Next Generation Operational Business Intelligence exploring the example of the bake-off process: using in-memory data management

Alexander Gossmann
Research Group Information Systems University of Mannheim
agossman@mail.uni-mannheim.de

Abstract: Rapidly changing markets and competitive pressure cause the need for agility on all management levels. Fast business decisions are driven by the need of having a holistic and updated view on the value chain, throughout the strategic, tactical and operational level, supported by appropriate information systems. Transferred to the retail domain, local store managers are focused on operational decision making, while top management requires a view on the business at a glance. Both requirements rely on information, derived from transactional data, whereas the analytic views on this data differ completely. Accordingly, different data mining capabilities in the underlying information system are targeted, especially related to processing masses of transactional data in real-time. The examined software system is a SAP HANA in-memory appliance, which satisfies the aforementioned divergent analytic capabilities, as shown in this paper.

1 Introduction

Operational Business Intelligence is becoming an increasingly important field in Business Intelligence (BI), which traditionally is targeting strategic and tactical decision-making [Wh05]. Operational BI provides information with a low latency, often referred to real-time, and related to day-to-day or intraday decisions. This implies that process relevant information has to be delivered at a reasonable, short time and an accessible format to the decision-maker. An adequate synchronization between the real world and its artificial representation in a data warehouse can satisfy these requirements [Ha04].

Providing real-time information can be relatively costly, as the raw, transactional data has to be gathered, enhanced and processed, to target the different views on the business and consequently to be turned into valuable knowledge. In contrast, real-time information can lead to a significant competitive advantage, by shortening the action time. This applies for decisions on strategic, as well as on operational level. Analytical applications contribute in addition to reach competitive advantage, as they provide a better basis of decision making and therefore supporting the decision maker. As the requirements differ for the appropriate management levels, in sense of granularity and analytical demands, a powerful and flexible abstraction level of the data layer, as well as the processability of huge amounts of transactional data are essential.

With the advent of in-memory data management (IMDM), it becomes possible to leverage the enormous computing power of modern hardware, meaning multi core architectures, with terabytes of available main memory. Sophisticated data management techniques, like column orientation and compression, provide additional performance advantages in processing huge amounts of data [Kr12].

For the ability of delivering operational BI, the appropriate data warehouse should satisfy real-time data updates as well as operational analytic requirements [Ec07]. The next evolutionary step of operational BI would be to provide a seamless provision of real-time information to all organizational levels (operational, tactical and strategic). An agile and highly effective data layer can fulfill this, by processing directly operative data. In contrast to traditional, disk-based data warehouse approaches, the lightweight SAP HANA architecture could close the gap between the different technological alignments of strategic, tactical and operational analytic demands.

This paper will present a case study in the field of fast moving goods of a large food retailer. The tremendous possibilities, offered by the recently introduced SAP HANA IMDM architecture, will be described below.

The remainder of this paper is organized as follows. The second chapter contains the general potentials of real-time BI. In Section 2.1 an introduction to the action time concept and its impact on the business value of decision making will be provided. Further, in section 2.2 and 2.3 related work in operational BI will be outlined, concluding with potentials provided by IMDM in section 2.4. In chapter 3 the conducted case study and the results will be described, from an architectural and a development point of view (sections 3.3-3.5), after a short introduction into process requirements (sections 3.1, 3.2). This paper concludes with an outlook on further work in chapter 4.

2 Potentials of operational BI

The following concepts describe the potentials of applying operational BI in general. As the genus 'real-time' information is used in this work, it is important to distinguish between the definition of real-time in other disciplines and the BI perspective. While real-time is associated with 'instantaneous' [Wa06] in the engineering domain for instance, for BI the real-time characteristic is dependent on the specific business need. Information provision in BI is often connected to an underlying data warehouse, containing historical and consolidated data, transforming transactional data into insight. Therefore, the data could be processed in real-time or near real-time, or at least 'fast enough', depending on the business requirement, which will be denoted as real-time BI in this work, this expression is often used synonymously to operational BI .

To give a more precise definition of what is 'fast enough' in decision context, the next section considers the concept of action time. Further, two categories of analytic alignments will be presented, as required by the strategic, tactical, and operational management. After an introduction into common operational BI architecture approaches, the last section will conclude with a technological view on real-time BI, especially in terms of the new possibilities established by in-memory data management.

2.1 The concept of action time optimization

Turning data into valuable information causes latencies, from a computational point of view, as depicted in the introduction. According to Hackathorn [Ha04] these latencies could be categorized as follows (cf. Figure 1):

- **data latency** ($t_1 – t_0$): describes the time for a transaction caused by a business event to be stored and made available for analysis
- **analysis latency** ($t_2 – t_1$): is the time needed to generate information out of the raw data, by applying business logic and providing a specific view on the data
- **decision latency** ($t_3 – t_2$): is the time for a decision maker to recognize the information delivered by the analytic engine of the system and convert it into actions

The whole process from the occurrence of a business event, until actions to be taken is concluded as ‘action time’ ($t_3 – t_0$). As depicted in Figure 1, the y-axis represents the business value in general, related to the action time. The graph illustrates a decay curve, the business value is supposed to decrease rapidly after a business event, with a flattening decay in time. This observation can be made for many business scenarios, and is especially true for the observed case study. However, there are some exceptions to this rule [Ha04].

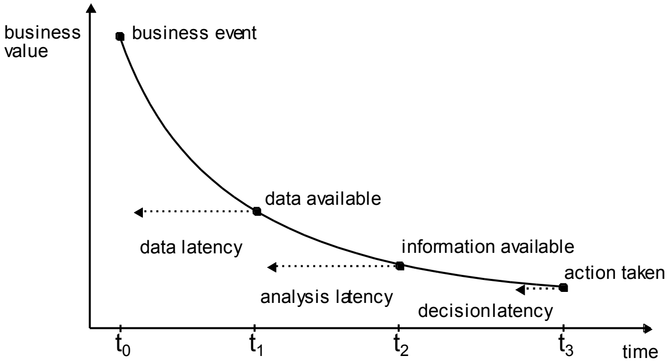

Figure 1: Action time and business value

The potential of decreasing the action time is denoted in Figure 1 by the dotted arrows. While the data latency and analysis latency can be increased dramatically by the used technology, the decision latency relies in most cases on the cognitive capability of a human decision-maker. Although, it is hard to estimate, how a decision-maker can be influenced by the underlying system, the quality of the analysis, as well as fast information delivery in an appropriate form, should have a positive effect on the decision making process, symbolized by a short arrow.

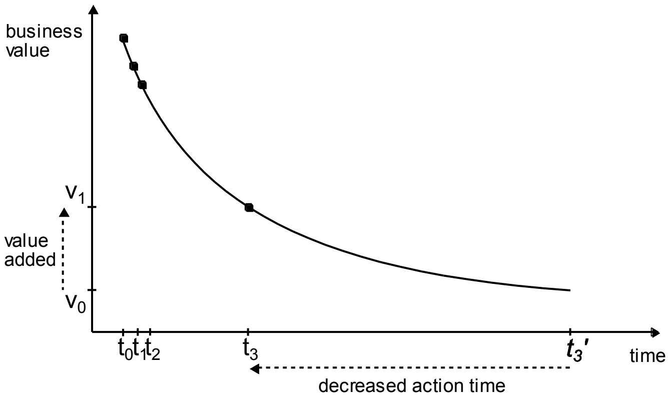

Figure 2: Added value by reduced action time

Figure 2 shows the impact of reduced action time ($t_3' - t_3$) on the business value in general. The added business value ($v_1 - v_0$), can be measured for instance in reduced costs, higher customer satisfaction, and competitive advantage. The business value is considered to increase from operational, through tactical to strategic decisions, while the amount of decisions decrease [Ec07]. Although operational decisions have to be made quickly, per se, tactical and strategic decisions would profit from decreased action time, having no timely disruption on their view of business provided by real-time information.

2.2 Analytic applications

Analytic applications in BI can be commonly classified into two categories from the data management point of view. On one side, the strategic alignment of an analytic system is to provide a flexible navigation through dynamically aggregated measures and dimensions. From an operational perspective on the other, the requirements can often be formulated as (business) process optimization problems [OLW08]. The first capability can be related to traditional on-line analytical processing (OLAP) approaches, the latter to the domain of machine learning and data mining. In-between these two extremes, fast and selective querying, with a certain amount of aggregation as well as mining capabilities, could be assigned to the tactical demand.

The OLAP approach requires an a priori knowledge from a requester, whereas the operational related data mining is based on learning directly from the data. In this case the system should automatically find patterns and derive decision support, like recommendations for a line of business manager [OLW08]. Predictive analytics in combination with a business rules engine could make human intervention even expendable [Ec07], but this observation is related to further work.

2.3 Technological challenges for common operational BI architectures

As operational BI is closely related to the transactional process level, there are two main architecture approaches for capturing information from transactional data, according to Eckerson [Ec07]):

- **direct querying** the on-line transactional processing (OLTP) system
- **off-load**, such as (near) real-time data replication into an operational data store (ODS) or a data warehouse

Direct querying
Direct querying of operational systems can further divided into the following solutions:

- built-in reporting packages, with the drawback of being static and inflexible
- a BI front-end tool for querying the transactional database
- enterprise information integration (EII), the federation approach with potentially complex multiple OLTP schemas to handle

All presented solutions have the disadvantage of potentially affecting significantly the performance of the operational system. Either, reporting is limited to a certain extent, or productive processes run at risk of being disturbed.

Off-load, operational data store

In order to avoid the downsides of direct querying, many companies use the operational data stores (ODS) separately or integrated in an OLAP data warehouse, to increase operational and tactical query performance and take the load off an OLTP system. The ODS is typically represented by database tables with a high tuple cardinality, neglecting the normalization constraint of OLTP, hence avoiding expensive join operations. Furthermore, the ODS is characterized by the ability of modifications, storing mirror images from relevant OLTP data. This approach allows the consolidation of transactional data as well as the integration of additional data (e.g., data crawled from the web, enhancing the operational information). However, the drawback of the ODS approach is the management of data synchronization, thus increased complexity. In case of a standalone ODS solution, only limited OLAP functionality is obtainable.

Real-time data warehouse, with consolidated operational reporting

In order to unify all organizational reporting needs, some companies integrate operational reporting into their existing data warehouse. This implies architectural challenges to a data warehouse, especially for the following two reasons [Ec07]:

- providing **real-time updates** of transactional data
- allowing fast and flexible tactical reporting, i.e. providing highly **selective point queries** (e.g. depicting only few, scoped rows out of millions of data records)

Both requirements are opposing to the common design of a data warehouse, providing OLAP functionality. Typically, the data in OLAP systems has to be modeled in a certain multidimensional schema, such as data-cubes [DT99]. These different requirements on data management, causing contrary database workloads, are known as mixed workload, which is especially true for the differentiation between OLAP and OLTP systems ([Ec07], [Re11]). From this perspective, real-time enabled data warehouse architectures have to provide a mixed workload handling.

In addition, performance constraints in disk-based data warehouses require pre-aggregating data in data marts (respectively data cubes). This approach is called view materialization [GM95]. In summary, the following problems mainly occur in common data warehouse approaches:

- synchronization of data marts and operational data increases data latency
- extended extraction transformation load (ETL) techniques are required, raising complexity and probability of erroneous data
- drill down to transactional line of the appropriate business event is limited, affecting especially tactical decision making, if supported at all, increasing the analysis time
- OLAP oriented systems often fail in providing the appropriate, operational data mining functionalities [OLW08], especially in case of pre-aggregations

The limitations, explained above can lead very soon to an ‘out of sync’ information system, where top management has a different, outdated view than tactical and operational management and vice versa. This could have a very negative impact on the action time, as well as on decision quality, ending in a limited trust in the delivered information, in other words in a self-defeating BI.

2.4 Areas of improvement by in-memory data management

Common, disk-based, real-time data warehouse approaches, as described in above section, tend to imply a highly complex system landscape. The primary reason for this complexity is the need to provide acceptable response times for the respective business analyst, often realized through specialized systems for the appropriate analytic purposes. Therefore, the provision of real-time BI is potentially expensive, especially caused by high development and maintenance costs in such piecemeal composed system landscapes [Si12].

Nevertheless, it is desirable to have a ‘single point of truth’ solution, for all organizational reporting demands. The main challenge is to unify the analytic requirements of strategic, tactical and operational decision-maker in one real-time BI system. The advantages for organizations are obvious:

- central management of the data warehouse and analytic applications by a business intelligence competence center (BICC) for instance, responsible for all reporting demands [Ec07]
- decreased costs for development and maintenance of the BI system, through simplification of the system architecture
- improving decision quality, by providing tailored analytics (OLAP, tactical, operational)

- reducing action time for all management levels, providing synergy effects on the management process
- improving agility, by the enablement of fast anticipation and prediction of uncertain market trends

The following innovations are introduced by in-memory data management, to overcome the constraints, listed in the previous sections, and allowing a unified real-time BI, with a lean data warehouse approach. Especially the concepts, in the SAP HANA appliance software will be matched with the requirements of such a system, and proved in their applicability in the prototype in the next chapter.

OLAP functionality

The SAP HANA IMDM supports column oriented storage, enhanced with sophisticated compression techniques [Kr12]. This approach allows both: leveraging parallel processing and the usage of the in-memory appliance as a data warehouse, storing huge amounts of data, and allowing fast querying. Furthermore, materialized views can be avoided, as column orientation provides aggregations on the fly [Pl09]. Multidimensional schema is being calculated at runtime as specific views, providing on-demand data marts, and supporting the full OLAP functionality. This approach, with 'virtual' data marts, dramatically diminishes complexity of a data warehouse particularly with regard to the following aspects:

- the OLAP interface to the strategic requester relies on the same physical tables containing transactional data, provided to the operational and tactical users, thus eliminating the synchronization effort in the ETL process
- a seamless querying is possible, considering the mixed workload problem (see section 2.3)
- data latency can be reduced to a minimum, as only update of the (compressed) column store has to be considered (see below)
- analysis time is likewise affected to get neglectable, due to the analytic alignment of the data flow

Mixed workload

Krueger at al. describes in detail the challenges of combining read optimized column storages with the handling of transactional updates [Kr11].Transactional data inserts and updates are handled automatically, by a so-called delta buffer approach. Furthermore, Sikka et al. introduced the life cycle of database records, involving a row-based first level delta for fast inserts, deletions and updates on line level [Si12] (cf. the initial ODS concept by Inmon [In93]). From architectural point of view, this built-in capability makes the usage of an explicit ODS obviously completely redundant. In conclusion, HANA provides mixed workloads by design, which is one of the central features to support operational BI requirements, at the same time diminishing architectural complexity.

Analytic applications

The SAP HANA appliance software is designed as a data management system (DBMS), with extended analytical capabilities. Consequently, SQL is no more feasible to target the different demands of data intensive processing solely. Therefore, especially for the support of business logic in the analytic context, specialized collections of operators are provided. Following a data flow graph model, these operators are organized as nodes in a so-called calculation graph model ([Si12]). This approach allows on the one hand the usage of optimized operators in regards of query plan optimization like parallelization (cf. calculation engine plan operators). On the other hand, domain specific operator nodes can be applied in conjunction, such as the following selection, as used in the case study:

- **The OLAP – and Join Engine**; providing on the fly (OLAP) views
- **SQL Script**; i. e. imperative language with SQL extensions, and calculation engine plan operators [SA13]
- **R Script**; these are encapsulated R Script calls, through the Rserve interface for statistical and machine learning algorithms
- **Predictive Analytics Library (PAL)**; as part of the SAP Application Function Library (AFL), a custom C++ node [Si12] for data intensive, high performance, statistical and machine learning algorithms [SP13]

The integration of the calculation graph model into the data layer, gives the application designer the full bunch of expressive data processing possibilities. Therefore, especially in conjunction with the support of mixed workloads, this architecture is closing the gap between strategic and operational analytic application alignments (see section 2.2, 2.3).

3 Case Study: managing fast moving goods in retail

In order to prove the claim of the novel operational BI capabilities, introduced by the SAP HANA incarnation of IMDM, a prototypical case study has been conducted. Specifically, the so-called bake-off environment is taken into account. Bake-off units reside in each store and are charged with pre-baked pastries based on the expected customer demand. The trade-off between product availability and loss hereby is extremely high. Customer satisfaction is strongly related to the freshness of pastries and especially to out-of-shelf situations. To meet this customer needs on one side and to avoid loss on the other, can only be achieved through operational excellence, supported by predictive analytics.

From the management point of view, the following user group driven requirements exist: On the one hand, placing orders in the day-to-day business, requires accurate and automated data processing, to increase the quality of the demand forecast. On the other hand, strategic decision makers need a flexible way to drill through the data on different aggregation levels, to achieve a fast action time to changing market conditions.
The observation period of two years is considered. The basic population consists of fine grained, minute wise data for thousands of bake-off units, providing all facts related to the bakery process.

In this case study the following central questioning has been addressed:

- Does the SAP HANA appliance software satisfy the analytic requirements of all different management user groups, based on one common, real-time enabled data foundation, with user satisfying response times (meaning of less than three seconds)?

Due to the use case setup, an off-load scenario, with (near) real-time data provision has been investigated.

3.1 Corporate Level Requirements

On the corporate level, top management is concerned about having a 'bird's eye view' on the whole bake-off business, measured by highly aggregated key figures. Primarily the availability, loss and sales amount measures and their dependencies are considered high level, whereas the reasons for the appearance of these indicators can vary strongly. Accordingly, the allocation of regional and timely trends as soon and precisely as possible, addresses tactical decisions, by decreasing the aggregation and increasing information granularity. For accurate decisions, it is essential to drill down to the store level, to indicate the reasons for certain business patterns. In this context, it is desirable to have a seamless navigation through a sufficient amount of historical data, aggregated freely in time, and at the same time, to have constantly updated information. Especially, in test scenarios, for instance of a new product introduction, customer acceptance and trends could be recognized very soon, by monitoring the customer behavior quasi in real-time. The same is true for marketing actions taken and the identification of their impacts. It is important that the system is having user satisfying response times, allowing the exploration of a huge amount of data. To recapitulate, real-time enabled reporting on strategic level allows actions on market changes, rather than reactions. In regards of reducing action time, especially on strategic and tactical level, enables the organization to reach an unprecedented level of effectiveness as the business value of the decisions is potentially high [Ec07].

3.2 Store Level Requirements

A store manager shall be supported in regards of decisions, concerning daily business. For the process execution, hence on intra-minute operation level, an event-based support with alerting capabilities is mandatory. Bake-off units, equipped with embedded systems could do this for example. As this is not primarily the aim of an analytic system, it is out of scope of this work. However, intraday monitoring is essential for analyzing the process, by an appropriate visualization near real-time.

In order to provide automated order recommendations, a sufficient amount of historical data has to be taken into account to satisfy the appropriate statistical calculation on time series. Additionally, location related and environmental information increases the accuracy of the forecasting model. Environmental variables, like historical weather and holidays, have to be considered in correlation with historical process data, to improve the results of the appropriate machine-learning algorithm. Furthermore, forecasted weather data and upcoming holidays should be included to ex-ante data as well, in order to improve the goodness of prediction. Model fitting and operational data analysis shall be executed ad hoc and on demand, by querying the system. This way, the machine learns automatically, by having incorporated the latest demand developments. The store manager, does not have to know the model behind, he is merely interested in reliable information, which helps him to deal with the uncertainty. Besides the calculated demand forecast for the upcoming three days, the store manager should be provided with additional relevant information, enhancing the recommendation. This information could be the weather forecast, upcoming public holidays and in this connection, the demands of the opening days in the past years, mapped to public events.

3.3 Prototype Architecture

The SAP HANA in-memory appliance software is currently released in SPS 05 [SD13]. Important peripheral technologies have been integrated, such as the SAP UI5 Framework for the presentation layer and the SAP Extended Application Services (XS Engine), a lightweight application layer. Consequently, the web based graphical user interface is built, using the SAP UI5 framework. The front-end implementation conducts the model view controller (MVC) pattern, as provided by the framework. The XS engine provides a tight integration of the presentation layer to the SAP HANA database. Desktop web browser and mobile versions of the application are based on two different view variants, reusing standard UI items and especially the sap.viz library for dashboard visualization.

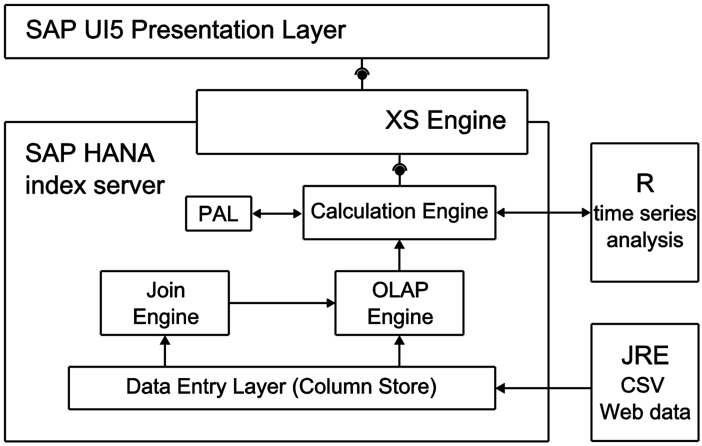

Figure 3 – Architecture

The used architecture, as described in the following is summarized in Figure 3.

The different management roles are distinguished by the specific HANA user roles, such as the strategic, tactical and operational management role. The strategic and tactical roles are showing almost the same reports, restricted by the aggregation level, related to one's responsibility (e.g. the region). On operational level, completely different reports are provided, mainly focused on daily analysis, and similarly restricted by the appropriate store. Particularly, order recommendations for the next three days and each product are provided, plus event related information (see requirements section 3.2). The intraday bakery process is visualized in addition, helping the store manager in his daily process analysis.

Through the tight integration of the database, the presentation layer profits of the advantage of a high abstraction level. The data-binding feature of the OData services is especially beneficial for tactical reporting. Flexible data navigation for the tactical management user is provided hereby, by selecting free time intervals and breaking down into different products, regions, or stores.

Data provisioning

A custom Java import module, using the JDBC API, handles the load of historic and recent transactional data. The reason for this implementation mainly relies on huge amount of heterogeneous, CSV formatted files, representing historical process data. Therefore, a special bulk load strategy has been pursued, especially in spite of the insert properties of column-oriented tables in the data entry layer. The automatic delta merge functionality of the SAP HANA database [Si12] could be exploited to its full extend. Furthermore, historic weather data, as well as weather forecast and holiday data, is loaded via the import module. The historical weather data has been imported from the web weather API ‘wunderground.com’. The near real-time data provision is invoked by periodic Java cron jobs, providing transactional data as well as weather forecast data, both applied in the same way as historical data.

3.4 SAP HANA data management

The HANA in-memory database is the core technology of this observation. In the following section, the data warehouse model will be shortly discussed.

The data entry layer consists of one fact table containing minutely measures derived from the bakery process. This fact table has a table cardinality of approximately two billion records. An appropriate partitioning policy, based on time related range partitioning is conducted here. Several master data tables contain information about stores, regions, products, and holidays. For model training of the forecast module, the corresponding time interval of the observation period of daily, city wise consolidated store data was called from the wunderground.com API. Consequently, the historical weather data is stored in an appropriate table, whereas weather forecasts are stored separately, being refreshed in full loading mode, daily. Since the historical data table contains only ex-post weather data, it is updated daily. All tables are implemented as column tables.

Upon this data entry layer, several attribute views (i. e. the Join operator) are implemented, building up the product, store and regional dimensions. The time dimension is based on the generated time table with minutely level of granularity, provided by HANA standardly. The OLAP operator is represented by one analytical view upon the fact table and its dimensions, respectively the attribute views.

Accordingly to the data calculation graph model (see section 2.4), calculation views, based on the OLAP operator (respectively the analytic view), are implemented to satisfy complex user reporting scenarios, showing availability, loss and sales on tactical and strategic level in different timely relations. Both, the graphical, as well as script based variants are utilized. Calculation views based on graphical representation, profit from enormous performance advantages, as they exploit completely the calculation engine optimizations. Unfortunately, graphical calculation views suffer from a limited expressiveness; this also applies to calculation engine plan operator statements (CE) in the SQL script variant. Whereas the usage of SQL statements in SQL script allows expressiveness, by decreasing significantly the performance.

Predictive Analytics

PAL is used for more sophisticated data mining on tactical level, as well as preprocessing of time series data. Therefore, several SQL Script procedures encapsulate the appropriate PAL calls, by passing parameters, and table types. Specifically the linear regression model function is used to draw trends of dynamically aggregated sales data over time. Furthermore, the anomaly detection function has been used for outlier detection in daily sales data. Following the calculation graph model, PAL would represent the SAP custom content node, while R script calls for demand forecast model execution, constitutes the R nodes. Time series are gathered on invocation, enhanced by additional predictor variables, like holidays and weather information, and passed as a data.frame object to the R environment. Model training and parameter fitting can be processed extremely efficiently, as R supports parallelization as well. Particularly, two different forecast models have been implemented for comparison reasons. One forecasting model applies the Auto Regressive Integrated Moving Average (ARIMA) approach, another an Artificial Neuronal Network (ANN) model.

ARIMA based forecast

An automated ARIMA model has been implemented in R script. The used package is mainly the forecast package available at Comprehensive R Archive Network (CRAN). The automated ARIMA fitting algorithm auto.arima() has been used for this project purposes, which is based on the Hyndman et al. algorithm [HK07].

ANN based forecast

Alternatively to the ARIMA approach, an Artificial Neuronal Network model, based on the RSNNS package has been implemented, particularly for capturing automatically nonlinear time series shapes. As expected in the food retail context, ANN is supposed to deliver more accurate forecast results [Do06].

3.5 Results

The focus of the case study was to prove, that the lean architecture approach of the SAP HANA appliance software could meet the analytic requirements of an advanced operational BI system. Mixed workload handling and different analytic alignments for such a system have been identified and verified. As a must, the response times of the different querying demands were expected to have a maximum of three seconds. Consequently, a user centric development has been pursued. Furthermore, the required development and maintenance effort for the observed scenario has been reviewed. For the given data population of two years, bake-off process data, enhanced by environmental data, the following observations have been made:

Analytic applications

All analytic queries are invoked on demand, by traversing the calculation graph, thus pulling information from the underlying column store tables with a granularity, as derived directly from the transactional level. In the off-load observation, a multidimensional data model has been implemented. In this context, the mixed workload handling could be confirmed for both: (near) real-time update and general query performance of less than three seconds user response times. It can be stated, that for especially time series calculations with the current R implementation, a linear growth in running time can be expected with the increased length of the time series intervals. Alternative developments of the proprietary PAL functionality could substitute the R environment to some extent. PAL has the advantage of processing data directly in the HANA database, but is very limited in its functionality and operability in the current stage of maturity. R profits of a huge community, contributing to this open source software, especially in the world of academia, therefore it is difficult to compete for PAL with R. With the calculation graph model introduced by Sikka et al. [Si12] these technologies could be used in a satisfying interaction, considering the trade-off between performance and analytic capability.

Development and maintenance

As stated above, all analytic requirements rely on only a few physical column tables, the main effort consists in providing different views on this data at run time, by modeling the calculation graph. In common ETL-centric data warehouses, business logic is often implemented in the different steps of the ETL process, to provide a specific business view in an iterative manner, with several intermediate, persistent aggregation states. This increases the complexity in addition, as business logic is mixed up with consolidation and transformation of the data, increasing the data and analysis latency, especially caused by look-ups between the different data marts. Real-time data provision and maintenance could therefore only be established at high costs, concerning experts with tacit knowledge and high manual effort. Compared to traditional disk-based data warehousing approaches, the described implementation is a completely new way of designing an analytic system. This approach is extremely beneficial as the different nodes in the calculation model can be reused in different contexts, by modularizing the implementation of business logic, decoupled from data persistency management tasks.

The usage of the calculation graph implementation paradigm was mainly developed implicitly, as the graphical representation, presented by Sikka et al. [Si12], is currently not available for external development. The design intention of this implementation framework is clearly noticeable, and absolutely needed. As the business logic is handled completely by the data layer, the application layer serves as a thin dispatching layer, providing an efficient way to present the data in the SAP UI5 application. Here, the framework unifies the latest standard web technologies (i.e. HTML5, JSON, OData and JavaScript etc.), allowing a focus on mobile applications. Thus, the complex analytic capability resides completely in the back-end allowing fast querying anytime and anywhere, with an appropriate mobile device and a standard mobile internet connection.

4 Conclusion and outlook

In this paper a case study in the field of fast moving goods of a large food retailer has been observed. The prototype has clearly highlighted the utilization of potentials in a real-time data warehouse architecture, based on in-memory data management. A novel operational BI understanding can be derived, by the possibility to provide real-time information to all organizational management groups. A common argument is that strategic decision making does not has to be based on real-time information, as the decision horizon is typically long. Contrary to this argument is a survey on operational BI, stating that the largest user group of 77% are managers, followed by business analysts (65%) and executives (48%) tracking intraday operational dashboards against their objectives (see [Ec07] p. 8). It is obvious, that 'long service paths' could be avoided, by delivering seamlessly updated and integrated information, without decoupling strategic from tactical and operational information consumption. Reducing action time and integrating different data mining capabilities on one common data foundation helps to achieve competitive advantage, by simultaneously decreasing complexity of the analytic system. In the observed use case, the action time of all management groups equals the interaction time with the application with neglectable data and analysis latencies, thus minimized to the decision time of a human operator. These characteristics are the major differentiators of the described in-memory appliance architecture to common, disk-based operational BI architectures.

Further observations have to be made regarding the data lifecycle management. In order to provide a reference architecture for advanced operational BI, the handling of historical transactional data that is beyond the scope of real-time analysis, should be clarified. As the in-memory appliance capacity is finite, even with advanced compression, a trade-off evaluation between the business value of very old transactions and cost has to be examined. Another field of interest in operational BI, would be a survey on the integration of analytics in a direct reporting scenario, running on a system that supports OLTP and OLAP as propagated by Plattner et al. [Pl09], especially how this would affect the embedded data warehouse architecture. Consequently, automated decision-making, based on business rules and predictive analytics could be investigated, specifically in a 'self-learning' operational system, enhanced with analytic capabilities in a closed-loop information flow.

Bibliography

[Do06] Doganis, P. et al.: Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. In Journal of Food Engineering, 2006, 75; pp. 196–204.

[DT99] Datta, A.; Thomas, H.: The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. In Decision Support Systems, 1999, 27; pp. 289–301.

[Ec07] Eckerson, W. W.: Best practices in operational BI. In TDWI, Renton, WA, 2007.

[GM95] Gupta, A.; Mumick, I. S.: Maintenance of materialized views: Problems, techniques, and applications. In Data Engineering Bulletin, 1995, 18; pp. 3–18.

[Ha04] Hackathorn, R.: The BI Watch Real-Time to Real-Value. In DM REVIEW, 2004, 14; pp. 24–29.

[HK07] Hyndman, R. J.; Khandakar, Y.: Automatic Time Series for Forecasting: The Forecast Package for R, 2007.

[In93] Inmon, W. H.: Information architecture for the 90's: legacy systems, operational data store, data warehouse. In PRISM Tech Topic, 1993, 1; pp. 118–124.

[Kr11] Krueger, J. et al.: Fast updates on read-optimized databases using multi-core CPUs. In Proceedings of the VLDB Endowment, 2011, 5; pp. 61–72.

[Kr12] Krueger, J. et al. Eds.: Leveraging Compression in In-Memory Databases, 2012.

[OLW08] Olafsson, S.; Li, X.; Wu, S.: Operations research and data mining. In European Journal of Operational Research, 2008, 187; pp. 1429–1448.

[Pl09] Plattner, H. Ed.: A common database approach for OLTP and OLAP using an in-memory column database. ACM, 2009.

[Re11] Richard, C.; et al. Eds.: The mixed workload CH-benCHmark. ACM, 2011.

[SA13] SAP: SAP HANA Appliance Software SPS 05. Document Version: 1.0 - 2013-03-01. http://help.sap.com/hana/hana_dev_sqlscript_en.pdf.

[SD13] SAP; Development: SAP HANA Appliance Software SPS 05. Document Version: 1.2 - 2013-03-01. http://help.sap.com/hana/hana_dev_en.pdf.

[Si12] Sikka, V. et al. Eds.: Efficient transaction processing in SAP HANA database: the end of a column store myth. ACM, 2012.

[SP13] SAP; PAL: SAP HANA Predictive Analysis Library (PAL) Reference. SAP HANA Appliance Software SPS 05. https://help.sap.com/hana/hana_dev_pal_en.pdf.

[Wa06] Watson, H. J. et al.: Real-time business intelligence: Best practices at Continental Airlines. In Information Systems Management, 2006, 23; pp. 7–18.

[Wh05] White, C.: The next generation of business intelligence: operational BI. In DM Review Magazine, 2005.