a WebSocket server using the WebSocket++ (http://www.zaphoyd.com/websocketpp) implementation which is used to broadcast updated query results. The communication between the dashboard backend and the database system is pull-based in order to allow portability to other database systems. In the following we describe the processes in the dashboard backend in more detail.

Preparation of the query execution. On startup, the dashboard backend prepares the execution of the predefined KPIs. Therefore it reads the queries for the KPIs from files and sends them to the HyPer database system. Once HyPer confirms that a query is prepared, the dashboard backend saves the identifier of the prepared query as well as the schema information for the query result as JSON strings. Once all confirmations have been received, the continuous execution of the predefined queries is started.

Executing the predefined queries. The dashboard backend requests the execution for each of the prepared queries. This step is continuously repeated after query results are received

	Mean time (standard deviation) [ms]			
Query	17	12	2	10
Result set size [tuples]	1	12	1425	58398
Parsing	0 (0)	0.3 (0.48)	21.5 (3.31)	256.6 (49.62)
Initial build of table	3.7 (1.95)	3 (1.41)	1.7 (0.67)	4.5 (2.17)
Initial build of chart	2.4 (1.96)	2.2 (2.25)	-	-
Table update	3.9 (1.37)	8.2 (3.55)	30.2 (8.89)	359.2 (60.59)
Chart update	14.8 (4.85)	36.8 (16.37)	-	-
Total	19.9 (5.93)	40.2 (17.16)	53 (11.19)	634.8 (110.88)

Table 1: Parsing and rendering times of selected queries in HyDash.

and a request for a fresh snapshot is sent. HyPer only creates a new snapshot if the timestamp of the latest write transaction is greater than the the latest snapshot time, i.e., the current snapshot is outdated.

Postprocessing of query results. Whenever a query result arrives, the query result is transformed into a JSON string that can be sent directly to all subscribed web clients. Hereby the previously stored schema information including the type of each field allows to save numeric data types as numbers in the JSON string. This step is necessary to enable the direct loading of data into the Google Data Table that serves as data basis for the displayed tabular view (Google Table) and Chart. The JSON string is cached together with the query identifier. This allows an instant response to requests for specific query results coming from new web clients.

Handling requests from web-clients and broadcasting new query results. The dashboard backend communicates with the web-clients via a WebSocket. New connections on the WebSocket are handled by worker threads. If a request for a predefined query is received, the cached query result is immediately forwarded to the client. If the request contains an ad-hoc query request, the query is sent to the database system and will from now on be executed continuously until the client disconnects, i.e., the new ad-hoc query is treated such as a predefined KPI and the aforementioned steps are also processed for the new KPI. In both cases, for predefined KPIs and ad-hoc queries, the backend stores a mapping between clients and the queries these clients are interested in. This mapping is used to multicast updated query results only to the clients that subscribed for the query.

3.2 Evaluation: The CH-BenCHmark Case Study

To evaluate the performance of HyDash, a case study based on the CH-BenCHmark, a combination of the TPC-C and TPC-H benchmarks, has been conducted. The 22 TPC-H queries are treated as predefined KPIs. A screenshot of the predefined KPI visualization is shown in Figure 3. The ad-hoc query interfaces can be seen in Figure 4.

We conducted an evaluation of the dashboard on an AMD Bulldozer (8 cores, 32 GB main memory) commodity machine. Thereby the HyPer main memory database system, the dashboard backend, and the web browser were running on the same machine. To simulate transactional load, 50,000 transactions per second were generated and sent to HyPer. The queries from the dashboard backend were repeatedly processed in parallel on

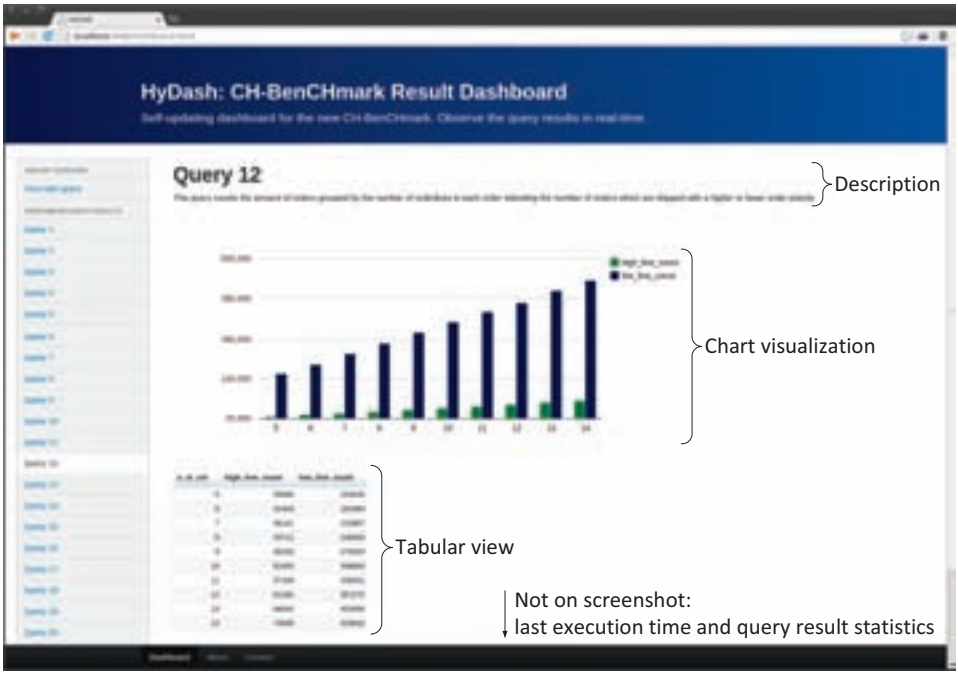


Figure 3: HyDash web frontend: predefined KPI visualization.

fresh snapshots. The majority of the query response times lay in the area of subseconds. We could not identify any performance issues related to the dashboard backend and the underlying HyPer database system. However, some performance bottlenecks could be revealed in the web application. The parsing of query results serialized as JSON strings is expensive and can, for large query results, take longer than the average query response time interval that produces the result. We experienced that total parsing and rendering times below 350 ms are not perceived by the user, whereas longer times, as, e.g., for Query 10 may freeze the webbrowser. One improvement is to parallelize the parsing using HTML5 Web Workers [Hic12a]. This allows the execution of parallel tasks in the web application without blocking the user interface. Displaying the query results in data tables or charts requires table and charting components that support a fluent data update without a high rendering time. The Google Table and Chart used in HyDash works well on data updates for a reasonable amount of data and provide good possibilities to show the transition of data in charts. Our evaluation showed that tables with up to 5,000 tuples allow subsecond update times, whereas bigger results can lead to increased rendering times that can exceed the average query response time that produces the result. The rendering time has been reduced by applying client-side paging for the data table and can be further optimized by enabling server-side paging of the query results. With these optimizations it is possible to efficiently visualize KPIs and query results in real-time. However, in cases where the parsing and rendering time exceeds the query's execution time, the time interval for the execution of the query must be adapted to avoid excessive demands towards the web application. Table 1 displays the parsing and rendering times of four selected queries.

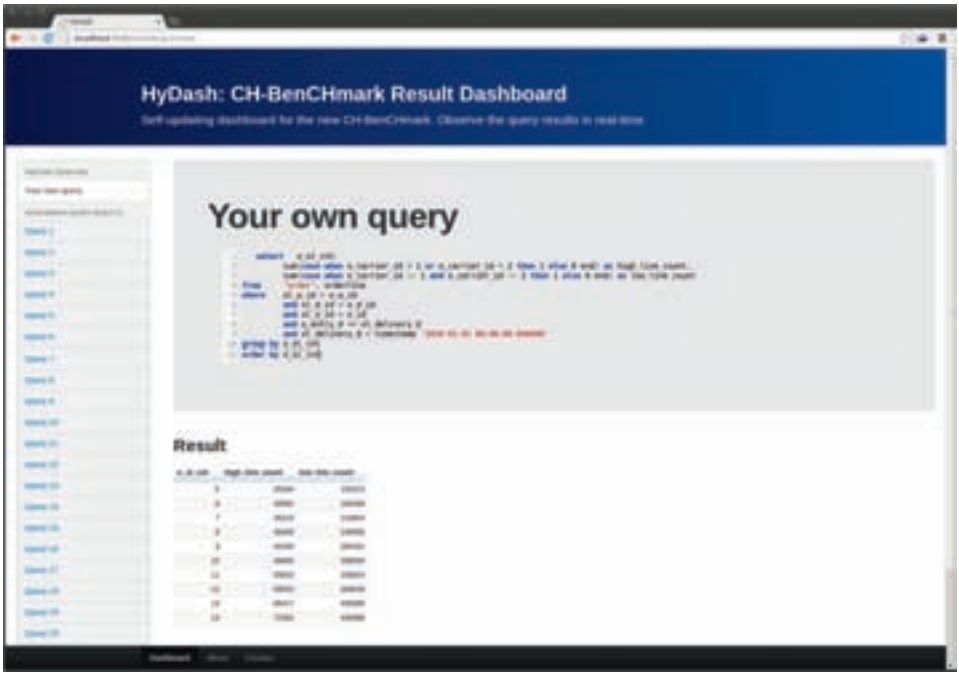


Figure 4: HyDash web frontend: ad-hoc query interface.

The evaluation is based on the non-optimized version of HyDash with only client-side paging enabled.

4 Related Work

BI is an umbrella term that, from a technical point of view, describes a variety of decision support tools [Eck10]. While, as described in Section 1, Eckerson believes that performance dashboards represent "the latest incarnation of BI", they often only provide an interface for various underlying technologies. HyDash is essentially a KPI visualization and ad-hoc OLAP query interface for hybrid main memory database systems which combine the functionality of a transactional database and a data warehouse. Besides relational data warehouses, BI tools can also be backed by, e.g., MapReduce-based systems [CLL+11, BEH+10], knowledge warehouses [NSIH02], or domain-specific applications.

Even though HyDash allows the user to dynamically extend the dashboard through ad-hoc queries, most KPIs are statically defined. Model-driven dashboard development approaches [PCMP07, PS07] aim at the automatic generation of dashboard implementations from business process models. In [PCMP07], Palpanas et al. introduce an approach to generate IBM Websphere-based dashboards from business process models. In contrast to model-generated dashboards, HyDash is a standalone solution that could however be integrated in a model-driven approach as a template.

In [CZ08], Chieu and Zeng identify the need for real-time data analysis and providing historical information for KPIs in performance monitoring dashboards. To achieve the goal

of real-time monitoring, they introduce a novel capture-transform-update (CTU) phase that replaces the ETL phase to update their data warehousing backend. HyDash is based on hybrid main memory database systems. These systems can efficiently provide real-time analysis capabilities. E.g., HyPer relies on relies on a virtual memory (VM) snapshotting mechanism that is efficiently supported by hardware and the operating system and does not face scalability issues that can occur when using CTU phases.

HyDash distinguishes itself from related web-based dashboards by providing human real-time analytics and push-based communication using HTML5 WebSockets. It can be dynamically extended by user-defined ad-hoc queries and can visualize complex OLAP query results with real-time update capabilities.

5 Outlook and Concluding Remarks

The current implementation of HyDash is only a starting point for a full-fledged real-time business intelligence (BI) dashboard. It can be extended in several directions. First, HyDash is currently based on the HyPer main memory database system, but can easily be integrated with other hybrid OLTP&OLAP database systems. Furthermore, there is huge potential to improve the dashboards's visualization capabilities, usability, and interactivity. Regarding data visualization, we plan to let not only users dynamically decide on the visualization format for a specific query result but rather also provide a recommendation system that, by using schema and query information, aids the user in choosing the right visualization format. HyDash's ad-hoc querying functionality provides a starting point to extend the dashboard with data exploration features by recommending new meaningful queries based on the user-defined queries as well as crowd-, statistical-, and OLTP as well as OLAP hotspot-knowledge. Especially the scientific community is asking for more efficient data exploration tools and methodologies [Abb09, KIML11]. Whereas the data acquisition problem is considered to be solved, meaningful data analysis and data exploration remains a big challenge. Data exploration aims at identifying relevant information within a huge amount of data [Abb09]. In [KIML11], Kersten et al. propose new approaches to handle data exploration more efficiently such as query morphing and one-minute kernels. We plan to extend the ad-hoc query interface of HyDash and the underlying HyPer main memory database system to implement these data exploration capabilities.

In this work we presented HyDash, a web-based dashboard for real-time business intelligence. HyDash visualizes key performance indicators (KPIs) based on predefined SQL queries and further provides an ad-hoc SQL query interface to dynamically extend the scope of the dashboard. To reflect transactional updates in KPIs and ad-hoc query results in real-time, HyDash relies on (i) hybrid main memory database systems, which efficiently separate query processing from mission-critical transactional processing and provide low response times and (ii) a push-based query result update mechanism based on the novel HTML5 WebSockets. On the backend side, the HyDash implementation presented in this work is based on the hybrid main memory database system HyPer — however, any other hybrid main memory database system could have been used. We evaluated the dashboard implementation by conducting a case study based on the CH-BenCHmark, a combination of the TPC-C and TPC-H benchmarks. The evaluation revealed that initially the visualization process in the web-based part of HyDash was not able to keep up with the rate query

result updates where propagated by the backend. This was due to the expensive parsing of large result sets and an increased rendering time of charting components for large result sets. For both issues solutions were proposed.

References

[Abb09] M. R. Abbott. A new path for science? In T. Hey, S. Tansley, and K. M. Tolle, editors, *The Fourth Paradigm*, pages 111–116. Microsoft Research, 2009.

[BEH+10] D. Battré, S. Ewen, F. Hueske, O. Kao, V. Markl, and D. Warneke. Nephele/PACTs: a programming model and execution framework. In *SOCC*, pages 119–130, 2010.

[C+11] R. Cole et al. The mixed workload CH-benCHmark. In *DBTest*, pages 8:1–8:6, 2011.

[CLL+11] B. Chattopadhyay, L. Lin, W. Liu, S. Mittal, P. Aragonda, V. Lychagina, Y. Kwon, and M. Wong. Tenzing: A SQL Implementation On The MapReduce Framework. In *PVLDB*, pages 1318–1327, 2011.

[CZ08] T. C. Chieu and L. Zeng. Real-time performance monitoring for an enterprise information management system. In *ICEBE*, pages 429–434, 2008.

[Eck10] W.W. Eckerson. *Performance dashboards: measuring, monitoring, and managing your business*. Wiley, 2010.

[Hic12a] I. Hickson. Web Workers. `http://dev.w3.org/html5/workers/`, Editor's Draft November 25, 2012.

[Hic12b] I. Hickson. The WebSocket API. `http://dev.w3.org/html5/websockets/`, Editor's Draft November 24, 2012.

[KIML11] M. L. Kersten, S. Idreos, S. Manegold, and E. Liarou. The Researcher's Guide to the Data Deluge: Querying a Scientific Database in Just a Few Seconds. *PVLDB*, 4(12):1474–1477, 2011.

[KN11] A. Kemper and T. Neumann. HyPer: A hybrid OLTP&OLAP main memory database system based on virtual memory snapshots. In *ICDE*, pages 195–206, 2011.

[MRR+13] T. Mühlbauer, W. Rödiger, A. Reiser, A. Kemper, and T. Neumann. ScyPer: A Hybrid OLTP&OLAP Distributed Main Memory Database System for Scalable Real-Time Analytics. In *BTW*, 2013.

[Neu11] T. Neumann. Efficiently compiling efficient query plans for modern hardware. *PVLDB*, 4(9):539–550, 2011.

[NSIH02] H. R. Nemati, D. M. Steiger, L. S. Iyer, and R. T. Herschel. Knowledge warehouse: an architectural integration of knowledge management, decision support, artificial intelligence and data warehousing. *Decision Support Systems*, 33(2):143–161, 2002.

[PCMP07] T. Palpanas, P. Chowdhary, G. Mihaila, and F. Pinel. Integrated model-driven dashboard development. *Information Systems Frontiers*, 9(2):195–208, 2007.

[Pow03] D.J. Power. A Brief History of Decision Support Systems. `http://DSSResources.COM/history/dsshistory.html`, version 2.8, May 31, 2003.

[PS07] D.J. Power and R. Sharda. Model-driven decision support systems: Concepts and research directions. *Decision Support Systems*, 43(3):1044–1061, 2007.

[PZ11] H. Plattner and A. Zeier. *In-Memory Data Management: An Inflection Point for Enterprise Applications*. Springer, 2011.

Optimierung der Exact-Match-Anfrage eines Lokal Sensitiven Hashverfahrens

Sarah Heckel
Fakultät für Mathematik
Otto-von-Guericke Universität
Universitätsplatz 2
39106 Magdeburg
sarah.heckel@st.ovgu.de

Abstract: Hochdimensionale Indexverfahren sind wichtig um einen schnellen Zugriff auf Multimediadaten zu gewährleisten. Eine Klasse dieser Verfahren ist das Lokal Sensitive Hashen (LSH). Beim LSH können sehr unterschiedlich ausgelastet Bereiche entstehen. Um die Exakt-Match-Anfrage beim Permutationsansatz, einer Variante des LSHs, effizient bearbeiten zu können, ist eine gleichmäßige Raumaufteilung von Vorteil. Dazu ist die Wahl der Protoypen von großer Bedeutung.
Im Folgenden wird ein mathematisches Optimierungsproblem aufgestellt, welches die Prototypen bestimmt. Die Idee dabei ist Kugeln mit minimalem gleichem Radius um die Prototypen zu legen, sodass jeder Datenpunkt in mindestens einer Kugel enthalten ist.
Werden optimierte Prototypen für die permutationsbasierte Variante des LSHs gewählt, so ist die Abweichung der Raumaufteilung gegenüber der Aufteilung bei zufällig gewählten Prototypen stabiler.

1 Einleitung

Hochdimensionale Indexstrukturen werden benötigt, um Daten mit vielen Attributen verwalten zu können. Dabei wird der Raum in mehrere kleinere Bereiche, im Folgenden Buckets genannt, aufgeteilt, damit Datenpunkte schneller gefunden werden können. Es gibt verschiedene Indexstrukturen um hochdimensionale Daten gut verwalten zu können. Dabei wird zwischen exakten und approximativen Indexverfahren unterschieden [AI08]. Exakte Verfahren geben bei Nachbarschaftsanfragen nach den k nächsten Nachbarn (kNN) genau die k nächsten Datenpunkte eines Datenpunktes aus, während die approximativen Verfahren zu einem Datenpunkt k ähnliche Elemente ausgeben. Zu den exakten Verfahren gehören zum Beispiel verschiedene Baumverfahren, wie der R-Baum und X-Baum [GG97]. Die Klasse der LSH-Verfahren gehört zu den approximierenden Verfahren [GG97]. Das LSH ist ein hochdimensionales Hashverfahren. Es wurde vorgestellt, um die kNN-Suche gegenüber anderen Hashverfahren zu verbessern [IM98]. Im Gegensatz zu anderen Hashverfahren, werden ähnliche Datenpunkte auf dasselbe Bucket abgebildet, statt sie über die Buckets zu streuen. Um die kNN eines Datenpunktes zu finden werden nur noch die angrenzenden Buckets durchsucht.

Ein erster Schritt für eine Verbesserung dieses Verfahrens stellt die Optimierung der Exact-Match-Anfrage dar. Die Erkenntnisse, die sich daraus gewinnen lassen, können später für die Anfrage der kNN genutzt werden.
Im Folgenden geht es um die permutationsbasierte Variante des LSHs. Diese wird in Kapitel 2 näher erläutert. Dabei wird der Raum anhand von einigen Datenpunkten, den sogenannten Prototypen, aufgeteilt. Untersucht man die dadurch entstehende Raumaufteilung, so ist festzustellen, dass sehr ungleichmäßige Aufteilungen entstehen können. So erhält man beispielsweise einige Bereiche, die im Vergleich zu anderen sehr viele Elemente enthalten. Um diese dann sequenziell zu durchsuchen, wird mehr Zeit benötigt, als wenn schwach belegte Buckets durchsucht werden.
Für die verschiedenen Anfragetypen Exact-Match-Anfrage, kNN-Suche und Bereichsanfrage sind unterschiedliche Raumaufteilungen wünschenswert. So soll bei der Exact-Match-Anfrage eine möglichst gleichmäßige Raumaufteilung erreicht werden, so dass in jedem Bucket annähernd die selbe Anzahl an Elementen enthalten ist. Es müssen dann im Schnitt die gleiche Anzahl an Elementen sequenziell durchsucht werden. Um eine gute Raumaufteilung zu erzielen, sollen Prototypen so gewählt werden, dass die Datenpunkte möglichst gleichmäßig auf die Buckets verteilt werden. Genau um dieses Problem geht es in dieser Arbeit.

2 Lokal Sensitives Hashen

Viele Hashverfahren streuen die Daten auf verschiedene Bereiche und zerstören so die Nachbarschaftsbeziehungen. Diese Verfahren sind daher nicht geeignet für die Suche nach den nächsten Nachbarn eines Datenpunktes. Bei der exakten Suche nach den kNN müssen daher alle Bereiche durchsucht werden. Im Gegensatz dazu bildet das LSH ähnliche Datenpunkte auf dasselbe Bucket ab.

2.1 Definition

Beim LSH gibt es statt nur einer Hashfunktion mehrere Hashtabellen, für die jeweils verschiedene Hashfunktionen gelten [IM98]. Dabei kommen die Hashfunktionen aus einer Familie $\mathcal{H}$ von Funktionen. Jede Funktion h aus $\mathcal{H}$ ist dabei (P1, P2, r, cr)-sensitiv [IM98], das heißt für je zwei Datenpunkte $p, q \in \mathcal{R}^d$ und $P1 > P2$ gilt:

1. $wenn \parallel p - q \parallel < r,\ dann\ Pr[h(p) = h(q)] > P1$
2. $wenn \parallel p - q \parallel > cr,\ dann\ Pr[h(p) = h(q)] < P2.$

Das heißt, wenn der Abstand von p und q kleiner als r ist werden p und q mit einer Wahrscheinlichkeit größer als $P1$ in dasselbe Bucket abgebildet. Dagegen werden p und q mit einer Wahrscheinlichkeit kleiner als $P2$ in dasselbe Bucket abgebildet, wenn ihr Abstand größer als cr ist. Diese Definition sorgt dafür, dass ähnliche Elemente mit hoher Wahrscheinlichkeit in dasselbe Bucket abgebildet werden.

2.2 Permutationsansatz

Chavez et al. stellen in [CFN08] den Permutationsansatz vor, der die Definition einer Funktionenfamilie $\mathcal{H}$ umgeht. Dazu werden zufällig l, so genannte Prototypen, aus der vorhandenen Datenmenge $\mathcal{D}$ gewählt und der Abstand von $D \in \mathcal{D}$ zu jedem der l Prototypen berechnet. Anhand dieser werden die Prototypen aufsteigend sortiert. Die so entstehende Reihenfolge der Prototypen gibt den Hashwert von D an.

Beispiel:
$dim = 2,\ \#Prototypen = 3$
$Prototypen:\ P1 = (1, 2),$
$P2 = (20, 200),\ P3 = (150, 60)$
$gesuchter\ Datenpunkt:$
$D = (15, 30)$
$\| D - P1 \|_2 \approx 31,3$
$\| D - P2 \|_2 \approx 170,1$
$\| D - P3 \|_2 \approx 138,3$
$\Rightarrow Aufsteigend\ sortiert\ nach\ den$
$Abständen\ ergibt\ sich\ für\ den$
$Hashwert\ von\ D\ :\ P1P3P2$

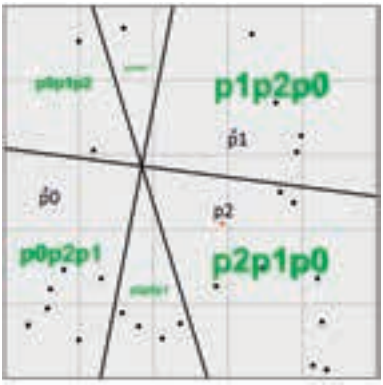


Abbildung 1: links: Ein Beispiel für die Berechnung des Hashwertes eines Datenpunktes mit drei Prototypen. rechts: Raumaufteilung mit dem Permutationsansatz im zweidimensionalen Raum mit drei Prototypen [B12].

Die Raumaufteilung entsteht dann dadurch, dass man jeweils zwischen zwei Prototypen eine imaginäre Strecke legt und orthogonal zu deren Mittelpunkt eine Hyperebene zieht. Dieses wird mit allen Paaren von Prototypen wiederholt. Die Buckets werden dann mit den entsprechenden Permutationen beschriftet (Abbildung 1 rechts).
Um eine effiziente Raumaufteilung zu erhalten, ist die Wahl der Prototypen, mit denen der Raum aufgeteilt wird, wichtig. Um Cluster innerhalb der Datenmenge gut zu unterteilen hat Broneske in [B12] herausgearbeitet, dass die Prototypen um das Cluster herum gelegt werden sollten.

2.3 Exact-Match-Anfrage beim Lokal Sensitiven Hashen

Beim LSH können sehr unterschiedlich ausgelastete Bereiche entstehen. Daher sollen die Prototypen so gesetzt werden, dass eine möglichst gleichmäßige Auslastung entsteht.
Damit die Exact-Match-Anfrage von einem Datenpunkt D effizient verarbeitet werden kann, wird der zugehörige Hashwert berechnet. Anschließend wird das dazugehörige Bucket mittels sequentieller Suche durchsucht, um festzustellen, ob D in der Datenmenge $\mathcal{D}$ enthalten ist.
Daher ist es von Vorteil, wenn jedes Bucket die selbe Anzahl an Elementen enthält. Sind

zum Beispiel in einem Bucket nur zwei Datenpunkte und in einem anderen 50, ist die Suche im zweiten Bucket im Vergleich zum ersten rechenintensiver. Deswegen soll eine ungleichmäßige Aufteilung der Datenpunkte auf die Buckets vermieden werden. Daher stellt sich die Frage, wie die l Prototypen zu wählen sind.
Betrachtet man die zufällige Wahl der Prototypen, wie sie in [CFN08] vorgestellt wird, so ist festzustellen, dass sehr unterschiedliche Raumaufteilungen entstehen können, die von unterschiedlicher Güte sind.
Deshalb wird im Folgenden ein mathematisches Optimierungsproblem aufgestellt, welches möglichst optimale Prototypen aus der Datenmenge $\mathcal{D}$ auswählt.

3 Optimierung der Exact-Match-Anfrage

Damit die Exact-Match-Anfrage effizient bearbeitet werden kann, ist es vorteilhaft, wenn die Buckets annähernd gleich viele Elemente enthalten. In diesem Fall lässt sich das Bucket in dem ein Datenpunkt D liegt leicht berechnen und das dazugehörige Bucket kann dann sequenziell durchsucht werden.

3.1 Verschiedene Optimierungsprobleme in der Übersicht

Es gibt verschiedene Möglichkeiten wie das Problem bearbeitet werden kann. Zum einen kann das Problem als Modell intuitiv geometrisch (1) formuliert werden, indem passende Beschreibungen für die entstehenden Polytope gefunden werden. Die Polytope entstehen dadurch, dass man zwischen je zwei Prototypen in der Mitte orthogonal eine Hyperebene legt. Betrachtet man nun die Schnitte von mehreren solcher Hyperebenen, entstehen Polytope, die sich im Verlauf des Optimierungsprozesses stetig ändern können. Weiterhin muss sichergestellt werden, dass in jedem dieser Polytope annähernd gleich viele Datenpunkte enthalten sind. Die Beschreibung der Polytope ist kompliziert, da diese sich im Verlauf der Optimierung ändern können. Daher ist dieses Modell rechnerisch sehr aufwändig.
Eine weitere Möglichkeit besteht darin, das Volumen der Polytope (2) zu optimieren. Dieses macht Sinn, wenn die Daten über denen optimiert wird gleichverteilt sind. Es soll erreicht werden, dass jedes Polytop annähernd dasselbe Volumen hat. Allerdings ist es mit der selben Begründung wie oben rechnerisch sehr aufwändig.
Eine weitere Idee besteht darin, dass eigentliche Problem approximativ zu formulieren (3). Somit ist das folgende Optimierungsproblem losgelöst von den Beschreibungen der Polytope. Dadurch ist es leichter zu verarbeiten.
Man legt jeweils gleich große Kugeln um die l Prototypen, sodass deren Mittelpunkt einer der l Prototypen ist. Nun besteht die Aufgabe darin, den Radius der Kugeln so zu minimieren, dass jeder beliebige andere Datenpunkt $D \in \mathcal{D}$ in mindestens einer dieser l Kugeln enthalten ist. In Abbildung 2 ist das Optimierungsproblem graphisch dargestellt. Die Kreuze stellen die Prototypen dar, um welche Kugeln gelegt werden sollen.

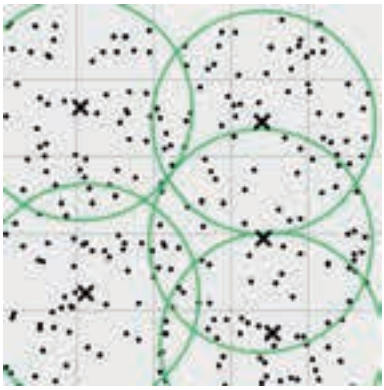


Abbildung 2: Optimierungsproblem (3) mit $l = 5$.

3.2 Mathematische Beschreibung des Optimierungsproblems

Es wird eine Datenmenge $\mathcal{D} \subseteq \mathcal{R}^d$ betrachtet, die n Datenpunkte enthält. Mathematisch sieht das Problem wie folgt aus:

$$\min_{r,x,y} r$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_i = l \tag{1}$$

$$\sum_{i=1}^{n} y_{i,j} \geq 1 \quad \forall j \in \mathcal{I} \tag{2}$$

$$y_{i,j}\sqrt{\| D_i - D_j \|_2} \leq r \quad \forall i,j \in \mathcal{I};\ D_i, D_j \in \mathcal{D} \tag{3}$$

$$y_{i,j} \leq x_i \tag{4}$$

$$r \geq 0 \tag{5}$$

$$x_i \in \{0,1\} \tag{6}$$

$$y_{i,j} \in \{0,1\} \tag{7}$$

Dabei ist x ein Entscheidungsvektor, für den $x_i = 1$ gilt, wenn der Datenpunkt D_i zu den l Prototypen gehört, ansonsten gilt $x_i = 0$. Die Matrix $y \in \{0,1\}^{n \times n}$ ist eine Entscheidungsmatrix, die angibt, ob ein Datenpunkt D_j von einer Kugel um D_i umgeben wird. Die Parameter $i,\ j$ sind aus der Menge $\mathcal{I} = \{1, 2, ..., n\}$. Zusammenfassend gilt:

$$x_i = \begin{cases} 1, & \textit{wenn } D_i \textit{ ein Prototyp ist} \\ 0, & \textit{sonst} \end{cases}$$

$$y_{i,j} = \begin{cases} 1, & \textit{wenn } D_j \textit{ in Kugel um } D_i \textit{ enthalten ist} \\ 0, & \textit{sonst} \end{cases} .$$

Die Nebenbedingungen des Optimierungsproblems werden im Folgenden erläutert: (1) gibt an, dass genau l Prototypen gesucht werden. In (2) wird beschrieben, dass jeder Datenpunkt in mindestens einer Kugel um einen Prototypen enthalten sein muss. Die Kugeldefinition mit positiven Radius r in (3) kommt genau dann zum Tragen, wenn $y_{i,j}$ eins ist. Dagegen sichert (4), dass $y_{i,j}$ nicht eins sein darf, wenn D_i nicht zu den Prototypen gehört, das heißt D_i ist dann kein Mittelpunkt einer Kugel.
Das oben genannte Optimierungsproblem ist linear und hängt nicht von der Dimension der Datenpunkte in $\mathcal{D}$ ab. Die verwendete Euklidische Norm lässt sich durch andere Normen ersetzen, zum Beispiel durch die Summennorm oder die Maximumsnorm. Man beachte, dass $y_{i,j}$ für große Datenmengen sehr groß wird.

Zur Lösung des Problems wird AMPL („A Mathematical Programming Language“) benutzt. AMPL ist eine mathematische Modellierungssprache, mit der ein mathematisches Optimierungsproblem in abstrakter Form formuliert werden kann [FGK03]. Dabei übersetzt AMPL das Optimierungsproblem für Optimierungsalgorithmen, die dieses dann lösen. Man beachte, dass passende Optimierungsalgorithmen gewählt werden müssen. Bei dem oben genannten Optimierungsproblem handelt es sich um ein lineares Programm, dafür ausgewählte Solver sind beispielsweise CPLEX und Gurobi.

4 Evaluation

In diesem Abschnitt wird die Auswertung des in 3.2 genannten Optimierungsproblems betrachtet. Dazu wird eine reale Datenmenge aus [AA96] verwendet. In einem Vergleich der Raumaufteilung mit optimierten und zufällig gewählten Prototypen wird die Relevanz des Optimierungsproblems dargestellt.

4.1 Setup und Durchführung

Die Testdaten bestehen aus 10.992 Datenpunkten mit jeweils 16 Dimensionen. Die Einträge der Daten sind ganzzahlige Werte zwischen 0 und 100. Aufgrund von Platzproblemen wird exemplarisch $l = 5$ gewählt, es sollen also 5 Prototypen aus den Datenpunkten so gewählt werden, dass der Raum möglichst optimal aufgeteilt wird. Als Optimierungssolver wurde CPLEX gewählt.
Bei der Berechnung des Optimierungsproblems hat sich herausgestellt, dass die Berechnungszeit von einer großen Datenmenge sehr hoch ist. So benötigt man für die Optimierung über einer Datenmenge mit 1000 Elementen mehrere Tage. Aufgrund der Berechnungszeit werden approximative Lösungen betrachtet, bei denen jeweils nur ein Teil der Daten zur Lösung des Problems gewählt wird. Daraus ergibt sich eine weitere Fragestellung, die später noch untersucht werden muss. Wie viele Daten müssen aus der Datenmenge betrachtet werden, damit keine großen Änderungen bezüglich der Abweichung auftreten. Exemplarisch wird die Auswirkung auf die Raumaufteilung durch den Permutationsansatz beim LSH betrachtet. Hierzu werden 100, 200 und 300 Daten aus der vorliegenden

Datenmenge ausgewählt. Auf diesen kleineren Datenmengen wird, mittels oben genanntem Optimierungsproblems, optimiert. Anschließend werden die ermittelten Prototypen auf die gesamte Datenmenge angewandt.

4.2 Ergebnisse

Als Erstes wird die Auswirkung der Optimierung der Prototypen auf den kleinen Datenmengen, mit 100, 200 und 300 Datenpunkten aus der gesamten Datenmenge, untersucht. Werden die Ergebnisse der Optimierung auf der Datenmenge mit 100 Elementen mit Ergebnissen vergleichen, die durch zufällige Wahl von 5 Prototypen aus der Datenmenge auswählt werden, ist zu erkennen, dass unterschiedlich viele Buckets entstehen können. Dies liegt daran, wie die Hyperebenen im Raum zueinander liegen. In diesem Beispiel entstehen 35, 30 und 24 Buckets für die Raumaufteilung in Abbildung 3. Der Durchschnitt gibt an, wie viele Datenpunkte in einem Bucket liegen müssen, damit die Aufteilung optimal ist. Die Balken des Diagramms stehen dafür, wie viele Buckets es in der entsprechenden Raumaufteilung gibt.
Bei der auf den optimierten Prototypen basierenden Raumaufteilung (a) in Abbildung 3 ist zu erkennen, dass die Mehrheit der Buckets dicht beim Durchschnitt liegt. Das bedeutet, dass die Mehrheit der Buckets annähernd optimal viele Elemente beinhalten. Vergleicht man dazu die Raumaufteilung (b), ist festzustellen, dass die Mehrheit der Buckets weiter vom Durchschnitt entfernt sind. Hierbei ist zu beachten, dass beide Verteilungen die gleiche Abweichung haben. Als Abweichung wird

$$\sum_{i=1}^{Anzahl\ der\ Buckets} |Anzahl\ Elemente\ in\ Bucket\ i - Durchschnitt|$$

bezeichnet.

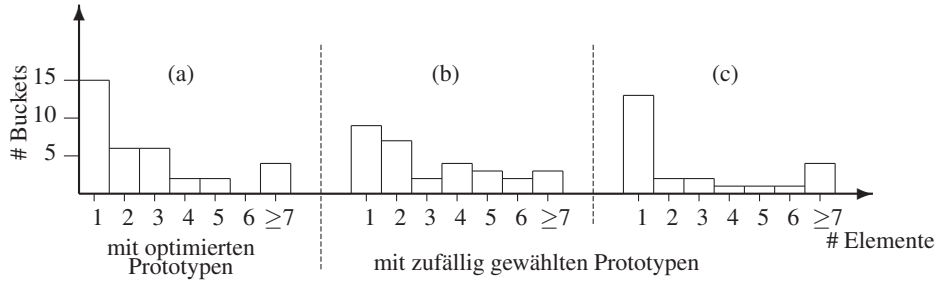


Abbildung 3: Verteilung der Daten auf die entstehenden Buckets für 100 Datenpunkte. Durchschnitt der optimierten Raumaufteilung (a): 2; Durchschnitt der Raumaufteilungen mit zufälligen Prototypen (b) und (c): 3 und 4.

Allerdings ist die optimierte Raumaufteilung besser verteilt, da es weniger Ausreißer nach oben hin gibt. Es gibt also weniger Buckets die viele Elemente enthalten. Die Raumaufteilung (c) in Abbildung 3 weist eine noch größere Streuung zu den Rändern, damit sind die

Buckets mit minimal beziehungsweise maximal vielen Datenpunkten gemeint, auf. Hier besitzt die Mehrheit der Buckets ein oder mehr als sieben Elemente, während das Optimum bei vier liegt. Ähnlich verhält sich die Raumaufteilung, wenn über einer Menge von 200 beziehungsweise 300 Daten optimiert wird.

Im Folgenden werden die Ergebnisse der Raumaufteilung der gesamten Datenmenge mit optimierten (Abbildung 5) und zufälligen Prototypen (Abbildung 6) betrachtet. Dazu werden die Prototypen, die zu den optimierten Raumaufteilungen in Abbildung 3 und die zu den optimierten Raumaufteilungen auf 200 und 300 Daten führen, gewählt. Wird über einer immer größer werdenden Menge optimiert, ist zu erkennen, dass die Schwankungen um das Optimum geringer werden. Je größer die Datenmenge über der optimiert wird ist, desto kleiner sollte die Abweichung werden. Weiterhin kommt es darauf an, welche Elemente man aus der gesamten Datenmenge wählt, um über diesen zu optimieren. In diesem Beispiel werden die ersten 100, 200 und 300 Elemente aus der gesamten Datenmenge gewählt.
Betrachtet man nun die zufälligen Ergebnisse in Abbildung 6, so ist festzustellen, dass die durch zufällige Wahl der Prototypen entstehenden Raumaufteilungen große Schwankungen aufweisen. So können durchaus bessere Lösungen als die in Abbildung 5 dargestellten Raumaufteilungen gefunden werden, jedoch entstehen auch schlechtere. Durch die optimierten Prototypen kann man die Schwankungen durch zufällige Wahl der Prototypen verringern.

In Abbildung 4 sind die Abweichungen vom Optimum graphisch dargestellt. Es ist gut zu erkennen, dass die Abweichung bei zufälliger Wahl der Prototypen stark variieren kann. Weiterhin wird deutlich, dass die Abweichungen geringer werden, wenn optimierte Prototypen gewählt werden, die über einer größer werdenden Teilmenge der gesamten Daten optimiert wurden.

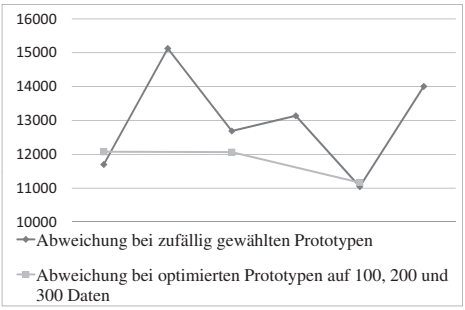


Abbildung 4: Vergleich der Abweichung mit zufällig gewählten und optimierten Prototypen.

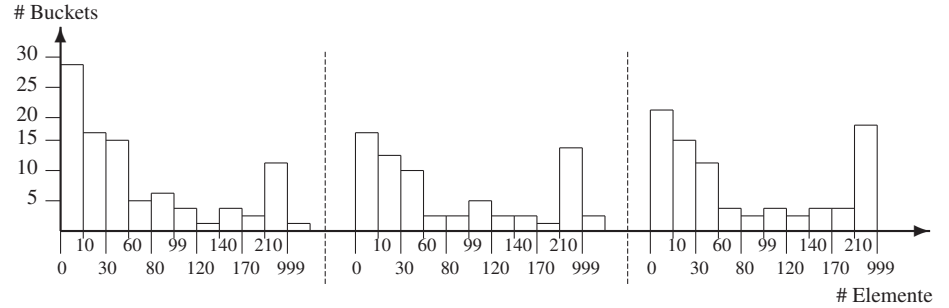


Abbildung 5: Verteilung der gesamten Datenmenge auf die entstehenden Buckets bei Anwendung der optimierten Prototypen auf 100, 200 und 300 Datenpunkten. Durchschnitt der jeweiligen Datenverteilungen auf den Raum: 109, 150 und 124.

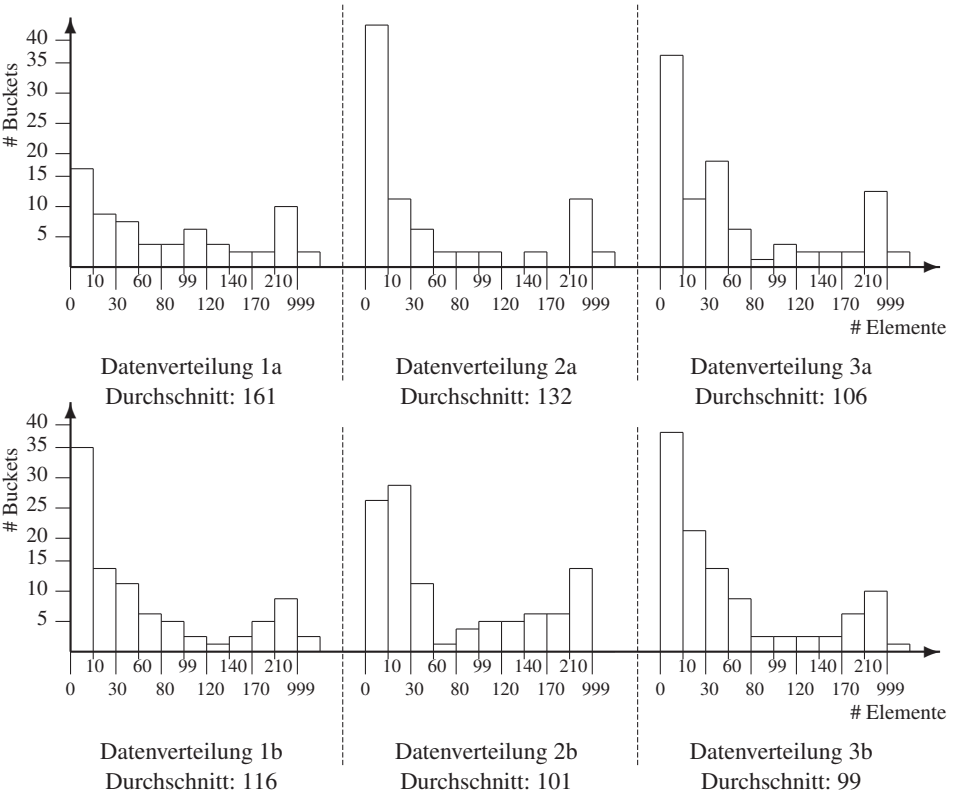


Abbildung 6: Verteilung der gesamten Daten auf die entstehenden Buckets für zufällig gewählte Prototypen.

5 Fazit und Ausblick

Es wurde ein Optimierungsproblem vorgestellt, welches die Wahl der Prototypen der Exact-Match-Anfrage beim permutationsbasiertem Ansatz des Lokal Sensitiven Hashens optimiert. Dazu wurden aufgrund von hohen Rechenzeiten Teilmengen aus dem gesamten

Datensatz ausgewählt, über denen optimiert wurde. Die so bestimmten Prototypen wurden anschließend auf die gesamte Datenmenge angewandt. Werden die Ergebnisse der Datenverteilung auf die Buckets mit Verteilungen der Daten bei zufälliger Wahl der Prototypen verglichen, so ist zu erkennen, dass die Abweichungen zum Optimum bei ersteren stabiler sind. Dabei sinkt die Abweichung bei einer größer werdenden Menge über der optimiert wird.
Da es bei der Wahl der Teilmenge aus der gesamten Datenmenge darauf ankommt, wie gut diese die gesamte Menge repräsentieren, bleibt zu untersuchen welche Elemente aus der Menge ausgewählt werden sollten.

Danksagung

Teile dieser Veröffentlichung beruhen auf Ergebnissen aus dem Forschungsvorhaben DigiDak (FKZ:13N10817), gefördert vom Bundesministerium für Bildung und Forschung (BMBF).

Literatur

[AA96] Alimoglu, F.; Alpaydin, E.: Methods of combining multiple classifiers based on different representations for pen-based handwriting recognition, In: TAINN. IEEE, pp.637-640, 1996.

[AI08] Andoni, A.; Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, Commun. ACM, 51(1): pp. 117-122, 2008.

[B12] Broneske, D.: Bachelorarbeit: Visuelle Analyse der Raumaufteilung und Bucketauslastung von permutationsbasierten Indexverfahren, Bachelorarbeit, Otto-von-Guericke Universität Magdeburg, 2012.

[CFN08] Chavez, E.; Figueroa, K.; Navarro, G.: Effective proximity retrieval by ordering permutations. In: IEEE Trans. on Pattern Analysis and Machine Intelli. 30, Nr. 9, pp. 1647-1658, 2008.

[FGK03] Fourer, R.; Gay, D. M.; Kernighan, B. W.: AMPL: A Modeling Language for Mathematical Programming, Second Edition, Brooks/Cole, Canada, 2003.

[GG97] Gaede, V.; Günther, O.: Multidimensional access methods, ACM Comp. Surveys , vol. 30, pp. 170-231, 1997.

[IM98] Indyk, P.; Motwani, R.: Approximate nearest neighbors: Towards removing the course of dimensionality. In: STOC, ACM, pp. 604-613, 1998.

D-VITA: A Visual Interactive Text Analysis System Using Dynamic Topic Mining

Nikou Günnemann
nikou.gholizadeh@rwth-aachen.de

Computer Science 5 - Information Systems & Databases
Prof. Dr. M. Jarke
RWTH Aachen University, Germany

Abstract: Recent developments in web technologies like Web 2.0 have led to the generation of massive amounts of data. The rapid growth of data makes knowledge extraction and trend prediction a challenging task. A recent approach for the unsupervised analysis of text corpora is dynamic topic mining. While there is a growing interest in using this technique, interactive analysis systems for dynamic topic mining are still in an early stage.

In this paper we present D-VITA, an interactive text analysis system that exploits dynamic topic mining to detect the latent topic structure and topic dynamics in a collection of documents. D-VITA supports end-users in understanding and exploiting the topic mining results, in visualizing the topic dynamics within document collections, and in browsing of documents based on shared topics. We present an application case for a scientific community that uses an instance of D-VITA for trend analysis in their data sources.

1 Introduction

There are many scientific, industrial or even historical issues which are researched and published online. Managing these large amounts of data is beyond human capability. It is almost impossible to gain an overview by hand. A possible solution to create an overview of large collections of documents is to use topic mining. Topic mining classifies the documents in topics on the basis of their content. The already existing documents can be substantially enhanced by publishing new documents. These dynamically changing document collections even complicate the topic mining since the aspects discussed in the stored documents might evolve.

A concrete example is given by the TEL-MAP research project [DK12, DCP+11], associated to the FP7 Cooperation Program of the European Commission, which tackles the challenge of analyzing the landscape of the Technology Enhanced Learning (TEL) research field. There are many technology providers and technology adapters in Europe that have a stake in the TEL research area. These actors are interested in finding topics which have recently attracted attention in the TEL community and are worth to be financially support in the future. Obviously, the recent trends in the TEL area change over time. By analyzing the TEL-MAP Mediabase, which stores information about TEL-related projects,

papers, and blogs [DK12, DCP+11], such trends might be automatically detected and can support the stakeholders decisions.

Overall, gaining an overview over large and dynamic collections of electronic documents is an increasing need in the area of industry and science. On this ground, a line of works has been focused on the development of algorithms for categorizing documents based on their inherent topics [BNJ03, BL06, WBH08].

Grouping of documents in topics by these mining methods is a prerequisite for gaining an overview over documents and finding hidden thematic structures. However, the generated results - huge lists of mere numbers - are still difficult to interpret by the user. Thus, visualizing the result of a topic mining method in a easily comprehensible way is also of fundamental importance. Via visualizing the result we finally enable the analysis of large document collections to interpret the right context of topics in different documents. Such a visualization should also be able to depict the temporal evolution of topics which plays a central role for dynamic data collections.

Based on this motivation we present in this paper an interactive text analysis tool for dynamic document collections. It supports the user, for example, in finding recent trends in a set of documents or visualizing the corresponding results. The remainder of this paper is structured as follows: In Section 2, we discuss relevant related work on topic mining. We then introduce in Section 3 the system architecture of D-VITA and we demonstrate different visualization concepts used in our system. In Section 4 we present results of a preliminary evaluation of D-VITA. We conclude in Section 5 with a summary and an outlook on further work.

2 Related Work

2.1 Topic Mining

Topic mining is the unsupervised discovery of thematic information hidden in a set of documents. Intuitively, a topic provides a compact description of the content of documents belonging to this topic. Technically, topics are often modeled as distributions over words. The basic idea is that documents discussing the same topic also primarily use the same words to a certain extent. Thus, these words can be used to describe the common topic and they give the topic its meaning. To generate an effective topic mining summarization, we have researched for different state-of-the-art summarization techniques for document databases. A prominent method to determine topics is Latent Dirichlet Allocation (LDA) [BNJ03]. Therefore, we employ LDA (and its extension for dynamic data; cf. next section) in our system to summarize topics for the whole database collection. In the LDA model, each document is allowed to belong to multiple topics. More specifically, each document is associated with a distribution over topics, i.e. documents belong to topics with different weights.

Since our system exploits the ideas of LDA, we first review the underlying statical assumption of the topic model described by [BNJ03]. LDA models the process of generating the

words of each document. Let K be a specified number of topics and V the total number of words. For each document the categorical distribution over topics is generated by the Dirichlet distribution $Dir_K(\vec{\alpha})$ controlled by $\vec{\alpha}$, a positive K-vector. Furthermore, LDA uses a Dirichlet distribution controlled by η to generate the actual topics (distributions over words). We let $Dir_V(\eta)$ denote a V-dimensional Dirichlet with scalar parameter η. The overall generative process of LDA is given as follows:

1. For each topic k,

(a) Draw a distribution over words $\vec{\beta}_k \sim Dir_V(\eta)$.

2. For each document d,

(a) Draw a vector of topic proportions $\vec{\theta}_d \sim Dir_K(\vec{\alpha})$.

(b) For each word,

(i) Draw a topic assignment $Z_{d,n} \sim Mult(\vec{\theta_d}), Z_{d,n} \in 1, ..., K$.

(ii) Draw a word $W_{d,n} \sim Mult(\beta_{\vec{Z_{d,n}}}), W_{d,n} \in 1, ..., V$.

This process is schematically shown in Figure 1. Each circular node in the figure corresponds to a random variable and edges denote dependencies between these variables.

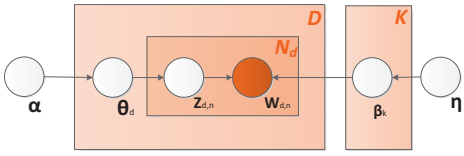


Figure 1: The graphical model of LDA

2.2 Dynamic Topic Mining

While static topic models consider a particular snapshot of a document collection without modeling the change over time, dynamic topic models consider the temporal dynamic by using a time-stamped collection of documents. Several dynamic topic models have been proposed in the literature [MZ05, BL06, WBH08, WM06, HCP+09, LBK09, AX10, JHL11, HYGD11]. We use one of the earliest approaches [BL06], to describe the general idea behind these models.

Considering the dynamic LDA in [BL06], the temporal dimension is addressed by two steps: First, the documents are partitioned into so called "epochs" with each epoch containing all documents of a certain time slices, e.g. of a specific year or a specific point in time. In a second step, the dependency between epochs is modeled. The generative process of this model is depicted in Figure 2. Figure 2 contains four instances of the static LDA illustration (cf. Figure 1) each of them representing one epoch.

The general idea of the dynamic LDA model is that the content of each topic, described by the categorical distribution β_k over words, changes smoothly over time. That is, words important for a topic at a specific time are very likely important also in the next time step. In Figure 2 this dependency is shown by the horizontal arrows pointing from left to right. Formally, we introduce for each topic k and epoch t a distribution $\beta_k^{(t)}$ and we model the

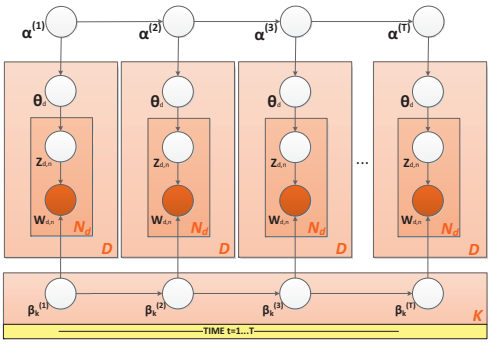


Figure 2: Graphical model of the dynamic LDA approach

smooth evolution between $\beta_k^{(t+1)}$ and $\beta_k^{(t)}$. We refer to [BL06] for more technical details about the actual evolution. Similarly, we model the evolution of the topic proportions by introducing $\alpha^{(t+1)}$ and $\alpha^{(t)}$.

3 The D-VITA System

In this section we present the D-VITA system which exploits the ideas of dynamic LDA and offers a web based user interface to visually interact with results of dynamic LDA on different data sources. We divided the system architecture in three layers: data layer, application layer, and presentation layer. Figure 3 shows an overview of the system architecture and Figure 4 a screenshot of the current version of D-VITA.

3.1 Data Layer

Besides the different kinds of raw data (e.g. blogs, posts, research papers, web content), the data layer stores the results generated by the application layer. This includes the documents in a form preprocessed for dynamic LDA as well as the detected topics. All relevant information is stored in a relational database management system.

3.2 Application Layer

As illustrated in Figure 3, we divide the application layer into two main components for data processing: online and offline component.

The offline component performs tasks that are only executed infrequently and potentially require lengthy processing. In the offline component we integrate the data preprocessing,

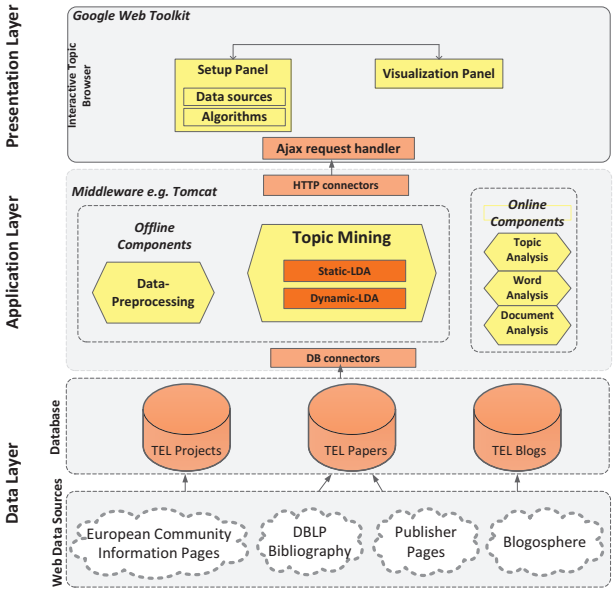


Figure 3: System Architecture of D-VITA

the execution of topic mining algorithms, and the computation of the documents' similarity. Data preprocessing is required since the architecture is intended to be independent of particular data schemas. In the TEL Mediabase, for instance, blog posts in the blog database are used as input documents for the topic mining algorithms. The preprocessing step also extracts the set of distinct words from the raw data, it applies stop-word list, and performs stemming to reduce inflected or derived words to their stem.

In contrast to the offline component, the tasks of the online component, are executed frequently in response to user actions in the D-VITA graphical user interface and should instantly retrieve and display results in the browser. The online component contains several tools that support the user in interacting with the topic mining results. The user interacts with three general concept (i.e. documents, topics and words). Subsequently, the results are visualized by the presentation layer which is described in Section 3.3.

Technical Realization The application layer is implemented based on Apache Tomcat[1]. Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technology. It supports the implementation and execution of Java programs (servlets) in a webserver environment. Each servlet provides publicly accessibly methods which can be invoked by the clients. The client-server communication is automatically handled using protocols like HTTP.

[1] http://tomcat.apache.org/

Figure 4: *D-VITA's visual summary of topics. In the visualization, the themeriver (top-right-pane) represents the selected topics and their evolution over different time slices. By selecting (i.e. clicking) a topic in the themeriver, a list of documents exhibiting the selected topic is shown in the bottom-right pane. By selecting a document from this list, e.g. the conference paper "The Language Technologies for Lifelong Learning Project" as in the figure, the distribution of topics in the document is visualized as a pie chart. In the example, the selected document is associated with 77.4 percent to the topic "learning, mobile, technology, language" and 22 percent with the topic "educational, development, based, technology". The document's content can be inspected by clicking the button "content". Additionally, a user can retrieve for each document a list of similar documents based on shared topics.*

3.3 Presentation Layer

The key idea in representing LDA's results and the correlation between the results is to exploit the paradigm of *Visual Analytics* [KMS+08]. Visual Analytics aim at integrating the human capabilities into the data analysis process by using visual representations and interaction techniques. Thus, the user is involved in the analysis process and its interpretation is supported by visualizing important information. Various methods of visual analytics for the analysis of document content have been proposed in previous work, e.g. for topic-based navigation in Wikipedia [CB12] and in TIARA [LZP+12]. Existing systems, however, often fail to support important use cases like handling of dynamic data or the detection and highlighting of emerging topics in the data. Furthermore, none of the existing tools decouples the topic mining from the data sources in a way that allows "plugging in" arbitrary databases, which is achieved in D-VITA by mapping the source data schema to a simple schema that can be used by the offline component for data preprocessing.

Presentation of Topic Analysis In D-VITA, topic analysis can be performed in different ways. One possibility, is to present topics in conjunction with the words assigned to

the corresponding topic. Based on the LDA procedure, each topic $k \in 1, ..., K$ can be described by the most occurring words in it. In D-VITA, we consider the most four occurring words to name a topic. Formally, the four most important words $words_k \subseteq \{1, \ldots, V\}$ for topic k can be computed based on its word distributions $\beta_k^{(t)}$:

$$words_k := 4 \arg\max_{w \in \{1,\ldots,V\}} \left\{ \sum_{t=1}^{T} \beta_k^{(t)}[w] \right\}$$

where the function $4 \arg\max$ denotes the generalization of the $\arg\max$ function to return the *four* elements with the highest values. In doing so, D-VITA generates at each start a list of selectable topics from the database. On this way, each user has an overview of topics in the database, as depicted in Figure 4 (left part).

By selecting several topics, the user can analyze the topics evolution, e.g. whether they are emerging or declining. For this purpose D-VITA visualizes the topics by a themeriver graph [HHN00]. This graph is illustrated in Figure 4 in the upper part. Each current in the themeriver presents the dynamic development of a selected topic in a time interval. The wider the current, the more relevant is a topic, i.e. the more documents contribute to this topic. Since each document contributes to each topic with a different degree, the relevance of topic k at time t is formally defined as

$$relevance(k, t) := \sum_{d \in DB_t} \theta_d[k]$$

where DB_t is the set of documents belonging to time t and θ_d is the topic distribution for document d. A user can also interact with the content of a certain topic in a dynamic topic view. Each current in the themeriver is selectable to allow an in-depth analysis. This is an another possibility to present topics in conjunction with documents that is explained in the following.

Presentation of Document Analysis In D-VITA document analysis can be used to analyze the topic-document correlations. By selecting a topic in the theme river corresponding to a certain point in time a list of documents is displayed in the "Document Browser" panel (at the bottom). These documents correspond to the most representative documents for the selected topic at the selected point in time. Note that the representative documents for the selected topic may change when considering different points in time. Formally, given a selected topic k at time t, the set of documents $Repr_{k,t} \subseteq DB_t$ is determined based on the following equation:

$$Repr_{k,t} := r \arg\max_{d \in DB_t} \{\theta_d[k]\}$$

Again, $r \arg\max$ denotes the function returning the r elements with the highest value and DB_t is the set of all documents belonging to time t.

Although these documents are the most representative ones for the selected topic, the documents will typically expose several additional topics. Thus, for each document we

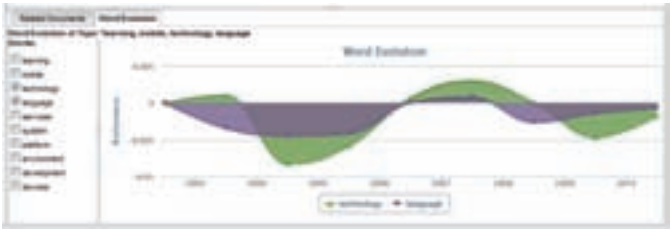


Figure 5: Word evolution pane of D-VITA

represent the share of topics by colored slices in a pie chart as shown in Figure 4. Additionally, the content of each document can be inspected by clicking on the button content from the table.

D-VITA also offers a way to analyze the document-document correlation. To achieve this, we first measure the similarity between the selected document d_1 and the whole documents $d_i \in DB_t$ in the respective time slice by using Jensen Shannon Divergence which is formally defined as:

$$JSD(d_1 \parallel d_i) := \frac{1}{2}D(d_1 \parallel M_i) + \frac{1}{2}D(d_i \parallel M_i)$$

where D is Kullback-Leibler-Divergence and

$$M_i := \frac{1}{2}(d_1 + d_i) \quad , \quad d_i \in DB_t$$

As explained earlier, the computation of similarity scores for pairs of documents is executed in the offline component, while searching for similar documents to the selected document and listing the result is executed by the online component. Note that the related documents might belong to the same time step as the input document as well as to different time steps. Our GUI enables the user to browse between the documents in different time steps (cf. buttons below "Hide Related Documents" in Figure 4). In doing so, a user navigates through future or previous time slices different than the current time slice.

Presentation of Word Analysis Finally, the D-VITA system allows to perform a word analysis. Giving a certain keyword in a search field, D-VITA returns the results in two aspects: as a list of documents and as a list of topics with the keyword in their context. Additionally, based on the properties of dynamic LDA, the word distribution of a topic may change over time. The word distribution depends on the number of produced articles and papers that contain the word. For this reason, each topic evolves in its relevant words. This word evolution can be illustrated in the lower part of our system as an alternative to the document browser (cf. Figure 5). Based on the dynamic nature of the data, also the above mentioned keyword search is time dependent, i.e. the lists of topics and documents containing the keyword may change at different times.

4 Runtime Metrics

The D-VITA system was deployed using different production databases from the TEL-Map project [DK12]: a DB2 database that stores blogs crawled from the web and an Oracle database that stores data based on DBLP and CiteSeerX. Additionally, D-VITA was tested on an extract of the 20 Newsgroups dataset[2]. The table below shows the runtime of the offline component for each data set.

Database	Size	Time steps	# Topics	Offline Components		
				LDA	Similar docs	Preprocessing
ICALT Papers	2,227	4 (Year)	15	4.1 h	1.9 min	18 sec
20Newsgroup	10,757	9 (Year)	6	16.4 h	9.4 min	2.5 min
Blogs	11,221	23 (Month)	15	6.6 days	10.3 min	9.7 min

Considering the elapsed time for the offline components, our results show that the run time increases depending on different factors. Besides the size of data, the number of time steps on which LDA runs, increases the runtime. LDA requires 6.6 days for the blogs data, which covers a period of 23 time steps, while it needs 4.1 hours for the ICALT papers, covering a period of 4 time steps only. Additionally, a high number of selected topics, influences the run of the LDA algorithm to fit the topics as precisely as possible. This, also increases the runtime as shown in the case of 20Newsgroup and blogs. Note that these experiments were conducted using the single-threaded code provided by the authors [BL06]. By using parallelized implementations of LDA, a higher efficiency can be achieved.

In contrast to the offline components, the online component needs only seconds of time to present the results on demand. The relevant factor for the run time of the online component is the speed of the internet connection since for each query D-VITA executes a connection to the database.

5 Conclusion

In this paper, we present D-VITA, a novel interactive visual text analysis system based on dynamic topic modeling that is designed to support users exploring and interacting with numbers of documents. It presents an overview of the topics hidden in the documents, highlights the evolution of selected topics, and also displays the evolution of words that establish a particular topic. To increase flexibility of D-VITA, we allow to process arbitary data sources. We have deployed a prototype of D-VITA in the TEL-Map project which maintains in its "Mediabase" several databases related to the research area of technology enhanced learning (TEL), including TEL-papers, blogs and projects. The runtime analysis showed that there is room for improving the performance of the offline components, which will be tackled in future work. Currently we are focusing on improving and empirically evaluating the usability of the web application for both data providers and end-users in an empirical study with TEL-Map partners.

[2] http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

References

[AX10] Amr Ahmed and Eric P. Xing. Timeline: A Dynamic Hierarchical Dirichlet Process Model for Recovering Birth/Death and Evolution of Topics in Text Stream. In *UAI*, pages 20–29, 2010.

[BL06] David M. Blei and John D. Lafferty. Dynamic topic models. In *ICML*, pages 113–120, 2006.

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[CB12] Allison June-Barlow Chaney and David M. Blei. Visualizing Topic Models. In *ICWSM*, 2012.

[DCP+11] Michael Derntl, Adam Cooper, Manh Cuong Pham, Ralf Klamma, and Dominik Renzel. Mediabase Ready and First Analysis Report, TEL-Map Deliverable D4.3. 2011.

[DK12] Michael Derntl and Ralf Klamma. A Mediabase for Technology Enhanced Learning in Europe. *IEEE Learning Technologies Newsletter*, 14(3):2–5., 2012.

[HCP+09] Qi He, Bi Chen, Jian Pei, Baojun Qiu, Prasenjit Mitra, and C. Lee Giles. Detecting topic evolution in scientific literature: how can citations help? In *CIKM*, pages 957–966, 2009.

[HHN00] Susan Havre, Elizabeth G. Hetzler, and Lucy T. Nowell. ThemeRiver: Visualizing Theme Changes over Time. In *INFOVIS*, pages 115–123, 2000.

[HYGD11] Liangjie Hong, Dawei Yin, Jian Guo, and Brian D. Davison. Tracking trends: incorporating term volume into temporal topic models. In *KDD*, pages 484–492, 2011.

[JHL11] Yookyung Jo, John E. Hopcroft, and Carl Lagoze. The web of topics: discovering the topology of topic evolution in a corpus. In *WWW*, pages 257–266, 2011.

[KMS+08] Daniel Keim, Florian Mansmann, Jrn Schneidewind, Jim Thomas, and Hartmut Ziegler. Visual analytics: Scope and challenges. *Visual Data Mining*, pages 76–90, 2008.

[LBK09] Jure Leskovec, Lars Backstrom, and Jon M. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD*, pages 497–506, 2009.

[LZP+12] Shixia Liu, Michelle X. Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. TIARA: Interactive, Topic-Based Visual Text Summarization and Analysis. *ACM TIST*, 3(2):25, 2012.

[MZ05] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD*, pages 198–207, 2005.

[WBH08] Chong Wang, David M. Blei, and David Heckerman. Continuous Time Dynamic Topic Models. In *UAI*, pages 579–586, 2008.

[WM06] Xuerui Wang and Andrew McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *KDD*, pages 424–433, 2006.

Performance-Analyse auf Mainframe-Systemen mittels Profiling

Stefan Laner und Matheus Hauder

Lehrstuhl für Informatik 19 - sebis
Technische Universität München (TUM)
Boltzmannstrasse 3
85748 Garching bei München
{laner,hauder}@in.tum.de

Abstract: Eine Optimierung der Performance von Anwendungen verbessert deren Laufzeiten und senkt deren Betriebskosten. Um Performance-Optimierungen vornehmen zu können, müssen zunächst Optimierungspotentiale identifiziert werden. Dazu werden Performancedaten erfasst und analysiert. Unternehmensanwendungen, die auf IBM Mainframes betrieben werden, lassen sich mit aktuellen Verfahren allerdings nur umständlich analysieren. Die derzeit gängigen Werkzeuge liefern eine Vielzahl an Informationen, die manuell ausgewertet werden müssen. Eigentlich zusammengehörige Informationen sind dort zum Teil über verschiedene Reports verteilt und auffällige Programmteile oftmals nur über geeignete Aggregation identifizierbar. Der in diesem Artikel vorgestellte Ansatz überführt die Daten aus einem konkreten Profiling-Werkzeug in ein Meta-Modell, welches für eine automatische Analyse herangezogen werden kann. Zudem beschreibt er Beispiele für Anti-Pattern, also Muster von bekannten Performance-Problemen, die in realen Systemen beobachtet wurden und wie sich diese aus dem Meta-Modell ermitteln lassen.

1 Einführung

IBM Mainframe Systeme finden entgegen früherer Prognosen nach wie vor starke Verwendung im Umfeld von Großunternehmen und Behörden. Eine Neuentwicklung der darauf betriebenen Anwendungen oder gar eine Migration auf andere Plattformen ist aus Zeit- und Kostengründen sowie auf Grund des damit verbundenen Umstellungsrisikos in der Regel nicht ohne Weiteres durchführbar. Eine Wartung und Optimierung dieser Systeme ist somit zwingend erforderlich, um Kosten zu senken und mit wachsenden Datenbeständen umgehen zu können.
Grundsätzlich stehen zwei Ziele im Fokus einer Optimierung - CPU-Nutzung und Laufzeit. IBM berechnet für die Nutzung eines Mainframes Lizenzgebühren, dessen Höhe sich nach einer individuellen Vereinbarung zwischen dem Nutzer und IBM richtet. Im Allgemeinen jedoch stellt die CPU-Nutzung einen wesentlicher Einflussfaktor dafür dar, weshalb eine Optimierung dieser unmittelbare finanzielle Vorteile für den Betreiber erzielt. Verringerte Laufzeiten verkürzen die Wartezeiten für Benutzer bei interaktiven Anwendungen sowie die Zeitfenster, die zur Bearbeitung von Batchläufen notwendig sind.

Aktuell existieren zwar Werkzeuge zur Messung der Performance von Anwendungen auf Mainframes, allerdings gestaltet sich die Auswertung ihrer Ergebnisse schwierig. Performanceexperten, die damit arbeiten sind gezwungen, manuell die notwendigen Informationen zu extrahieren. Abhängig von der Erfahrung dieser Experten werden gewisse Probleme möglicherweise garnicht erkannt. Eine automatische Auswertung der Reports von solchen Messungen soll helfen, die notwendigen Informationen zu erfassen und bereits gewonnenes Expertenwissen in Form von Anti-Pattern einfließen zu lassen. Dazu werden die Reports eines solchen Werkzeuges eingelesen und in ein Meta-Modell überführt. Auf diesem Modell wird anschließend eine automatische Suche nach definierten Anti-Pattern vorgenommen. Performanceexperten sollen dadurch unterstützt werden, die für sie relevanten Informationen zu bekommen und automatisch auf mögliche Optimierungspotentiale hingewiesen werden.
Dazu werden in Abschnitt 2 dieses Artikels die derzeitigen Verfahren und deren Probleme erläutert. Darauf aufbauend wird in Abschnitt 3 das eingesetzte Meta-Modell zur Speicherung von Performance-Daten und beispielhaft einige Anti-Pattern und deren Erkennungsverfahren vorgestellt. Der Abschnitt 4 fasst die Ergebnisse dieses Artikels noch einmal zusammen und gibt einen Ausblick auf weitergehende Forschungsthemen.

2 Derzeitiger Stand der Analyseverfahren

Grundsätzlich unterscheidet man zwei verschiedene Ansätze zur Analyse des Laufzeitverhaltens von Anwendungen: Statische Analyseverfahren und dynamische Analyseverfahren. Die folgenden beiden Unterkapitel beschreiben diese Verfahren mehr im Detail.

2.1 Statische Analyseverfahren

Statische Verfahren untersuchen die zu prüfende Software ohne diese auszuführen. Sie arbeiten in der Regel auf Basis des Sourcecodes oder Objektcodes, aber auch auf höheren Abstraktionsschichten, wie z.B. auf Modellen bei MDD Ansätzen. Statische Analyse-Werkzeuge überführen den Code des zu untersuchenden Programms in ein Modell, welches die möglichen Programmzustände und deren Übergänge repräsentiert. Auf diesem Modell werden anschließend Analysen durchgeführt. Das gebildete Modell umfasst alle möglichen Zustandsübergänge [Ern03], zur tatsächlichen Ausführung des Programms werden jedoch in Abhängigkeit der Eingabe nur bestimmte Zustandsübergängen wirklich ausgeführt (zum Beispiel durch bedingte Anweisungen). Ausführungspfade für konkrete Eingaben lassen sich aus diesem allgemeinen Modell ableiten. Da die Eingaben im Allgemeinen jedoch nicht bekannt sind, müssen Verfahren entwickelt werden, um die Wahrscheinlichkeit für die Wahl eines bestimmten Pfades abzuschätzen.
Ein großer Vorteil statischer Analyseverfahren ist, dass alle potentiellen Ausführungspfade untersucht werden können, da diese aus dem gebildeten Modell abgeleitet werden können. Da die statische Analyse ohne tatsächliche Ausführung des Programms durchgeführt wird, müssen zudem keine Abhängigkeiten zu bestimmten Hardware-Plattformen, Betriebssys-

temen oder externen Programmen erfüllt werden.
Dieser Vorteil erweist sich jedoch gleichzeitig als Nachteil bei der Messung von Performancedaten. Da die Analyse wie beschrieben umgebungsunabhängig stattfinden, können auch keine umgebungsspezifischen Performanceeinflüsse detektiert werden.
Hauptnachteil statischer Analyseverfahren ist jedoch, dass eine Berücksichtigung des dynamischen Verhaltens nicht erfolgen kann. So können beispielsweise die Anzahl an Schleifendurchläufen oder die Ausführungen bedingter Anweisungen nicht ermittelt werden, sondern müssen wie bereits erwähnt mittels besonderer Verfahren näherungsweise abgeschätzt werden.

2.2 Dynamische Analyseverfahren

Bei diesen Verfahren wird auch das dynamische Verhalten der untersuchten Programme berücksichtigt. Die Programme werden dafür zur Ausführung gebracht, die notwendigen Informationen während dessen entweder von zusätzlichem Code, der mit dem zu untersuchenden Programm ausgeführt wird (Messcode) geliefert oder von externen Messprogrammen „beobachtet“ (Sampling). Der Messcode kann beispielsweise einen Zähler definieren und am Anfang des Rumpfes einer bestimmte Funktion diesen inkrementieren, um die Anzahl der Aufrufe dieser Funktion zu zählen. Werkzeuge, die solch dynamische Analysen durchführen nennt man „Profiler“ und deren Anwendung „Profiling“. Unter Instrumentierung versteht man die Einbringung von zusätzlichem Messcode für die dynamische Analyse, welche

- manuell durch den Programmierer
- durch den Compiler
- zur Programmausführung

erfolgen kann. Eine manuelle Instrumentierung des Quellcodes eignet sich, wenn lediglich einzelne Abschnitte, beispielsweise einzelne Funktionen, gemessen werden sollen. Eine Messung größerer Programmabschnitte oder gar ganzer Programme sowie eine feingranulare Messung wird schnell sehr aufwändig, da viele solche Messbefehle eingefügt werden müssen. Zudem wird die Lesbarkeit des Codes durch die zusätzlichen Anweisungen stark beeinträchtigt. Aus diesem Grund werden die Instrumentierungen häufig automatisiert vom Compiler eingefügt. Der Quelltext bleibt dadurch unverändert und eine hohe Granularität kann einfach erreicht werden. Eine weitere Möglichkeit besteht darin, den Messcode erst zur Programmausführung einzufügen. Dies kann vor dem Laden des Programms in den Speicher oder bei Ausführung in einer virtuellen Maschine durch diese Maschine erfolgen. Dynamische Analyseverfahren können ebenfalls danach eingeteilt werden, von welcher Ebene ausgehend die Instrumentierung erfolgt [Net04]:

- Source analysis
  Analyse ausgehend vom Source-Code. Abhängig von der verwendeten Programmiersprache, unabhängig von der Zielplattform (Architektur und Betriebssystem)

- Binary analysis
  Analyse ausgehend von Maschinen-Code. Unabhängig von der verwendeten Programmiersprache, abhängig von der Zielplattform

Nicholas Nethercote beschreibt in seiner Dissertation [Net04] die Software „Valgrind“ [1], welche dynamische Instrumentierung zur Laufzeit verwendet.
Vorteile eines Profilings sind dessen Genauigkeit [Ern03] und die Menge der erfassbaren Informationen. Je nach verwendeter Profiling-Methode können neben Laufzeiten beispielsweise auch Call-Stacks oder Aufrufparameter von Funktionen aufgezeichnet werden. Ungenauigkeiten, die bei einer statischen Analyse durch Näherungen und Annahmen über die Zielplattform entstehen, treten bei dynamischen Verfahren nicht auf, da ein dynamisch analysiertes Programm direkt auf der Zielplattform zur Ausführung kommt. Effekte, die beispielsweise durch I/O Latenzen oder Nebenläufigkeiten entstehen, gehen somit ebenfalls in die Messung ein.
Nachteil dynamischer Verfahren ist, dass lediglich einzelne Programmausführungen gemessen werden, also Ausführungen mit einer konkreten Eingabe und einer konkreten Menge an Randbedingungen. Die Messung eines dynamischen Verfahrens kann also keine allgemeine Aussage für alle möglichen Eingaben treffen, ebenso wenig wie für alle möglichen Rahmenbedingungen (z.B. Auslastung des Systems während der Programmausführung).
Die Einbringung von zusätzlichem Code führt selbstverständlich zu einer Verlangsamung der Anwendung. Ein weiterer Nachteil bei der manuellen Instrumentierung und der Instrumentierung durch den Compiler besteht darin, dass das zu untersuchende Programm neu übersetzt werden muss. Diese Eigenschaft macht die beiden genannten Verfahren ungeeignet für die Messung von produktiven Anwendungen, da mit einem neuen Übersetzen auch ein erneutes Deployment nötig wäre. Eine Instrumentierung zur Laufzeit behebt dieses Problem, bringt jedoch weiteren Overhead mit sich.
Ein weiteres Profiling-Verfahren ist das sogenannte „Sampling“, bei dem periodisch von einem eigenständigen Messprogramm der aktuelle Ausführungszustand einer zu messenden Anwendung abgefragt wird. Die gemessene Anwendung liegt dabei unverändert vor, eine Neuübersetzung entfällt vollständig, weshalb mit dem Sampling-Verfahren auch produktive Anwendungen gemessen werden können. Durch das Sampling ist die Genauigkeit der Messung nicht so hoch wie bei einem Profiling mittels Instrumentierung. Jegliche Programmausführung, die zwischen zwei Samples geschieht, wird nicht erfasst. Auf eine ausreichende Abtastrate ist deshalb zu achten, wobei eine höhere Rate zwar die Genauigkeit, aber auch den Messaufwand erhöht. Da ein Sample lediglich den aktuellen Zustand erfasst und nicht erkennt, wie das Programm in diesen Zustand gelangt ist, werden keine Call-Stacks erfasst.
Die Software „Strobe“ von Compuware [2] führt ein Profiling auf Mainframe-Systemen mittels Sampling durch. Die von Strobe generierten Reports werden für den unter Abschnitt 3 beschriebenen Ansatz eingesetzt.

[1] http://www.valgrind.org, zuletzt aufgerufen am 26.11.2012
[2] http://www.compuware.com, zuletzt aufgerufen am 26.11.2012

2.3 Auswertung

Grundsätzlich liefert ein Profiling von großen Anwendungen eine Vielzahl an Informationen, die nur mittels Filtern, Aggregationen und visuellen Darstellungen menschlich auswertbar sind. [Par08] beschreibt diese Informationsflut im Rahmen der Analyse von JEE Systemen. Strobe beispielsweise erzeugt ein Textdokument, dessen Inhalt häufig mehr als 10000 Zeilen umfasst. Die benötigten Informationen um konkrete Performance-Probleme zu untersuchen sind auf verschiedene Abschnitte (Sections) innerhalb des gesamten Dokuments verteilt. Abbildung 1 zeigt beispielhaft die in einem Strobe-Report enthaltenen Sections.

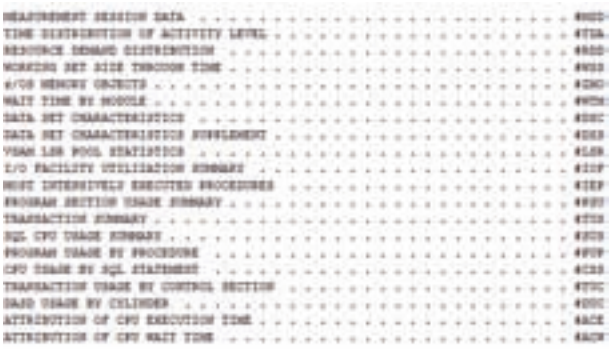


Abbildung 1: Sections innerhalb eines Strobe-Reports

Für einige Systemmodule weist Strobe die CPU-Nutzung aufgeschlüsselt nach Aufrufern aus. Aufrufe von Systemmodule werden stets (direkt oder indirekt) durch Anwendungsprogramme ausgelöst. Somit können Situationen auftreten, in denen ein Anwendungsmodul selbst kaum CPU-Ressourcen verbraucht, jedoch Systemmodule aufruft, welche in Summe dann einen nennenswerten Verbrauch verursachen. Solche Szenarien sind mittels manueller Analyse nur sehr schwer zu erkennen, da ggf. die einzelnen Systemmodule selbst ebenfalls keinen auffällig hohen Verbrauch aufweisen. Abbildung 2 zeigt einen Ausschnitt aus einem solchen Report. Das Modul „ABC530“ verursacht in diesem Beispiel 0,3% des gesamten CPU-Verbrauchs. Ebenso kann man ablesen, in welchem Bereich des Codes wieviel Verbrauch verursacht wurde.

3 Ansatz zur Performance-Analyse auf Mainframe-Systemen

Die von Strobe erzeugten Textdokumente haben eine - wenn auch ggf. versionsabhängige - klar definierte Struktur, was ein automatisches Einlesen ermöglicht. Die dadurch ermittelten Informationen werden in ein Meta-Modell überführt und in einer Datenbank für nachfolgende Analysen persistiert. Anschließend werden Beispiele für häufig auftretende Anti-Pattern in Unternehmensanwendungen vorgestellt und erläutert, wie diese auf Basis

Abbildung 2: Auszug aus einem Strobe-Report

des beschriebenen Datenmodells detektiert werden können.

3.1 Meta-Modell zur Speicherung von Performance-Daten

Die extrahierten Informationen werden in ein Meta-Modell (Abb. 3) überführt, welches später sowohl zur Erzeugung von für manuelle Interpretationen optimierten Darstellungen, als auch zur automatischen Interpretation genutzt wird.

Kern des Modells ist die Klasse *ProgramPart*, welche die Generalisierung der verschiedenen Teile eines Programms repräsentiert. Dabei findet das Composite-Pattern nach Gamma et al. [GHJV94] Verwendung. Für jede Instanz von *ProgramPart* bzw. dessen Subklassen wird die prozentuale CPU-Nutzung und die verursachte Wartezeit gespeichert; der Zugriff auf diese Informationen ist, durch die Vererbung bedingt, für alle Subklassen einheitlich. Diese hierarchische Struktur erlaubt es über einen Drilldown den gewünschten Detaillierungsgrad einzustellen und somit die Menge der Informationen erfassbar darzustellen. Die Klasse *Program* repräsentiert das gesamte gemessene Programm. Die von Strobe für das gesamte Programm gelieferten „ServiceUnits“ stellen einen maschinenunabhängigen Wert für die CPU-Nutzung dar. Dieser Wert ist nicht zu verwechseln mit den von IBM zur Abrechnung der Lizenzkosten genutzten Einheiten, die ebenfalls „ServiceUnits“ genannt werden. IBM’s ServiceUnits sind nicht maschinenunabhängig. Programme bestehen aus mindestens 1 Modul, jedes Modul wiederum kann verschiedene Sections (Control-Sections) beinhalten. Dieser Unterteilung spiegelt sich in den entsprechenden Klassen (*Module*, *Section*) im Meta-Modell wider. Das Sampling von Strobe liefert Ausführungszeiten noch detaillierter unterteilt in hexadezimale Intervalle innerhalb einer Section. Deshalb stellt das Meta-Modell noch die Klasse *SectionInterval* zur Aufnahme der Start- und Endadresse eines solchen Intervalls bereit. Eine Sonderform der Klasse *Module* stellt das *SystemModule*

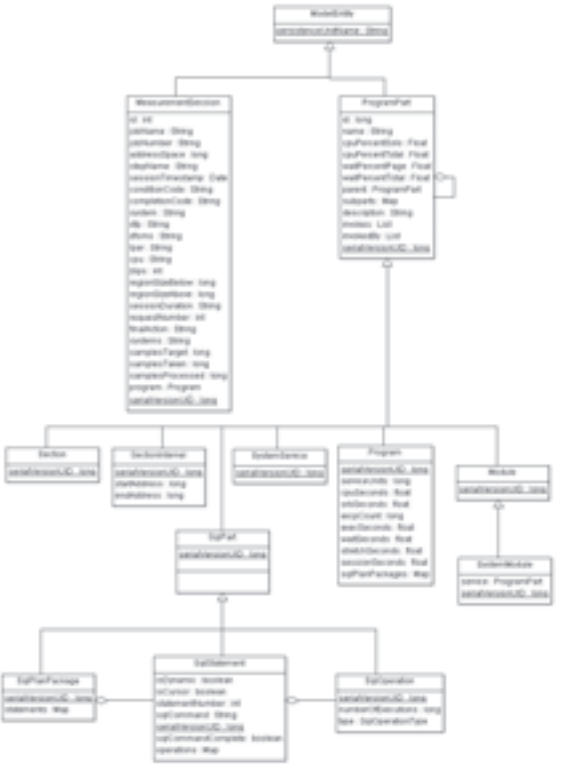


Abbildung 3: Meta-Modell zur Speicherung von Performance-Daten

dar. Ein *SystemModule* ist einem bestimmten Systemdienst (*SystemService*) zugeordnet. Für SQL-Ausführungen auf DB2 liefert Strobe den Namen des Plans (bzw. des Packages) und die ausgeführten Statements mit deren SQL-Text. Je nach Einstellung der Messung ist der SQL-Text jedoch nicht vollständig dargestellt (Standard-Einstellung 300 Zeichen). Die Anzahl der Ausführungen wird separat für jede Operation eines Statements erfasst. Eine Operation ist dabei ein Open, Fetch oder Update. Abbildung 4 zeigt dabei die Darstellung von SQL-Statements in Strobe-Reports.

SQL-Statements können statisch oder dynamisch sein. Statische Statements sind vorcompiliert, deren Zugriffspfad wird bei der Erstellung (Binden zu einem Plan) ermittelt. Dynamische Statements hingegen werden erst zur Laufzeit interpretiert und optimiert. Diese Unterscheidung ist für Performance-Betrachtungen wichtig, da statische Statements zwar den zusätzlichen Aufwand zur Interpretation des SQL-Textes und Erstellung des Zugriffspfads sparen, jedoch den Zugriffspfad nicht auf die zur Laufzeit aktuelle Datenverteilung und dem Statement übergebenen Parameterausprägungen anpassen können. SQL-

Abbildung 4: CPU-Nutzung durch SQL-Statements im Strobe-Report

Statements können einen Cursor verwenden oder ohne Cursor ausgeführt werden. Diese Information wird ebenfalls im Meta-Modell erfasst.
Die Messung betreffende, allgemeine Informationen werden mittels der Klasse *MeasurementSession* verwaltet. Dazu zählen beispielsweise die Anzahl der genommenen Samples, die Sampling-Rate, die eingesetzten Systeme und deren Version usw. Die Klasse *ModelEntity* generalisiert lediglich die Verwaltung der Subklassen wie z.B. deren Persistierung und trägt keine semantische Bedeutung. Der aus Instanzen dieser Klassen aufgebaute Baum mit Performance-Daten kann traversiert werden, um benötigte Informationen, wie z.B. den aggregierten Anteil der CPU-Nutzung aller *UPDATE* Sql-Statements zu extrahieren. Für die Traversierung bietet sich ein Visitor [GHJV94] an.

3.2 Anti-Pattern Suche

Viele Performance-Probleme spiegeln sich in Form von gewissen Pattern in den Profiling-Reports wider. Diese Pattern können auch automatisch auf Basis der eingelesenen Daten gesucht werden. Einen vergleichbaren Ansatz auf Basis von JEE Anwendungen verfolgt auch [Par08]. Nachfolgend werden zwei Beispiele für Anti-Pattern beschrieben, die in Strobe-Messungen von Cobol Programmen mit DB2 Datenbank - einer typischen Konfiguration für Legacy-Anwendungen auf Mainframes - in produktiven Systemen bereits beobachtet wurden.

3.2.1 Einzelsatz-Fetch mit Cursor

Ein Cursor ist eine Datenstruktur, die verwendet wird, um die Ergebnismenge einer Datenbankabfrage zu durchlaufen. Häufig werden bei Datenbankabfragen, welche die Rückgabe

von höchstens einem Wert sicherstellen, Cursor eingesetzt anstatt unmittelbar den gesuchten Wert abzufragen. In diesem Fall ist der Aufwand zur Erzeugung, Nutzung und Freigabe des Cursors vermeidbar. Die Zusicherung, dass nicht mehr als ein Datensatz in der Ergebnismenge auftritt ist z.B. bei Abfragen auf Primärschlüssel oder bei Nutzung von Aggregatfunktionen ohne GROUP BY Klausel gewährleistet.
In den Performance-Daten erkennt man dieses Muster am Verhältnis von Open- zu Fetch-Ausführungen für ein SQL-Statement:

$$N_{Fetch} \leq N_{Open}, bzw. \frac{N_{Fetch}}{N_{Open}} \leq 1 \quad (1)$$

Diese Formel beschreibt eine notwendige, aber nicht hinreichende Bedingung für einen Einzelsatz-Fetch mit Cursor. Eine weitere Untersuchung des eingesetzten SQL-Statements und ggf. der DDL hat in einem weitern Schritt zu erfolgen. Zum Auffinden möglicher Kandidaten für dieses Anti-Pattern, wird für alle im Datenmodell abgespeicherten SQL-Statements das oben beschriebene Verhältnis berechnet. Liegt dieses zwischen einem konfigurierbaren Schwellwert und 1, so wird das betrachtete Statement als Kandidat für dieses Anti-Pattern betrachtet. Der Schwellwert dient dazu, ggf. auftretende Abfragen, die keine Records liefern, ebenfalls zu akzeptieren (z.B. wenn ein Schlüssel gesucht wird, der nicht exisitert).

3.2.2 Cobol Library

Einige Anweisungen in Cobol veranlassen den Compiler dazu, Routinen aus der Cobol Library aufzurufen. Die Aufrufe der einzelnen Routinen finden sich in den Strobe-Reports wieder. Kann einem solchen Aufruf eine hohe CPU-Nutzung zugeordnet werden, deutet dies auf eine potentielle Optimierungsmöglichkeit durch Ersatz des verursachenden Aufrufs hin.
Beispiel hierfür sind der „variable length move“, welcher verwendet wird, wenn eine überlappende Struktur kopiert wird oder „Inspect“ zur Bearbeitung von Strings. Letzterer Aufruf ist nach [She02] mit einer einfachen Schleife und Substring-Operationen deutlich performanter. In typischerweise auf Mainframes ausgeführten Unternehmensanwendungen werden in der Regel keine Fliesskommaoperationen mit doppelter Genauigkeit benötigt. Auch hierfür erzeugt der Cobol Compiler Bibliotheksaufrufe, die sich leicht durch eine Änderung der verwendeten Datentypen eliminieren lassen.
Abbildung 5 stellt einen Auszug aus einem Strobe-Report eines produktiven Systems dar. Gemessen wurde ein Batch, der 67,39 % CPU-Nutzung alleine für Divisionen und Multiplikationen mit doppelter Genauigkeit verursacht.

Eine Nutzung solcher „verdächtigen“ Bibliotheksroutinen kann auf dem vorgestellten Datenmodell einfach gefunden werden, indem dieses traversiert und nach dem betreffenden Modul (im Beispiel „IGZCPAC“), ggf. auch nach einer bestimmten Section gesucht wird. Ein Abgleich der dadurch verursachten CPU-Nutzung mit einem Schwellwert kann dieses Pattern weiter einschränken und nur bei signifikanter Auswirkung entsprechend negativ klassifizieren.

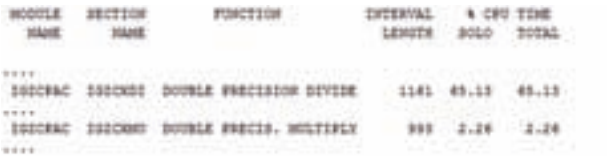


Abbildung 5: Intensive Nutzung der Cobol Library im Strobe-Report

4 Zusammenfassung und Ausblick

Die alleinige Auswertung der Informationen aus den Strobe-Reports bringt bereits einige potentielle Probleme ans Licht. [She02] führt noch weitere Anti-Pattern auf, die mit dem beschriebenen System gesucht werden können. Die Existenz einzelner SQL-Statements, die pro Ausführung viel CPU-Nutzung relativ zu anderen Statements erzeugen, ist ebenfalls eine intessante Fragestellung, die damit beantwortet werden könnte. Eine aufbereitete Darstellung für Performanceexperten könnte eine explorative Analyse (Drilldown etc.) der Daten ermöglichen. Die Modulbezeichnungen betrieblicher Anwendungen unterliegen häufig gewissen Namensschemata, welche bestimmte Funktionalitäten gruppieren. Eine Aggregation an Hand dieses anwendungsspezifischen Namensschemas liefert eine weitere mögliche Sicht auf die Daten. Eine Anreicherung bzw. Verknüpfung der vorliegenden Informationen mit weiteren Daten könnte weitergehende Performance-Analysen ermöglichen. Denkbar wäre eine Verbindung mit einer DDL-Datei für Datenbankzugriffe. Hier könnte man beispielsweise bei Einzelsatz-Fetch mittels Cursor prüfen, ob die Selektion tatsächlich auf Schlüsselattribute erfolgt. Ebenso könnten die oben beschriebenen Informationen verwendet werden, um statische Analysverfahren zu optimieren, indem Gewichtungen für die Kosten aus den Ausführungszeiten einzelner Module oder Programmabschnitte abgeleitet werden.

Literatur

[Ern03] Michael D. Ernst. Static and dynamic analysis: synergy and duality. *Proceedings of the ICSE Workshop on Dynamic Analysis (WODA 2003)*, Seiten 25–28, 2003.

[GHJV94] Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides. *Design Patterns - Elements of Reusable Object-Oriented Software*, Jgg. 1. Addison-Wesley Longman, Oktober 1994.

[Net04] Nicholas Nethercote. *Dynamic Binary Analysis and Instrumentation*. Dissertation, University of Cambridge, 11 2004.

[Par08] John Parson, Trevor Murphy. Detecting Performance Antipatterns in Component Based Enterprise Systems. *Journal of Object Technology*, 7(3):55–90, März 2008. Sucht Anti-Patterns in JEE Anwendungen - Profiling mit Ïnstrumentierung"(Proxy).

[She02] Tony Shediak. Performance Tuning Mainframe Applications "Without trying too hard". 2002.

Parallel Execution of kNN-Queries on in-memory *K*-D Trees

Tim Hering
Faculty of Computer Science
University Otto-von-Guericke
Universitätsplatz 2
39106 Magdeburg
thering@st.ovgu.de

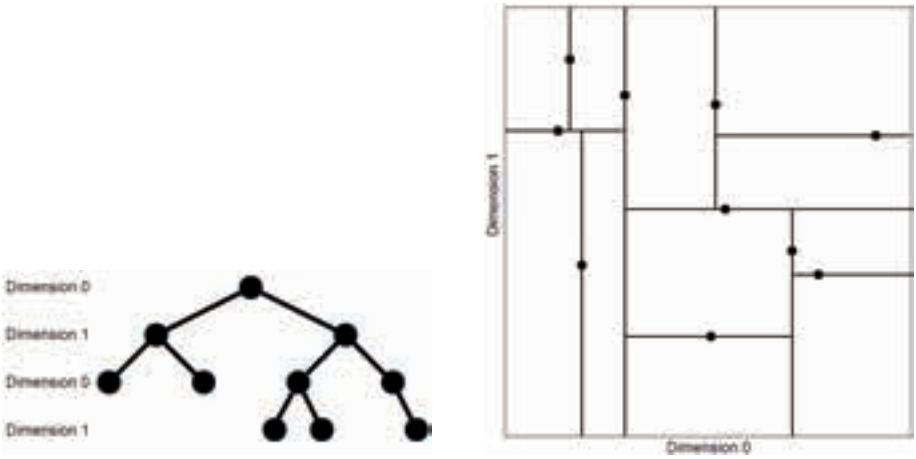
Abstract: Parallel algorithms for main memory databases become an increasingly interesting topic as the amount of main memory and the number of CPU cores in computer systems increase. This paper suggests a method for parallelizing the *k*-d tree and its kNN search algorithm as well as suggesting optimizations. In empirical tests, the resulting modified *k*-d tree outperforms both the *k*-d tree and a parallelized sequential search for medium dimensionality data (6-13 dimensions).

1 Introduction

Multidimensional datasets are a collections of data points with multiple dimensions. They are produced by a number of applications, one of the most cited of which are multimedia databases [BBK01]. One of the characteristic requirements to storage solutions for multidimensional datasets is the ability to process similarity-based queries fast. One class of those queries are kNN-queries. They retrieve the k points in the dataset that are closest to a given query point. Notable index structures that are designed for multidimensional databases include the R-Tree [Gut84], the Pyramid Technique [BBK98], the VA-File [WB97] and Locality Sensitive Hashing [IM98].

Two developments in computer hardware are starting to change the possibilities of and requirements for index structures. Firstly, the amount of main memory is increasing. As a result, more database tasks can be executed in main memory. Secondly, most computers possess multi-core processors or graphic cards. Both can be used to conduct parallel computations. They reduce the processing time of algorithms, provided they run on main memory data. If disk accesses need to be performed, the speed-up becomes less relevant. The topic of parallelizing multidimensional index structures on main memory databases is largely unexplored. Most attempts at parallelization focus on multiple disks or systems instead of multiple cores (as discussed in section 3.2).

This paper explores the possibility of using *k*-d trees for multidimensional kNN-queries [Ben75]. *K*-d trees are binary trees that are suitable for multidimensional data and were designed for main memory applications. Each level of the tree splits according to the dimension after the one of the previous level. As a consequence, the data space is divided according to a brick wall pattern (Figure 1).



(a) Tree representation (b) Partitioning of the data space

Figure 1: A 2-d tree containing 10 data points

An approach for utilizing multi-core processors to speed up the search on *k*-d trees is subject of this paper. Additionally, several improvements to the resulting index structure are described. They focus on lessening the impact of the disadvantages of parallelization, primarily the need for inter-thread communication. Lastly, suitable parameters for these modifications are determined using empirical tests.

2 The *k*-d tree

The *k*-d tree is an index structure for multidimensional data that was introduced in 1975 by Jon Louis Bentley [Ben75]. At its core, it is a binary tree, that can efficiently store data points with more than one dimension by cycling through those dimensions as an algorithm descends down the tree (Figure 1). This means that each level l of the tree has a designated dimension $d_l = l \bmod d_{max}$, where d_{max} is the dimensionality of the data. If an algorithm tries to compare the query point to the discriminator point in a node at level l, it uses the values of the points in d_l for comparison. The lower child of that node and all its descendants have a lower value than the discriminator point regarding d_l, whereas the higher child and its descendants have higher values. As a consequence, all dimensions receive an approximately equal treatment while the logarithmic running time for insert operations and point queries of binary trees is maintained.

The kNN search algorithm presented in [FBF77] traverses the *k*-d tree recursively. For a given node, the search function executes the following steps: (1) If its point is closer to the query point than any other within a global list if the k best result candidates, update the list with that point. (2) If the node is not a leaf node, the function is recursively called with the child node closer to the query point. (3) The minimum possible distance between the

query point and the other, more distant child node is calculated. If this distance is larger than the maximum distance for a candidate (calculated from the result list), no point in that node or its descendants can be a result candidate. Otherwise, the search function is called with that node.

The algorithm is initiated by calling the search function with the root node. After control returns to the caller, the result list contains the k nearest neighbours to the query point.

3 Related work

This section covers four existing categories of index structures that are related to the content of this paper: (1) index structures that are explicitly designed for main memory applications, attempting to optimize the usage of caches, (2) parallel index structures that are not designed for main memory databases, (3) parallel *k*-d trees designed to run on GPUs and (4) parallel index structures for main memory databases that are not designed for kNN-queries.

3.1 Cache sensitive index structures for main memory databases

There is a number of index structures for main memory databases that rely on the optimization of cache usage. Cache memories are small, but fast buffers for data blocks in the main memory [Smi82]. Being conscious of these buffers can speed the access times of searches up.

Among the index structures following this idea are Cache-Sensitive Search Trees (CSS-trees) [RR98]. They take advantage of the cache size by possessing leaf nodes that are as large as a block in the cache. Another, similar example is the Cache Sensitive B^+-Tree (CSB^+-Tree) [RR00]. It is a variant of the B^+-Tree that reduces the size of its nodes by having less or no child pointers. Children are instead identified by their address offsets.

3.2 Parallel index structures

As parallelization is a commonly used method to speed up computations, various index structures have employed it. In contrast to the parallel *k*-d tree, they tend to focus on parallel disks or parallel systems instead of multiple threads. An example of the former are Parallel R-Trees [KF92]. They are a class of variants of R-Trees, whose nodes are spread over multiple disks to improve query throughput. Additionally, attempts at porting them to an index structure on parallel systems (spreading nodes over multiple computers) have been made [AQSK98].

3.3 Parallel *k*-d trees for GPUs

There are various existing papers on parallel *k*-d trees for GPUs. They mostly cover parallel inserts, as fast insert operations are required for 3d rendering [ZHWG08, SSK07]. Examinations on parallel queries are less common, but existing [GDB08]. GPUs have significantly more cores than CPUs, but they also have disadvantages in regards to the execution of queries on *k*-d trees: Firstly, they are less suited for control flow. This means that jumps within the code, like those created by if-statements, execute slower. Secondly, data and queries have to be loaded into the GPU memory. After the execution of the query the results have to be returned. Both processes create an overhead that is avoidable by using the CPU. Additionally, not all computers are equipped with a GPU, especially servers that are remote controlled. This makes GPU solutions less universally applicable than CPU solutions. Nevertheless, they are an interesting alternative, if only for the increase in computational power that can be gained.

3.4 Parallel in-memory index structures for other query types

Parallel processing has been utilized by index structures that are designed to answer queries other than kNN-queries. One of them is FAST (Fast Architecture Sensitive Tree), a tree-based index structure built with the characteristics of modern hardware in mind [KCS+10]. The parts of the index are compressed and divided to specifically reduce the limiting influence of memory bandwidth. Another example is the KISS-Tree, which is a prefix tree of 3 levels [KSHL12]. The levels deploy different addressing techniques that are tailored to the requirements of each respective level, allowing the KISS-Tree to reach a performance similar to that of hash-based indexes.

4 Modifications to the *k*-d tree

To utilize multi-threading, the kNN search algorithm of the *k*-d tree has to be changed. The basic idea is to introduce a queue that holds all nodes which are yet to be visited. There is a fixed number of threads that work independently and communicate over that queue. In turn, threads retrieve the head of the queue and evaluate it. If one or both of its children are candidates for the query results, they are inserted at the back of the queue (see Figure 2 for the full algorithm). A node is a candidate if the minimum distance of the query point to its bounding rectangle is smaller than the k-th nearest processed neighbour [FBF77].

Communication between threads has the tendency to decrease performance of multi-threaded applications significantly. As one thread uses the communication resource, it is locked and other threads have to wait for it to finish. To avoid introducing the list of intermediate results as a bottleneck (its last entry is needed to determine if a node is a candidate), each thread has its own set of intermediate results. After all threads have finished, their result lists are merged to get the global answer. This measure reduces the speed with which the

```
01  knnQuery(queryVector) {
02    while (not nodeQueues.allEmpty) {
03      node = pollNodeQueues()
04      if (not node.isLeafNode) {
05        if (isCandidate(node.leftChild, queryVector)) {
06          addToNodeQueue(node.leftChild)
07        }
08        if (isCandidate(node.rightChild, queryVector)) {
09          addToNodeQueue(node.rightChild)
10        }
11      } else {
12        addToResultList(node)
13      }
14    }
15    return resultList
16  }

17  node pollNodeQueues() {
18    int rand = random(0, numberOfNodeQueues - 1)
19    if (not nodeQueues.get(rand).isEmpty) {
20      return nodeQueues.get(rand).poll()
21    } else {
22      for (int i from 0 to numberOfNodeQueues - 1) {
23        if (not nodeQueues.get(i).isEmpty) {
24          return nodeQueues.get(i).poll()
25        }
26      }
27    }
28    return lowPriorityQueue.poll()
29  }
```

Figure 2: Pseudo code for the modified kNN search algorithm.

algorithm converges, but experiments showed an overall increase in performance.

The remaining bottleneck is the queue of nodes. It is critical for inter-thread communication and thus cannot be easily removed, but the impact of the delays can be reduced by splitting the queue. Whenever a thread would execute an operation on the queue, it instead selects one of an array of queues at random. The probability for a thread to find a queue in use is almost inversely proportional to the number of queues employed. However, since all queues have to be checked to determine if the query is finished, it is not optimal to maximize the number of queues.

Furthermore, an additional 'low priority queue' is introduced. Whenever a node is a potential, but unlikely candidate for the results, it is inserted into this queue instead of the others. The heuristic for an unlikely candidate is one of two conditions: (1) its sibling node is known to be no candidate, i.e. the node in question is at the border of the region that has to be considered or (2) its sibling node has a lower minimum distance to the query point. When a thread retrieves a node from the queues, it first tries to poll a random queue. If that queue is empty, it searches all queues in order, ending with the low priority queue. This ensures that all normal queues are empty, before the unlikely candidates are considered.

While the search algorithm descends the tree, the minimum distance that a node has to have to be a candidate for the results converges. At the same time, the regions that can be excluded from the search (based on their minimum distance to the query point) get smaller. If only small regions of the data space can be excluded, these comparisons become inefficient and - eventually - an overhead. It is thus advantageous to fall back to a sequential search at a certain depth. The optimal solution is to perform this switch at the break-even-point when the cost to descend a node and its descendants becomes higher than the cost of comparing all points contained in those node sequentially. To support this approach, points are only stored in leaf nodes. Whenever a node is split during the creation of the tree, its points are equally distributed to both children. The split value is the median of the values of all points within the node in the dimension of the split. Side effects are a lower risk of degeneration and a simplified search algorithm (Figure 2).

The resulting index structure is similar to the K-D-B-Tree [Rob81]. The main difference is the size of its nodes - leaf nodes contain up to a user-defined number of points, whereas all other nodes have exactly two children. Additionally, the motivation for having nodes contain more than two points is not the usage of virtual memory, but rather avoiding a search overhead. The K-D-B-Tree was not designed to perform well in a main memory database and is consequently not used as a reference point for evaluation the modified *k*-d tree.

5 Empirical parameter optimization

The modifications to the *k*-d tree suggested in this paper require two parameters that were introduced previously: (1) the number of queues and (2) the size of leaf nodes. Their optimum values are dependent on the specific application. This test generates a set of default values that can be used if optimization is not possible or not feasible in consideration of the expected gains.

The experiments are conducted on random, equally distributed data sets with independent dimensions. They are 'neutral' in the sense that they contain neither correlations not clusters. Yet that feature does at the same time separate them from most real data sets.

For the tests, datasets of 100,000 points were generated. Each point had 10 dimensions. Both values were chosen as an average value, being neither particularly high nor low for a multidimensional database. Query points were generated the same way as the data points. They were used as query parameters for 5-NN-queries that were conducted in batches of 1,000 until the size of the confidence interval was smaller than a given threshold. This threshold was chosen to be 0.02 times the size of the mean value, with a confidence interval calculated using $\alpha = 0.05$.

The experiments were executed in 4 threads on a 4-core CPU. The time difference between starting the query and receiving the results was measured, that is to say the optimization target was the mean execution time.

For the number of queues, the optimum value was found to be around 2 to 3, plus the low priority queue (Figure 3). The results are also clearly confirm the advantage of the

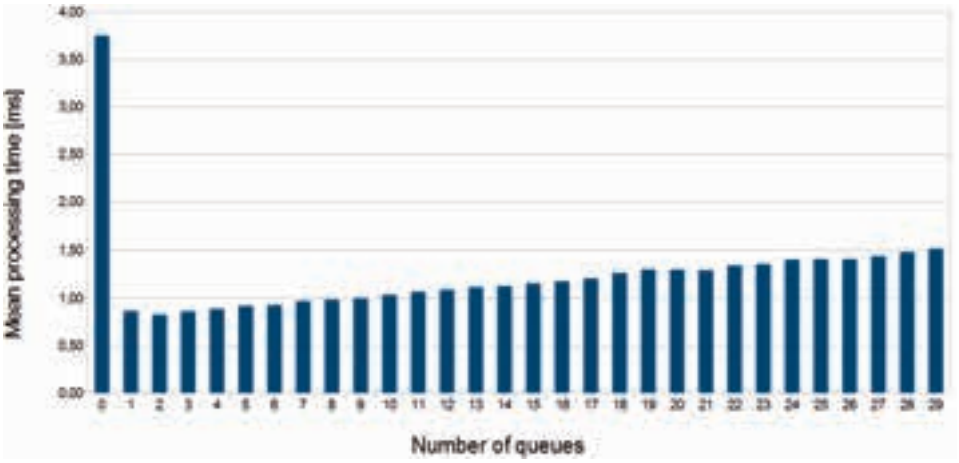


Figure 3: The influence of the number of queues on the performance. Numbers do not include the low priority queue, i.e. the case '0' uses only the low priority queue.

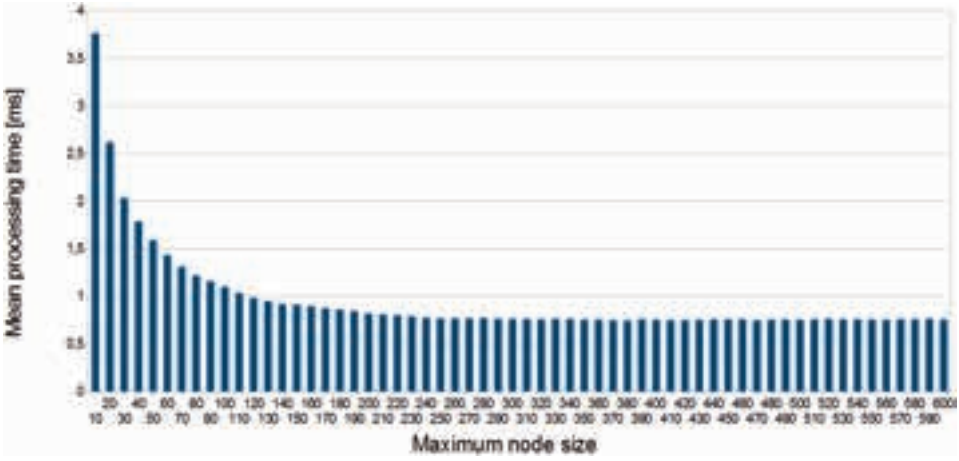


Figure 4: The influence of maximum size of a node on the performance.

separation into high and low priority queues, as the execution time with only one queue total is significantly worse than otherwise.

The optimum node size, which is to say the optimum switching point from descending the tree to a sequential search, was found to be at about 280 points (Figure 4). Consequently, a *k*-d tree for this scenario should have leaf nodes that contain 280 points each.

6 Evaluation

For evaluation purposes, the modified *k*-d tree is tested in comparison to the original *k*-d tree as well as a parallel sequential search. The testing process is the same as the one used in section 5. To find out how the index performances behave with different numbers of dimensions, dimensionality was varied from 1 to 20.

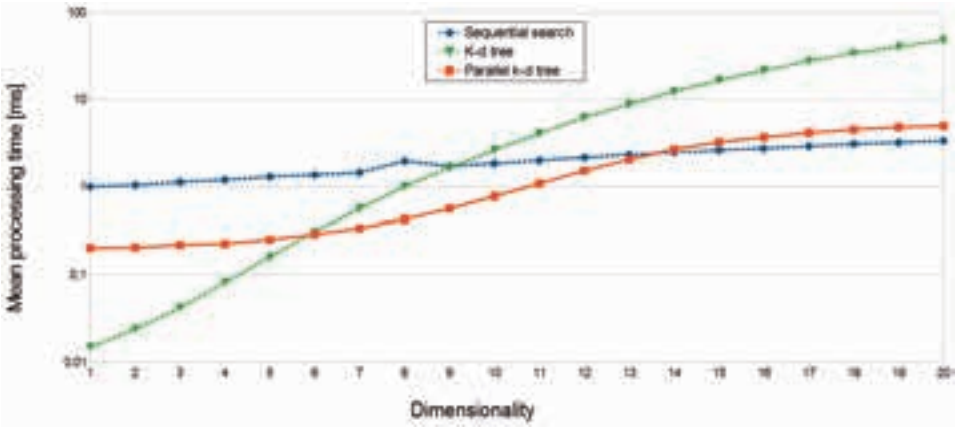


Figure 5: Performance of the performance of sequential search, *k*-d tree and modified *k*-d tree on datasets of varying dimensionality.

The results (Figure 5) show the dependency of the relative performances on the dimensionality. For low dimensionalities, the *k*-d tree is fastest, as it does not have as much of an overhead as the modified version. In the high dimensionality area, the kNN search algorithm for both versions of the *k*-d tree degenerate into a sequential search with an overhead. Nevertheless, the modified *k*-d stays relatively close to the performance of the sequential search. Between both areas, from 6 to 13 dimensions, the modified *k*-d tree is the fastest of the tested algorithms by a factor of up to two. It should also be noted that it is at least the second-best solution across the board, making it the stable choice for an unknown environment.

7 Conclusion and future work

A parallelized *k*-d tree implementing the modifications suggested in this paper is a powerful solution to the kNN problem in medium dimensionality data spaces (6-13 dimensions). Additionally, it degenerates slower than the *k*-d tree as the number of dimensions increases. This property makes it useful for unknown environments or 'one size fits all' solutions, where index structures might have to cope with unexpectedly high dimensionalities.

Moving forward, performance tests and parameter optimizations for real datasets are a short term goal. Farther out, the approach for parallelization implemented here can be transferred to other index structures. These could also include suitable disk-based index structures. Lastly, the utilization of GPUs - potentially in parallel to the CPU - is a tempting goal in spite of the difficulties.

Acknowledgements

The work in this paper has been partially funded by the German Federal Ministry of Education and Science (BMBF) through the Research Program under Contract No. FKZ:13N10817.

References

[AQSK98] Ning An, Liujian Qian, Anand Sivasubramaniam, and Tom Keefe. Evaluating parallel R-tree implementations on a network of workstations (an extended abstract). In *Proceedings of the 6th ACM international symposium on Advances in geographic information systems*, GIS '98, pages 159–160, New York, NY, USA, 1998. ACM.

[BBK98] Stefan Berchtold, Christian Böhm, and Hans-Peter Kriegal. The pyramid-technique: towards breaking the curse of dimensionality. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, SIGMOD '98, pages 142–153, New York, NY, USA, 1998. ACM.

[BBK01] Christian Böhm, Stefan Berchtold, and Daniel A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv.*, 33(3):322–373, September 2001.

[Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975.

[FBF77] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Trans. Math. Softw.*, 3(3):209–226, September 1977.

[GDB08] Vincent Garcia, Eric Debreuve, and Michel Barlaud. Fast k nearest neighbor search using GPU. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1 –6, June 2008.

[Gut84] Antonin Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the ACM Special Interest Group on Management of Data International Conference*, pages 47–57. ACM, 1984.

[IM98] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the 30th annual ACM symposium on Theory of Computing*, pages 604–613. ACM, 1998.

[KCS+10] Changkyu Kim, Jatin Chhugani, Nadathur Satish, Eric Sedlar, Anthony D. Nguyen, Tim Kaldewey, Victor W. Lee, Scott A. Brandt, and Pradeep Dubey. FAST: fast architecture sensitive tree search on modern CPUs and GPUs. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 339–350, New York, NY, USA, 2010. ACM.

[KF92] Ibrahim Kamel and Christos Faloutsos. Parallel R-trees. *SIGMOD Rec.*, 21(2):195–204, June 1992.

[KSHL12] Thomas Kissinger, Benjamin Schlegel, Dirk Habich, and Wolfgang Lehner. KISS-Tree: smart latch-free in-memory indexing on modern architectures. In *Proceedings of the Eighth International Workshop on Data Management on New Hardware*, DaMoN '12, pages 16–23, New York, NY, USA, 2012. ACM.

[Rob81] John T. Robinson. The K-D-B-tree: a search structure for large multidimensional dynamic indexes. In *Proceedings of the 1981 ACM SIGMOD international conference on Management of data*, SIGMOD '81, pages 10–18, New York, NY, USA, 1981. ACM.

[RR98] Jun Rao and Kenneth A. Ross. Cache conscious indexing for decision-support in main memory. 1998.

[RR00] Jun Rao and Kenneth A. Ross. Making B+- trees cache conscious in main memory. *SIGMOD Rec.*, 29(2):475–486, May 2000.

[Smi82] Alan Jay Smith. Cache Memories. *ACM Computing Surveys*, 14(3):473–530, September 1982.

[SSK07] Maxim Shevtsov, Alexei Soupikov, and Alexander Kapustin. Highly Parallel Fast KD-tree Construction for Interactive Ray Tracing of Dynamic Scenes. In *Computer Graphics Forum*, volume 26, pages 395–404. Wiley Online Library, 2007.

[WB97] Roger Weber and Stephen Blott. An approximation based data structure for similarity search. *Report TR1997b, ETH Zentrum, Zurich, Switzerland*, 1997.

[ZHWG08] Kun Zhou, Qiming Hou, Rui Wang, and Baining Guo. Real-time KD-tree construction on graphics hardware. *ACM Transactions on Graphics*, 27(5):126:1–126:11, December 2008.

Eine effiziente Indexstruktur für dynamische hierarchische Daten

Robert Brunel · Jan Finis

Technische Universität München
Institut für Informatik
Lst. III: Datenbanksysteme
Boltzmannstr. 3
85748 Garching
vorname.nachname@in.tum.de

Abstract: Bis heute fällt es relationalen Datenbanksystemen schwer, große Mengen dynamischer hierarchischer Daten effizient zu verwalten, obwohl dies eine ständig wiederkehrende Anforderung in fast allen betrieblichen Informationssystemen ist. In dieser Arbeit stellen wir eine Datenstruktur vor, die beliebig geformte Hierarchien derart indexiert, dass strukturbezogene Anfragen beschleunigt werden, während gleichzeitig strukturelle Änderungen wie das Einfügen und Löschen von Knoten in logarithmischer Zeit möglich sind. Wir verfolgen damit langfristig das Ziel, in relationalen Datenbanken künftig Hierarchien als native, den Relationen praktisch gleichgestellte Datenobjekte zu unterstützen.

1 Einleitung

Das Ablegen hierarchischer Daten gehört zu den ältesten Anforderungen von Geschäftsanwendungen an Datenbanksysteme überhaupt. Betriebliche Informationssysteme nutzen Hierarchien etwa zur Verwaltung personeller und geografischer Organisationsstrukturen, in Projektplanungsprozessen, zur hierarchischen Untergliederung von Bauplänen (so genannte Bills of Materials), oder auch im Bereich der Business Intelligence, wo man die zu analysierenden Geschäftsdaten mittels „Slice and Dice“ entlang so genannter Dimensionshierarchien filtrieren und auf die gewünschte Detailstufe reduzieren möchte. Seit relationale Datenbanksysteme (RDBMS) die hierarchischen Datenbanken wie IBM IMS weitestgehend abgelöst haben, sind sie dafür kritisiert worden, hierarchische Daten nicht adäquat abbilden zu können. Diese Tatsache ist auf die inhärent flache Natur des relationalen Modells zurückzuführen – ein klassischer „Impedance Mismatch“. In den vergangenen 35 Jahren hat sich daran erstaunlich wenig geändert: Während objektorientierte, XML-, Graph- und mehrdimensionale Datenbanksysteme gewisse Formen von Hierarchien als primäre Datenobjekte nativ unterstützen, stehen den Benutzern handelsüblicher RDBMS wenn überhaupt nur sehr beschränkte SQL-Erweiterungen zu Verfügung – beispielsweise rekursive `WITH`-Ausdrücke oder das `CONNECT-BY`-Konstrukt [Hau11, Ora10] –, die weder den benötigten Funktionsumfang abdecken noch eine für uns akzeptable Performanz haben.

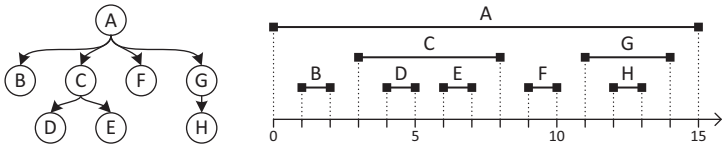


Abbildung 1: Eine Beispielhierarchie und ihre Kodierung durch geschachtelte Intervalle.

Im Rahmen unseres Forschungsprojektes suchen wir nach Möglichkeiten, relationale Datenbanksysteme um spezielle Indexstrukturen für Hierarchien so zu erweitern, dass sie die typischen Anwendungsfälle von Geschäftsanwendungen weitestgehend abdecken. Zum einen konzentrieren wir uns dabei auf sehr *große* Hierarchien mit Knotenzahlen im zweistelligen Millionenbereich oder noch höher. Auch degenerierte, z. B. sehr flache und breite oder sehr tiefe Strukturen sollen die Anfrageauswertung möglichst nicht beeinträchtigen. Zum anderen gehen wir von stark *dynamischen* Datensätzen aus. Die wesentlichen auf Hierarchien denkbaren Änderungsoperationen „Knoten einfügen“, „Knoten entfernen“ und „Teilbaum verschieben“ möchten wir in hohen Raten von mindestens einigen zehntausend pro Sekunde unterstützen. Anders als manche vergleichbare Ansätze zur Indexierung von XML-Daten (vgl. Abschnitt 5) bauen wir dabei nicht auf dem relationalen Schema auf, sondern streben eine Erweiterung des Datenbankkerns an, was uns zusätzlichen Spielraum gibt, den Speicherplatzverbrauch zu minimieren und gleichzeitig die Performanz zu maximieren.

Im Folgenden stellen wir eine in diesem Zusammenhang entwickelte Indexstruktur für dynamische hierarchische Daten vor, die allgemeine Baumstrukturen[1] unterstützt und von so genannten Labeling Schemes aus dem XML-Umfeld inspiriert ist. Sie benötigt für Hierarchien mit n Knoten Speicherplatz in der Größenordnung $\mathcal{O}(n)$, unterstützt die genannten Änderungsoperationen in $\mathcal{O}(\log n)$, und erlaubt gleichzeitig diverse Anfragen in $\mathcal{O}(\log n)$ auszuwerten.

2 Kodierung von Hierarchien durch geschachtelte Intervalle

Unser Ansatz ist inspiriert von der bekannten Kodierung einer Baumstruktur durch *geschachtelte Intervalle*. Die Baumkanten werden dabei nicht explizit abgespeichert; stattdessen wird jedem Knoten v ein Intervall $[v.l, v.u]$ zugeordnet, wobei $v.l$ und $v.u$ Ganzzahlen sind und jeweils $[v.l, v.u]$ die Intervalle der direkten und indirekten Kindknoten von v einschließt. (Wir benutzen die Bezeichnungen l und u für „lower“ und „upper“). Im relationalen Schema würden wir die Intervalle in einer Tabelle mit drei Spalten node-id, l und u ablegen. Die Kantenstruktur ist durch die Knotenmarken vollständig kodiert, weshalb dieses und verwandte Verfahren auch unter dem Begriff *Labeling Schemes* bekannt sind. Abbildung 1 zeigt eine Beispielkodierung.

[1] Genauer: *rooted*, *ordered*, *labeled* forests, d. h. Wälder von Bäumen mit jeweils eindeutigem Wurzelknoten und eindeutig festgelegter Reihenfolge von Geschwisterknoten, wobei zusätzlich die Knoten mit Marken versehen werden können.

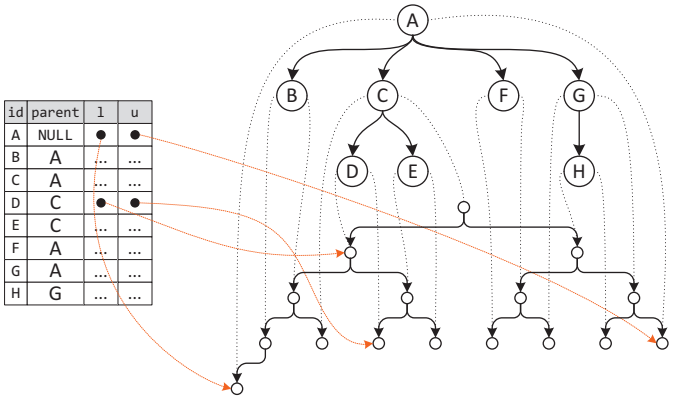


Abbildung 2: Die Beispielhierarchie mit zugehöriger Order-Tree-Repräsentation.

Der größte Vorteil der Intervallkodierung liegt darin, dass man viele Anfragen in $\mathcal{O}(1)$ auswerten kann, indem man die Marken der beteiligten Knoten betrachtet. Speziell für die aus dem XPath-Umfeld bekannten *Axis Checks* entlang verschiedener Achsen der Hierarchie (Vor-/Nachfahren, Cousins etc.) ist dies besonders nützlich, etwa:

$$\text{is-ancestor}(v, w) := v.l < w.l \wedge w.u < v.u$$

Gegenüber Änderungen ist die Kodierung andererseits sehr unflexibel: Möchte man einen neuen Blattknoten an der Stelle l einfügen, muss man alle Intervallgrenzen, die größer oder gleich l sind, um 2 erhöhen, um eine Lücke für das neue Intervall $[l, l + 1]$ zu schaffen. Im Durchschnitt müssen $n/2$ Intervallgrenzen angepasst werden – Einfügen hat somit die Komplexität $\mathcal{O}(n)$. Verschiedene Vorschläge, dieses Problem zu umgehen (vgl. Abschnitt 5), etwa durch Vorreservieren von Lücken oder durch Verwendung von Gleitkommazahlen, können an der Tatsache, dass früher oder später eine Neunummerierung von $\mathcal{O}(n)$ Knoten notwendig wird, nichts ändern.

3 Indexierung hierarchischer Daten durch Order Trees

3.1 Das Konzept des Order Trees

Unsere Idee ist nun, das Problem der Neunummerierungen zu beseitigen, indem wir auf die numerischen Intervallgrenzen ganz verzichten. Dies basiert auf der Beobachtung, dass die absoluten Zahlen für die Beantwortung von Anfragen irrelevant sind; alleine die relative Anordnung der Intervallgrenzen, d. h. die entsprechende Ordnungsrelation $<$, ist entscheidend. Um Änderungen effizient zu unterstützen, kodieren wir die Ordnungsrelation in einer dynamischen, balancierten Baumstruktur, die wir **Order Tree** nennen. Abbildung 2 illustriert deren Aufbau: Jedem Knoten der Hierarchie (oben) werden zwei Intervallgrenzen zugeordnet, denen wiederum zwei Baumknoten im Order Tree (unten) entsprechen. Die Ordnungsrelation auf den Intervallgrenzen ergibt sich aus der Anordnung der Einträge

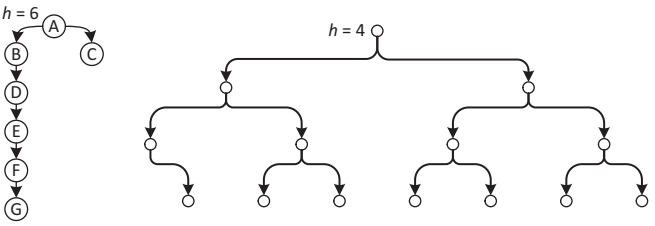


Abbildung 3: Eine degenerierte Hierarchie und die balancierte Struktur des zugehörigen Order Trees.

innerhalb des Order Trees: Eine Intervallgrenze b_1 ist *kleiner* als eine Grenze b_2, wenn sich der entsprechende Order-Tree-Eintrag e_1 *links* vom Eintrag e_2 befindet. Den Datensätzen der zugehörigen Knotentabelle (links) ordnen wir nun, statt numerischer Intervallgrenzen, Zeiger auf die Order-Tree-Einträge zu. Beim ersten Zugriff zur Laufzeit wird der Order Tree aus den Informationen in der Tabelle, insbesondere der parent-Spalte, aufgebaut. Da der Order Tree die hierarchische Struktur vollständig kodiert, können wir fortan alle strukturbezogenen Anfragen und Änderungsoperationen rein auf dem Order Tree durchführen und brauchen die Hierarchie selbst nicht explizit zu materialisieren.

Ein Order Tree R soll folgende Operationen in jeweils $\mathcal{O}(\log n)$ unterstützen:

R.**compare**(e_1, e_2) vergleicht zwei Einträge bezüglich der Ordnungsrelation $<$.

$e' \leftarrow R$.**insert**(before e) fügt einen neuen Eintrag e' unmittelbar vor e hinzu. "before" bezieht sich hierbei auf die Reihenfolge gemäß $<$.

R.**remove**(e) entfernt einen Eintrag e.

R.**move-range**$([e_1, e_2],$ before $e)$ verschiebt alle Einträge im Bereich $[e_1, e_2]$ unmittelbar vor e, vorausgesetzt, dass $e_1 < e_2$ ist und e sich nicht in $[e_1, e_2]$ befindet.

Um das abstrakte Konzept des Order Trees umzusetzen, kann man prinzipiell jede Baumstruktur einsetzen, die folgende Eigenschaften erfüllt: (1.) Im Baum ist die relative Reihenfolge seiner Einträge kodiert; sie bleibt insbesondere unter Balancierungsoperationen erhalten. (2.) Die genannten Operationen lassen sich auf der Baumstruktur mit einer Komplexität von $\mathcal{O}(\log n)$ realisieren. Um das zu garantieren, wird der Baum balanciert gehalten. (3.) Zeiger auf Einträge sind „stabil", werden also nicht durch strukturelle Änderungen invalidiert. — Suchbaumstrukturen wie der AVL- oder der Rot-Schwarz-Baum [CLRS09] bieten sich unmittelbar an, wobei wir auf einen Großteil der üblichen Funktionalität eines Suchbaums verzichten. Insbesondere speichert der Order Tree *keine* Schlüssel in Knoten: alle genannten Operationen beziehen sich auf bereits gegebene Einträge (e, e_1, bzw. e_2); wir müssen also nie nach einem bestimmten Schlüssel suchen.

Eigenschaft 2 garantiert uns ein gutartiges Verhalten auch im schlimmsten Fall – unabhängig davon, wie stark die Struktur der Hierarchie degeneriert sein mag (siehe Abb. 3). Die Komplexität der Algorithmen zum Einfügen und Löschen ist in der Regel $\mathcal{O}(h)$, wobei h die Baumhöhe ist. Da der Order Tree einer Hierarchie mit n Knoten $2n$ Einträge enthält und balanciert ist, ist die Komplexität folglich $\mathcal{O}(\log n)$.

Eigenschaft 3 ist nötig, damit wir in die Hierarchietabelle Zeiger auf die zugehörigen Order-Tree-Einträge verwalten können. Die Implementierung darf also Einträge nicht nach

Belieben im Arbeitsspeicher verschieben. Bei zeigerbasierten Strukturen wie dem AVL- oder Rot-Schwarz-Baum ist das problemlos möglich; bei Array-basierten Strukturen wie dem B-Baum lassen sich stabile Zeiger aber i. d. R. nur erreichen, indem man eine Indirektionsstufe, nämlich eine Zuordnung von (stabilen) Referenzen auf Order-Tree-Einträge zu Speicherbereichen, einführt und bei Änderungsoperationen entsprechend aktualisiert.

3.2 Implementierung der Order-Tree-Operationen

Bei **insert** und **remove** handelt es sich um einfache Varianten der bekannten Suchbaumoperationen, wobei es nicht nötig ist, den referenzierten Eintrag e anhand seines Schlüssels zu suchen, da e als Argument bereits gegeben ist. Die Aufgabe besteht also im Wesentlichen nur noch darin, mittels Rotationen die Balance des Baumes wiederherzustellen.

compare ist eine Operation, die in Suchbäumen üblicherweise nicht unterstützt und auch gar nicht benötigt wird. Wir können sie implementieren, indem wir von den beiden gegebenen Einträgen aus die Elternzeiger verfolgen, bis wir bei der Wurzel ankommen. Anschließend können wir durch Vergleichen der beiden Pfade zur Wurzel feststellen, welcher Eintrag „weiter links“ im Baum liegt. Die Baumknoten müssen dazu Zeiger auf ihre jeweiligen Elternknoten speichern, was sowohl beim AVL- als auch beim Rot-Schwarz-Baum ohnehin üblich ist, nicht jedoch beim B-Baum. Im Falle des B-Baums ist der Zusatzaufwand für die Verwaltung der Elternzeiger aber vernachlässigbar.

Wenn wir zusätzlich in jedem Order-Tree-Knoten die Teilbaumhöhe – d. h., den Abstand zum tiefsten Kind im Teilbaum – speichern, können wir compare dadurch beschleunigen, dass wir nicht bis zur Wurzel nach oben laufen, sondern nur bis zum tiefsten gemeinsamen Vorfahren (*least common ancestor*, LCA) der beiden gegebenen Einträge: Wir laufen zunächst von demjenigen Eintrag aus nach oben, der die geringere Teilbaumhöhe hat, da dieser offenbar weiter vom LCA entfernt ist. Sobald die Teilbaumhöhe der Einträge übereinstimmt, laufen wir von beiden aus simultan nach oben, bis wir beim LCA ankommen. Nun lässt sich leicht feststellen, welcher der beiden Pfade „weiter links“ liegt. Zur Speicherung von Teilbaumhöhen bis zu 256 genügt ein Byte. Sowohl gegenüber den dann überflüssigen Balance-Bits beim AVL- oder Rot-Schwarz-Baum als auch gegenüber der Größe eines B-Baum-Knotens ist dieser zusätzliche Speicheraufwand vernachlässigbar.

move-range implementieren wir dadurch, dass wir sie auf die beiden Operationen split und join auf balancierten Bäumen zurückführen:

- $(T_1, T_2) \leftarrow T.\textbf{split}(\text{before } e)$ teilt den Baum T links vom Eintrag e in zwei Teilbäume auf: T_1 enthält die Einträge, die kleiner sind als e, T_2 alle anderen (einschließlich e). Die relative Anordnung der Einträge bleibt dabei erhalten. Außerdem sind sowohl T_1 als auch T_2 balanciert.
- $T \leftarrow \textbf{join}(T_1, T_2)$ konkateniert T_1 und T_2 unter Beibehaltung der relativen Anordnung der Einträge und so, dass der resultierende Baum T balanciert ist.

Sowohl split als auch join sind recht unbekannte Operationen, es gibt aber Algorithmen in $\mathcal{O}(\log n)$ für die meisten verbreiteten Suchbäume. Mit ihrer Hilfe können wir move-

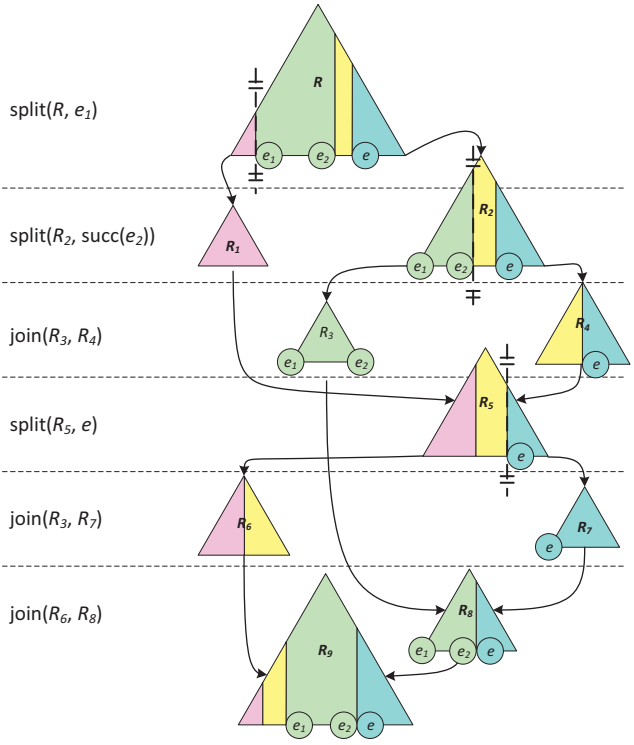


Abbildung 4: Ausführung von move-range$([e_1, e_2]$, before $e)$ auf einem Order Tree.

range in logarithmischer Zeit ausführen (vgl. Abbildung 4): Dazu schneiden wir den zu verschiebenden Bereich $[e_1, e_2]$ mittels split aus, trennen dann den Baum bei e mittels split auf, und setzen schließlich mittels join die entstandenen Teilbäume wieder entsprechend zusammen.

Bulkloading. Mittels insert lässt sich ein Order Tree inkrementell aufbauen, was für n Einträge insgesamt eine Komplexität von $\mathcal{O}(n \log n)$ hat. Wenn aber sämtliche Strukturinformationen der Hierarchie bereits vorliegen, können wir den Baum auch durch „Bulkloading" aufbauen, was mit Hilfe einer Hashtabelle in $\mathcal{O}(n)$ gelingt. Dazu benötigen wir zuerst eine Zwischenrepräsentation der Hierarchie, die uns eine effiziente Tiefensuche erlaubt. Diese erhalten wir in einem ersten Schritt, indem wir eine Hashtabelle von den Knotenschlüsseln zu jeweils einer Kindknotenliste erzeugen, und diese in einem Durchlauf durch die ursprüngliche Hierarchietabelle auffüllen. Einen während dem Durchlauf neu angetroffenen Knoten v der Hierarchietabelle müssen wir jeweils in die Kindliste von dessen Elternknoten p einfügen. Die Hashtabelle liefert uns die Kindliste von p in $\mathcal{O}(1)$. In einem zweiten Schritt führen wir dann anhand der Hashtabelle eine Tiefensuche durch die Hierarchie durch, und erzeugen bei jedem Knoten v *vor* dem Betrachten seiner Kinder einen „unteren" Order-Tree-Eintrag $v.l$ sowie *nach* dem Betrachten seiner Kinder einen „oberen" Eintrag $v.u$. Da dabei die Intervallgrenzen bzw. Order-Tree-Einträge in geordneter Reihenfolge von unten nach oben erzeugt werden, können wir allgemein bekannte Algorithmen

```
function H.insert-node(below v)
    e_1 ← R(H).insert(before v.u)
    e_2 ← R(H).insert(before v.u)
    return (e_1, e_2)

function H.remove-node(v)
    R(H).remove(v.l)
    R(H).remove(v.u)

function H.relocate-subtree(v_1, below v_2)
    R(H).move-range(
            [v_1.l, v_1.u], before v_2.u)

function H.is-ancestor(v_1, v_2)
    return R(H).compare(v_1.l, v_2.l)
        ∧ R(H).compare(v_2.u, v_1.u)

function H.next-sibling(v)
    e ← R(H).succ(v.u)
    if R(H).is-upper-bound(e)
        return ⊥    // no sibling
    return R(H).node(e)
```

Abbildung 5: Umsetzung von Operationen auf einer Hierarchie H in Operationen auf dem zugehörigen Order Tree $R(H)$.

zum Aufbauen eines AVL-, Rot-Schwarz- oder B-Baums in $\mathcal{O}(n)$ einsetzen.

3.3 Der Order Tree als Hierarchie-Index

In Abbildung 5 zeigen wir nun, wie sich Anfragen und Operationen auf einer Hierarchie unmittelbar in Operationen auf dem zugehörigen Order Tree übersetzen lassen. Mit **insert-node** bezeichnen wir das Einfügen eines neuen Blattknotens als letztes Kind unterhalb (below) eines gegebenen Referenzknotens v. Diese Operation entspricht dem Einfügen zweier Einträge in den Order Tree, nämlich der unteren und der oberen Intervallgrenze; sie werden bezüglich der Ordnungsrelation $<$ zu direkten Vorgängern (daher "before") der oberen Intervallgrenze $v.u$ ihres Elternknotens. Einen Knoten zu löschen (**remove-node**) bedeutet, seine zwei Intervallgrenzen $v.l$ und $v.u$ zu löschen. Das Umhängen (**relocate-subtree**) eines Teilbaums mit Wurzel v_1 als letztes Kind unter (below) einen Knoten v_2 übersetzen wir direkt in move-range des Intervalls von v_1 unmittelbar vor die obere Intervallgrenze von v_2. Was strukturbezogene Anfragen betrifft, zeigen wir beispielhaft die Implementierung des Axis-Checks **is-ancestor**. Analog zur Definition in Abschnitt 2 vergleichen wir die Intervallgrenzen der Knoten, wobei wir nun die Vergleichsfunktionalität des Order Trees benutzen. Axis-Checks entlang anderer Achsen lassen sich analog implementieren. Als weiteres Beispiel betrachten wir die Anfrage **next-sibling**, die den nachfolgenden Geschwisterknoten eines Knotens ermittelt. Ihre Implementierung bedient sich der Möglichkeit, im Order Tree in amortisiert konstanter Zeit zum nachfolgenden Eintrag zu gehen (succ) und dann den Zeiger zum zugehörigen Knoten zurückzuverfolgen (node).

4 Auswertung

Um die Anwendbarkeit unseres Ansatzes in der Praxis zu beurteilen, haben wir verschiedene Varianten des Order Trees in C++ implementiert: einerseits basierend auf dem AVL-Baum, andererseits basierend auf dem B-Baum mit wählbarem minimalem Verzweigungs-

grad d, wobei wir für Messungen die Werte $d = 2$, 16 und 128 benutzt haben. Beide Varianten speichern die Teilbaumhöhe h in jedem Knoten. Die B-Baum-Implementierung benutzt zudem eine indirekte Eintragstabelle, um stabile Zeiger auf Einträge zu ermöglichen, wie in Abschnitt 3.1 skizziert. Unsere Messungen haben wir auf einer HP Z600 Workstation durchgeführt, bestückt mit einer 6-kernigen Intel Xeon X5650 CPU (2,66 GHz, 12 MB Cache), 24 GB RAM und SuSE Linux Enterprise Server als Betriebssystem.

Für die Messungen wird eine Hierarchie durch Einfügen an zufälligen Positionen aufgebaut, wobei wir die Knotenanzahl n variieren über $n = 10^4$, 10^6 und 10^8. Nach dem Aufbauen werden mit gleichen Wahrscheinlichkeiten unter Zeitmessung die Order-Tree-Operationen insert, remove oder compare auf jeweils zufällig gewählten Hierarchieknoten ausgeführt. Aus Platzgründen können wir die Ergebnisse an dieser Stelle nicht vollständig wiedergeben, fassen unsere Beobachtungen aber wie folgt zusammen: **(1.)** Bei $n = 10^8$ – einer Größe, welche die wenigsten Anwendungsfälle benötigen werden – erreichen wir absolute Raten von durchschnittlich 1–5 Millionen Operationen pro Sekunde für jede der verglichenen Operationen. Die verschiedenen Order-Tree-Varianten liegen in derselben Größenordnung; insbesondere sind die Zahlen für den AVL-basierten Baum nahezu gleich zu denen des B-Baums mit $d = 2$ (der einem 2-3-4-Baum entspricht). **(2.)** Außerdem liegen die Raten in derselben Größenordnung wie vergleichbare Operationen, die wir auf einem *Cache-sensitiven B$^+$-Baum* [RR00] durchgeführt haben, einer hochperformanten Indexstruktur, die für moderne Hardware optimiert wurde. **(3.)** Bei der B-Baum-Variante lässt sich durch Erhöhen von d die Performanz der compare-Funktion signifikant steigern. Das rührt daher, dass mit größerem d die B-Baum-Höhe geringer wird. Gleichzeitig erhöht sich aber der Verwaltungsaufwand für die Knoten, was Änderungsoperationen merklich verlangsamt. Man kann also durch Anpassen von d einen dem jeweiligen Anwendungsfall angepassten Kompromiss zwischen Änderungs- und Anfrageperformanz erreichen. **(4.)** Durch Erhöhen von n fällt die Performanz etwas schneller ab, als die asymptotische Komplexität von $\mathcal{O}(\log n)$ vermuten lässt. Wir führen dies auf Caching-Effekte zurück: Je mehr Arbeitsspeicher benutzt wird, desto mehr Cache-Misses verursachen die zufällig gewählten Operationen, da immer geringere Anteile der Hierarchie in den Cache passen.

5 Verwandte Arbeiten

Allgemein lässt sich sagen, dass Hierarchien im Kontext von relationalen Datenbanken in der Vergangenheit erstaunlich wenig Aufmerksamkeit durch die Datenbankforschung erfahren haben. Anders ist die Situation bei XML-Datenbanken: Für XML-Dokumente – die inhärent hierarchisch aufgebaut sind – gibt es viele Indexstrukturen, oft aufbauend auf dem relationalen Schema. Eines der am weitesten entwickelten Schemata, das XML-Dokumente auf eine Weise kodiert, dass sämtliche XPath-Achsen unterstützt werden, ist das *Pre/Post*-Labeling-Schema von Grust et al. [GvKT04]. Wie viele vergleichbare Schemata für XML ist es aber nicht dynamisch. In einem späteren Beitrag erreichen Boncz et al. [BMR05] mit MonetDB größere Änderungsraten mit einem modifizierten Pre/Post-Schema.

Weitere bekannte Labeling Schemes sind unter den Begriffen *Prefix-based Encoding* oder *Dewey Encoding* bekannt – siehe die Arbeit zu ORDPATH von O'Neil et al. [OOP$^+$04],

auf der Haustein et al. [HHMW05] weiter aufbauen. Mit dem Vorreservieren von Lücken in Labeling Schemes und speziell der geschachtelten Intervallkodierung befassen sich Yu et al. [YLML05] und Li et al. [LM01]. Trophasko [Tro05] diskutiert numerische Lösungsansätze zum Problem der Neunummerierung, wie *Farey Fractions* oder *Dyadic Fractions*. All diese Ansätze lassen sich jedoch durch ungünstige Einfügesequenzen der Länge n außer Gefecht setzen, so dass eben doch eine Neunummerierung in $\mathcal{O}(n)$ notwendig wird, oder aber die Labels auf Größen proportional zu n anwachsen.

Ein weiterer, in unserem Zusammenhang relevanter Forschungszweig sind *Succinct Data Structures* für Bäume – erinnern doch geschachtelte Intervalle ohne die numerischen Intervallgrenzen stark an eine Baumrepräsentation, die in diesem Bereich unter dem Begriff *Balanced Parentheses* bekannt ist. Statische Kodierungen wurden hier sehr gründlich untersucht (z. B. [GRR06]), sind aber in unserem Kontext irrelevant. In [FM09] untersuchen Farzan et al. auch effizient änderbare Kodierungen. Vielversprechend ist der Ansatz von Navarro et al. [Nav09], eine vergleichsweise elegante dynamische Repräsentation durch einen so genannten *Range-Min-Max-Tree*, welche eine große Menge an Operationen in $\mathcal{O}(\log n)$ unterstützt. Es ist allerdings unklar, inwieweit sich die eher Algorithmentheoretischen Beiträge mit ihren oft recht aufwendigen Konstruktionen in der Praxis implementieren, geschweige denn in Code von akzeptabler Performanz überführen lassen.

6 Zusammenfassung

Der sicherlich interessanteste Beitrag unserer Arbeit besteht darin, dass wir mit dem Order Tree einen generischen Mechanismus entwickelt haben, um beliebig geformte Hierarchien durch einen jederzeit balancierten Baum darzustellen. Dieser Mechanismus erlaubt es uns, gewisse Anfragen auf hierarchischen Daten zu beschleunigen. Soweit wir unsere Indexstruktur in dieser Arbeit konkret vorgestellt haben, unterstützt diese das Einfügen und Löschen von Knoten und das Umhängen von Teilbäumen in jeweils $\mathcal{O}(\log n)$, sowie die üblichen Axis-Check-Anfragen in $\mathcal{O}(\log n)$.

Das Konzept des Order Trees lässt sich unter Wiederverwendung bekannter Baumindexstrukturen umsetzen. Wir haben mit Implementierungen experimentiert, die auf dem AVL-Baum und auf dem B-Baum aufbauen; prinzipiell sind aber auch andere Arten von Baumstrukturen als Grundlage denkbar. Speziell im Datenbankbereich ist es von Vorteil, auf vorhandene Index-Implementierungen zurückgreifen zu können, die gründlich erprobt, optimiert und ausgereift sind.

In Zukunft konzentrieren wir uns zum einen darauf, den Speicherplatzbedarf und die Cache-Effizienz unseres Indexes weiter zu verbessern, indem wir das Datenlayout kompaktifizieren, Zeiger einsparen und Techniken aus dem Gebiet der Cache-conscious Data Structures einsetzen. Zum anderen möchten wir zusätzliche Anfragen auf dem Order Tree implementieren und untersuchen, wo die Grenzen bezüglich der durch den Order Tree theoretisch zu unterstützenden Anfragen liegen.

Literatur

[BMR05] Peter Boncz, Stefan Manegold und Jan Rittinger. Updating the pre/post plane in MonetDB/XQuery. In *Inf. Proc. Int. Workshop XQuery Implementation, Experience, and Perspectives (XIME-P)*, 2005.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest und Clifford Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 3. Auflage, 2009.

[FM09] Arash Farzan und J. Ian Munro. Dynamic succinct ordered trees. In *Automata, Languages and Programming*, Jgg. 5555 of *Lecture Notes in Computer Science*, Seiten 439–450. Springer, Berlin/Heidelberg, 2009.

[GRR06] Richard F. Geary, Rajeev Raman und Venkatesh Raman. Succinct ordinal trees with level-ancestor queries. *ACM Trans. Algorithms (TALG)*, 2(4):510–534, 2006.

[GvKT04] Torsten Grust, Maurice van Keulen und Jens Teubner. Accelerating XPath evaluation in any RDBMS. *ACM Trans. Database Systems (TODS)*, 29:91–131, 2004.

[Hau11] Birgitta Hauser. Hierarchical Queries with DB2 Connect By. `http://www.ibm.com/developerworks/ibmi/library/i-db2connectby/`, 2011.

[HHMW05] Michael P. Haustein, Theo Härder, Christian Mathis und Markus Wagner. DeweyIDs—the key to fine-grained management of XML documents. *Proc. Brasilian Symp. Databases*, 20:85–99, 2005.

[LM01] Quanzhong Li und Bongki Moon. Indexing and querying XML data for regular path expressions. In *Proc. Int. Conf. Very Large Databases (VLDB)*, Seiten 361–370, 2001.

[Nav09] Gonzalo Navarro. Fully-functional static and dynamic succinct trees. *Tech. Rep., abs/0905.0768*, 2009.

[OOP+04] Patrick O'Neil, Elizabeth O'Neil, Shankar Pal, Istvan Cseri, Gideon Schaller und Nigel Westbury. ORDPATHs: insert-friendly XML node labels. In *Proc. Int. Conf. Management of Data (SIGMOD)*, Seiten 903–908, 2004.

[Ora10] Oracle Corp. Hierarchical Queries — Oracle Database SQL Language Reference 11g Release 1 (11.1). `http://docs.oracle.com/cd/B28359_01/server.111/b28286/queries003.htm`, 2010.

[RR00] Jun Rao und Kenneth A. Ross. Making B+-trees cache conscious in main memory. In *Proc. Int. Conf. Management of Data (SIGMOD)*, Seiten 475–486. ACM, 2000.

[Tro05] Vadim Tropashko. Nested intervals tree encoding in SQL. *ACM Rec. Int. Conf. Management of Data (SIGMOD)*, 34(2):47–52, 2005.

[YLML05] Jeffrey Xu Yu, Daofeng Luo, Xiaofeng Meng und Hongjun Lu. Dynamically updating XML data: numbering scheme revisited. *World Wide Web: Internet and Web Information Systems*, 8:5–26, 2005.

Tutorium: Neue Hardwarearchitekturen für das Datenmanagement (DPMH)

Cagri Balkesen, Louis Woods, Jens Teubner

Über Jahrzehnte hinweg gelang es außerordentlich gut, aus der ”Dividende” von Moore’s Law immer schnellere Computersysteme zu bauen, die Anwendungssoftware quasi ganz automatisch beschleunigten. Die Grenzen dieses Ansatzes werden immer deutlicher und es ist zwischenzeitlich klar, dass signifikante Leistungssteigerungen in Zukunft nur noch durch einen hohen Grad an Hardware-Spezialisierung möglich sein werden. Für Fließkomma- oder Grafikoperationen hat sich dieser Trend bereits erfolgreich in der Praxis durchgesetzt.

Im Tutorium werden daher die Konsequenzen diskutiert, die neue Hardwaretechnologie auf datenintensive Anwendungen hat; allen voran diskutieren wir die Konsequenzen für Datenbankmanagementsysteme. Zu Beginn des Tutoriums werden wir Erneuerungen in der Computer- Systemarchitektur betrachten, die sich bereits in der Praxis durchgesetzt haben, von Konzepten wie tiefen Cache-Hierarchien über Parallelität und Multi-Core-Systeme bis hin zu sogenannten NUMA-Systemen. Dabei wird deutlich werden, wie wichtig ein geeignetes Zugriffsmuster (zum Beispiel auf den Hauptspeicher des Systems, aber auch zwischen parallelen Einheiten des Systems) für die Laufzeit von Algorithmen sein kann und es wird gezeigt, wie Datenbankalgorithmen konstruiert werden können, damit sie geeignete Zugriffsmuster aufweisen.

Wir werden dann Co-Prozessoren als echt spezialisierte Hardware diskutieren. Dabei werden wir im besonderen auf Grafikprozessoren (GPUs) und programmierbare Logikbausteine (FPGAs) eingehen, da diese sich im Kontext eines Datenbankeinsatzes als besonders nutzbringend erwiesen haben. Aufbauend auf der internen Architektur der jeweiligen Technologie werden wir zeigen, wie Co-Prozessoren programmiert werden und wie ihre Funktionalität in ein Gesamtsystem eingebunden werden kann

Ziel des Tutoriums wird es sein, die spezifischen Charakteristika der verschiedenen Technologien zu vermitteln. Dadurch werden die Teilnehmer nachher in der Lage sein, Aufwand und Nutzen des Einsatzes spezieller Hardware für einen gegebenen Anwendungskontext zu bewerten.

Über die Autoren

Cagri Balkesen ist wissenschaftlicher Mitarbeiter in der Systems Group an der ETH Zürich. Im Rahmen seiner Promotion beschäftigt er sich dort mit Datenbanktechnologien für den Einsatz im Hauptspeicher sowie mit Datenstromverarbeitung. Er hält einen MSc (Informatik) der ETH Zürich sowie einen BSc (Computer Engineering) der Middle East Technical University in der Türkei.

Louis Woods ist wissenschaftlicher Mitarbeiter in der Systems Group an der ETH Zürich. Sein Forschungsgebiet umfasst FPGAs im Kontext von Datenbanksystemen, parallele Algorithmen, Datenstromverarbeitung und Echtzeit-Mustererkennung. Er hält einen BSc und MSc der ETH Zürich in Informatik.

Jens Teubner ist Oberassistent und Leiter des Avalanche-Forschungsprojekts an der ETH Zürich. Das Projekt ist führend in der Verwendung von FPGA für den Datenbankeinsatz. Weitere Forschungsinteressen betreffen die Datenverarbeitung im Hauptspeicher sowie Hochgeschwindigkeits-Netzwerke. Jens Teubner promovierte 2006 an der TU München auf dem Gebiet der XML-Verarbeitung und arbeitete von 2007 bis 2008 im IBM-Forschungslabor in Hawthorne, New York.

GI-Edition Lecture Notes in Informatics

P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.

P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Lan-guage to Information Systems, NLDB'2001.

P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.

P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.

P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.

P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.

P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.

P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.

P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.

P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.

P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.

P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.

P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.

P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.

P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.

P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.

P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.

P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.

P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungsband).

P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.

P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.

P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit

P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen

P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003

P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web

P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin

P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Er-fahrungen und Visionen

P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems

P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications

P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003

P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft

P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)

P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)

P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik

P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik

P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003

P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003

P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004

P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste

P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistance, Scalability, Transactions – Database Mechanisms for Mobile Applications

P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services

P-45 Bernhard Rumpe, Wofgang Hesse (Hrsg.): Modellierung 2004

P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment

P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society

P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications

P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven

P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik

P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004

P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration

P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration

P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government

P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen

P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality

P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments

P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): “Heute schon das Morgen sehen“

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006

P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur

P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006

P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006

P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1

P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2

P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen

P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies

P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices

P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik

P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik

P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten

P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments

P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)

P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme

P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007

P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops

P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications

P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures

P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1

P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2

P-111 Christian Eibl, Johannes Magenheim, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik

P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis

P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)

P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics

P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007

P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007

P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and manegement The 4th International Conference on Grid Service Engineering and Management GSEM 2007

P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007

P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications

P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges

P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik

P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)

P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Speciel Interest Groups PARS and PARVA

P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT:
Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung

P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)

P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008

P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany

P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems

P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
1. DFN-Forum Kommunikations-technologien Beiträge der Fachtagung

P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting. CC

P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik

P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 1

P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 2

P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik –
Aktuelle Forschungsergebnisse

P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics GCB 2008

P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures

P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008

P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwick-lung und -wartung im Fachbereich Wirtschaftsinformatik

P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics

P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informations-systeme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management

P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung

P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)

P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management

P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)

P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009

P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I^{2}CS 2009

P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
2. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering
2009 - Workshopband

P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software

P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop EMISA 2009

P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik

P-154 Stefan Fischer, Erik Maehle Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben

P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«

P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics 2009

P-158 W. Claupein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture Reloaded – Informationsgestützte Landwirtschaft

P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010

P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 – Workshopband
(inkl. Doktorandensymposium)

P-161 Gregor Engels, Dimitris Karagiannis Heinrich C. Mayr (Hrsg.)
Modellierung 2010

P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010

P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre Informationssysteme
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler Kollaboration

P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)
10th International Conference on Innovative Internet Community Systems (I^2CS) – Jubilee Edition 2010 –

P-166 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on Electronic Voting 2010
co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)
Didaktik der Informatik
Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik

P-169 Michael Kerres, Nadine Ojstersek Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.

P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010
Sicherheit, Schutz und Zuverlässigkeit

P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2010)
Modellgestütztes Management

P-172 Stefan Klink, Agnes Koschmider Marco Mevius, Andreas Oberweis (Hrsg.)
EMISA 2010
Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme
Beiträge des Workshops der GI-Fachgruppe EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung)

P-173 Dietmar Schomburg, Andreas Grote (Eds.)
German Conference on Bioinformatics 2010

P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)
perspeGKtive 2010
Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

P-175 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1

P-176 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2

P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fähnrich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)
INFORMATIK 2010
Business Process and Service Science – Proceedings of ISSS and BPSC

P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)
Vom Projekt zum Produkt
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschafts-informatik (WI-MAW), Aachen, 2010

P-179 Stefan Gruner, Bernhard Rumpe (Eds.)
FM+AM`2010
Second International Workshop on Formal Methods and Agile Methods

P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW)
14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)

P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft

P-182 Ronald Maier (Hrsg.)
6th Conference on Professional Knowledge Management
From Knowledge to Action

P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)
Software Engineering 2011
Fachtagung des GI-Fachbereichs Softwaretechnik

P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)
Software Engineering 2011
Workshopband
(inkl. Doktorandensymposium)

P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)

P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)
11th International Conference on Innovative Internet Community Systems (I^2CS)

P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
4. DFN-Forum Kommunikations-technologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn

P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. 5.–8. September 2011, Dresden

P-189 Thomas, Marco (Hrsg.)
Informatik in Bildung und Beruf
INFOS 2011
14. GI-Fachtagung Informatik und Schule

P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2011)

P-191 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2011
International Conference of the Biometrics Special Interest Group

P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)
INFORMATIK 2011
Informatik schafft Communities

P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)
IMDM 2011

P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)
Informationstechnologie für eine nachhaltige Landbewirtschaftung
Fokus Forstwirtschaft

P-195 Neeraj Suri, Michael Waidner (Hrsg.)
Sicherheit 2012
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)

P-196 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2012
Proceedings of the 11th International Conference of the Biometrics Special Interest Group

P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012

P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)
Software Engineering 2012
Fachtagung des GI-Fachbereichs Softwaretechnik

P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)
Software Engineering 2012
Workshopband

P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)
ARCS 2012 Workshops

P-201 Elmar J. Sinz Andy Schürr (Hrsg.)
Modellierung 2012

P-202 Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.)
MMS 2012:Mobile und Ubiquitäre Informationssysteme

P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
5. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung

P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)
12th International Conference on Innovative Internet Community Systems (I2CS 2012)

P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)
5th International Conference on Electronic Voting 2012 (EVOTE2012)
Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)
EMISA 2012
Der Mensch im Zentrum der Modellierung

P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
24.–26. September 2012

P-208 Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert Matthies, Wolf-Tilo Balke, Lars Wolf (Hrsg.) INFORMATIK 2012

P-209 Hans Brandt-Pook, André Fleer, Thorsten Spitta, Malte Wattenberg (Hrsg.) Nachhaltiges Software Management

P-210 Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, Silke Spitzer (Hrsg.) Automotive – Safety & Security 2012 Sicherheit und Zuverlässigkeit für automobile Informationstechnik

P-211 M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen (Hrsg.) Massendatenmanagement in der Agrar- und Ernährungswirtschaft Erhebung - Verarbeitung - Nutzung Referate der 33. GIL-Jahrestagung 20. – 21. Februar 2013, Potsdam

P-213 Stefan Kowalewski, Bernhard Rumpe (Hrsg.) Software Engineering 2013 Fachtagung des GI-Fachbereichs Softwaretechnik

P-214 Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mit schang, Theo Härder, Veit Köppen (Hrsg.) Datenbanksysteme für Business, Technologie und Web (BTW) 2013 13. – 15. März 2013, Magdeburg

P-215 Stefan Wagner, Horst Lichter (Hrsg.) Software Engineering 2013 Workshopband (inkl. Doktorandensymposium) 26. Februar – 1. März 2013, Aachen

P-216 Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.) Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband 11. – 12. März 2013, Magdeburg

The titles can be purchased at:

Köllen Druck + Verlag GmbH
Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn
Fax: +49 (0)228/9898222
E-Mail: druckverlag@koellen.de

Gesellschaft für Informatik e.V. (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into
• seminars
• proceedings
• dissertations
• thematics
current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: http://www.gi.de/service/publikationen/lni/

ISSN 1617-5468
ISBN 978-3-88579-610-7

The BTW 2013 in Magdeburg is the 15th conference of its kind reflecting the broad range of academic research and industrial development work within the German database community. This year's conference focuses on a broad range of database topics covering information extraction and integration, data analytics, web data management, service-oriented Architectures, cloud computing ,and virtualization. This volume contains contributions from the refereed scientific program, the refereed industrial program, and the refereed demo program. Furthermore, the dissertation award winner and three keynotes are presented in this volume.