D. Demmler, D. Krupka, H. Federrath. (Hrsg.): INFORMATIK 2022,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2022 1093

Implementation Aspects for Linking Data in Hybrid Quantum Applications

Markus Zajac[1] Michelle Dang[2] Uta Störl[3]

Abstract: Linking logically related data for applications in production and logistics remains a challenge. Links between logically related data are difficult to establish at the time of collection. Moreover, executing searches for the right data often requires a full collection scan. In this short paper, we address implementation aspects of an already designed hybrid system that uses a quantum search algorithm for linking data.

Keywords: Hybrid polystore systems; Quantum computing applications; Data linkage; Quantum search; QRAM

1 Introduction

Managing and analyzing production and logistics data remains a challenge. One of these challenges concerns the interconnection of data [MSN17, Wa18]. Data, in particular machine and sensor data, data from production and logistics processes, product and parts data and manufacturing orders is collected and stored in a decentralized manner (more precisely in the different types of database systems). This causes a loss of connection to other stored data records, although there is a logical connection between them. However, when implementing industrial use cases, it is important to link logically related data, so this circumstance must be mastered first. This can lead to simple data linkage search queries having a worst case complexity of $O(n)$ each time (assuming n records have to be searched). Search queries can be significantly accelerated by secondary indexes. However, using secondary indexes for faster data access to distributed data is not always trivial and can have disadvantages. The various data-driven services of the use cases with their individually tailored data views, require the use of many different secondary indexes in a distributed data storage. In addition to this organizational aspect, maintaining secondary indexes poses another problem. Index structures must be maintained during write operations while ensuring consistency between secondary indexes and data tables [QCH18]. In [ZS22], we describe a typical use case in the industry, investigate the reason for this runtime complexity, and propose a new approach to data linking. The approach forgoes the use of secondary indexes and rethinks the existing problem for linking logically related data. It conceives itself in the sense of PISQ (Perfect

[1] University of Hagen, Databases and Information Systems, Hagen, Germany markus.zajac@fernuni-hagen.de
[2] University of Hagen, Hagen, Germany michelle.dang@studium.fernuni-hagen.de
[3] University of Hagen, Databases and Information Systems, Hagen, Germany uta.stoerl@fernuni-hagen.de

doi:10.18420/inf2022_92

Intermediate-Scale Quantum computing) [BSA21]. Central components of the approach include an interface between classical and quantum data (here we specifically take up the QRAM idea) as well as Grover's quantum search algorithm. The goal is to establish connections between logically related data on demand using appropriate queries, each with sublinear time. This short paper describes some aspects of the implementation of the concept described in [ZS22]. Section 2 briefly reviews the relevant work. Section 3 addresses implementation aspects before Section 4 describes a summary and next steps.

2 Related Work

Using the principle of a quantum superposition one showed, that a database with N numeric entries ($N = 2^n$, where n bits are needed to store the numeric entries) can be searched using $\sqrt{N}$ queries [JK20]. At the beginning of the Grover's quantum search algorithm, a sufficiently large register is prepared in an equal superposition of all states

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle. \tag{1}$$

Then the so-called Grover iteration

$$G = (2|\psi\rangle\langle\psi| - I)O \tag{2}$$

is applied $O(\sqrt{N})$ times [NC11]. The element searched for is specified via the oracle O. The search term is to be understood as abstract: It refers to a solution space and not to search in database systems. One way to provide quantum algorithms with input data is the so-called Quantum Random Access Memory (QRAM) [KH20, LB20, BSA21]. For our work we take up the idea of QRAM. We consider the following works as important or motivating for us. Papers [We20] and [We21] present data encoding techniques for quantum computing and propose encoding for use with QRAM. Input data can also be realized via quantum circuits instead of QRAM. In [Ka21], a two-qubit search is implemented using this method. There, searching for a specific *value code* (within unsorted values) returns an index. Even though this is an example for teaching purposes, it should still be applicable in other contexts. Paper [SW19] is particularly interesting because it relates quantum-based search problems using a hybrid approach. For the realization of a Recommendation System, a database with over 12,000 entries is used. The entire relational database is however not mapped as a quantum system, but only its relevant features. These correspond to primary keys of a relational classic database. On the reduced quantum database, Quantum k-NN and Grover's algorithm are applied. In [AMA18] and [AMA19] the authors proved the effectiveness of Grover's algorithm in NoSQL databases in a theoretical study and conclude that it can significantly reduce search time compared to classic algorithms. These database systems are increasingly used for storing data in production and logistics. The embedding of quantum technologies in database architectures is also discussed in [Gr21]. There, the *SHM3P* database is presented as an alternative of an IoT database for which an embedding is envisioned.

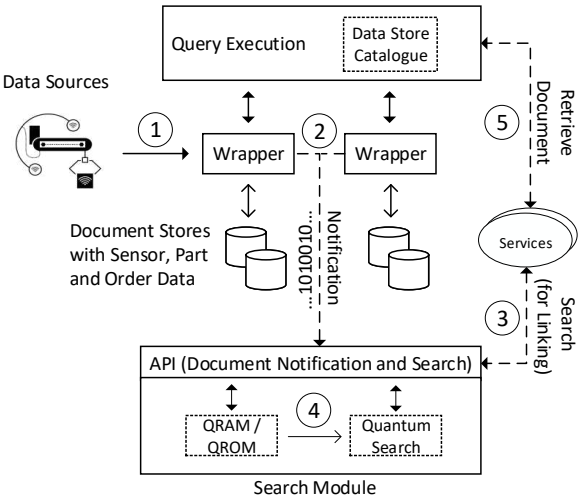

Fig. 1: Use case's hybrid polystore architecture (following [ZS22])

3 Implementation Aspects

In implementing the concept [ZS22], we have so far addressed two issues: First, data acquisition and storage in QRAM/QROM, and second, adaptations of the Grover's algorithm to exploit information from QRAM/QROM for data linking (see Figure 1). We address these issues in the following.

Test data and polystore system: For our use case, we construct two document stores (MongoDB instances) as a minimal example of a polystore system. The document store *Container* provides sensor information from shipping containers and the document store *Order* stores the orders of customers. As reference for the container data, we use an extracted and anonymized dataset which was published by the company DB Schenker under the CC BY 4.0 license. To begin with, we will focus on data from the light sensor. Sensor values are collected at different times and are stored with further information (like time, coordinates etc.) in JSON documents in document stores. To make the light sensor values accessible for quantum search, they are first encoded together with the IDs of the documents as a bit sequence, transmitted to the *Search Module*, and stored in a quantum memory (QRAM/QROM) in the order of their arrival. For a search hit, the corresponding document ID is to be delivered (see following explanations).

Quantum Random Access Memory (QRAM): QRAM can be understood as an interface for the exchange of classical and quantum data [KH20]. In QRAM multiple addresses can

be queried by querying them in superposition. This process can be expressed by means of a so-called LOAD function as follows [We21, MGM20]:

$$\frac{1}{\sqrt{k}} \sum_{i=0}^{k-1} |a\rangle_i |0\rangle \xrightarrow{LOAD} \frac{1}{\sqrt{k}} \sum_{i=0}^{k-1} |a\rangle_i |s_a\rangle \tag{3}$$

The first register (address register) is in a superposition of k queried memory addresses, the second register is the output register in superposition. Thereby $|a\rangle_i$ specifies the memory address and $|s_a\rangle$ represents the content at the i-th location. Algorithms, such as the Grover's algorithm, use the loaded data as a starting point. There are various QRAM concepts with their specific advantages and disadvantages. The authors find it rather difficult to make a general commitment to one concept. Based on a use case, the advantages and disadvantages must be weighed up. A general distinction is made between two types of QRAM: an "explicit" and an "implicit" QRAM (in the case of implicit memory, one also speaks of QROM) [MGM20]. We'll take a brief review of the two types of memory. In "explicit" QRAM, data is stored in real physical qubits. These data are read out or queried via additional circuits. The main disadvantage is the high requirement for logical qubits, which is $O(2^n)$ for n-bit addresses [MGM20]. The main advantage is the fact that the circuit of this memory type only needs to be compiled and optimized once.

In QROM, data ("database") is stored directly in a circuit itself, without the need for explicit qubits to store data. The main disadvantage is that content of the memory must be known in advance, the main advantage is that QROM circuits can be well optimized. Simple circuits with large depth and small width needs $O(n)$ qubits for n-bit addresses [MGM20]. QROM can be used in the sense of a lookup table. Today, however, there are still no efficient implementations of these concepts [LB20]. Furthermore, it can be assumed that the memory must be actively error corrected for both types. Nevertheless, the community mentioned further below offers several reference implementations. We evaluate these implementations for use with the Grover's algorithm. However, the high demand for qubits of the "explicit" QRAM didn't allow us to construct a reasonable minimal example. Therefore, we are currently evaluating QROM implementations. For the use case itself, the QROM idea is rather not optimal in the opinion of the authors, but still usable as an interface. Incoming sequences can either be buffered to generate an optimal QROM circuit ("database") at certain times. If incoming sequences should be considered directly, then a recompilation and optimization of the circuit would be necessary each time. [BSA21] also postulates that initially a "good" QROM realization is sufficient.

Grover's algorithm: Besides the type of memory, we also consider how the data should be represented in it. For the use case, we need to encode $(document\ ID, light\ value)$ tuples as bit sequences in memory. To achieve this, we represent a tuple as a bit sequence, which logically consists of two bit sequences of defined length: The first sub-sequence represents *document ID*, the second *light value*. In terms of the Grover's algorithm, bit sequences correspond to states in the solution space (there is a maximum of 2^s distinguishable states when sequences consist of s bits). Before the algorithm is executed, the sequences are

loaded into the above-mentioned register $|s_a\rangle$, which is in a superposition (according to the start condition in the Equation 1). A reference implementation of the algorithm has to be modified in such a way that the Grover oracle marks the solution (the state corresponding to a sequence) whose sub-sequence corresponds to the searched light value. In this state, both the document ID and the searched value are encoded. Based on this ID, the corresponding document can be retrieved from the polystore system mentioned above. Applications can search for different values and in this way link the logically related data on demand.

Technologies used and first experiments: We use the open source Quantum Development Kit (QDK) with the language Q# and QRAM reference implementations of *qsharp-community* on GitHub[4]. We have already started to use the IonQ quantum computer[5] for experiments with small data sets to test Grover's algorithm Oracle implementations. On the one hand, we intended to ensure that the code is not only executable on the IonQ simulator, but also on a machine. This has something to do with the fact that QPUs are more constrained (they are subject to so-called profiles), while a simulator is more tolerant. Secondly, we wanted to observe statistical deviations and tendencies of a NISQ machine. In the tests, we used sequences consisting of four qubits in order to search for specific values there. For statistical relevance, 1,000 resp. 10,000 shots were performed. We found strong deviations especially when multiple search hits are expected (i.e., when searched values are present multiple times).

4 Summary and Future Work

In this short paper, we address implementation aspects of a novel approach to fast linking logically related data on demand. In addition to interfaces such as QRAM/QROM, the Grover's algorithm is used for this purpose. At the same time, this contribution represents lessons learned. In our experience, implementation ideas or concepts and existing reference implementations do not always accompany each other smoothly. This in turn helps us to develop new ideas and suggest possible improvements. In the next steps, we continue to evaluate QROM as an interface, as well as other data encoding variants. We also develop a method to compare the asymptotic runtime of hybrid and classical systems.

Acknowledgments

This work has been funded by Deutsche Forschungsgemeinschaft (German Research Foundation) - 385808805.

[4] https://github.com/qsharp-community/qram

[5] Provided as part of the Azure Quantum Credits program

Bibliography

[AMA18] Amellal, Hicham; Meslouhi, Abdelmajid; Allati, Abderahim El: Reduce Data Processing Time in NoSQL Databases Based on Grover's Algorithm. In: Proceedings of the 3rd International Conference on Smart City Applications. SCA '18. Association for Computing Machinery, 2018.

[AMA19] Amellal, Hicham; Meslouhi, Abdelmajid; Allati, Abderahim El: Processing Unstructured Databases Using a Quantum Approach. In (Ben Ahmed, Mohamed; Boudhir, Anouar Abdelhakim; Younes, Ali, eds): Innovations in Smart Cities Applications Edition 2, Lecture Notes in Intelligent Transportation and Infrastructure, pp. 275–285. Springer International Publishing, 2019.

[BSA21] Bertels, Koen; Sarkar, Aritra; Ashraf, Imran: Quantum Computing - From NISQ to PISQ. IEEE Micro, 41(5):24–32, 2021.

[Gr21] Groppe, Sven: Semantic Hybrid Multi-Model Multi-Platform (SHM3P) Databases. In (Jain, Sarika; Groppe, Sven, eds): Proceedings of the International Semantic Intelligence Conference 2021 (ISIC 2021), New Delhi, India, February 25-27, 2021. volume 2786 of CEUR Workshop Proceedings. CEUR-WS.org, pp. 16–26, 2021.

[JK20] Jóczik, Szabolcs; Kiss, Attila: Quantum Computation and Its Effects in Database Systems. In: New Trends in Databases and Information Systems, volume 1259 of Communications in Computer and Information Science, pp. 13–23. Springer International Publishing, 2020.

[Ka21] Kain, Ben: Searching a quantum database with Grover's search algorithm. American Journal of Physics, 89(6):618–626, 2021.

[KH20] Kang, Yujin; Heo, Jun: Quantum Minimum Searching Algorithm and Circuit Implementation. In: 2020 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, pp. 214–219, 2020.

[LB20] Leymann, Frank; Barzen, Johanna: The bitter truth about gate-based quantum algorithms in the NISQ era. Quantum Science and Technology, 5(4):044007, sep 2020.

[MGM20] Matteo, Olivia Di; Gheorghiu, Vlad; Mosca, Michele: Fault-Tolerant Resource Estimation of Quantum Random-Access Memories. IEEE Transactions on Quantum Engineering, 1:1–13, 2020.

[MSN17] Maier, Alexander; Schriegel, Sebastian; Niggemann, Oliver: Big Data and Machine Learning for the Smart Factory—Solutions for Condition Monitoring, Diagnosis and Optimization. In: Industrial Internet of Things, Springer Series in Wireless Technology, pp. 473–485. Springer International Publishing, 2017.

[NC11] Nielsen, Michael A.; Chuang, Isaac L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, 10th edition, 2011.

[QCH18] Qader, Mohiuddin Abdul; Cheng, Shiwen; Hristidis, Vagelis: A Comparative Study of Secondary Indexing Techniques in LSM-Based NoSQL Databases. In: Proceedings of the 2018 International Conference on Management of Data. SIGMOD '18. Association for Computing Machinery, p. 551–566, 2018.

[SW19] Sawerwain, Marek; Wróblewski, Marek: Application of Quantum k-NN and Grover's Algorithms for Recommendation Big-Data System. In: Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT 2018. Springer International Publishing, pp. 235–244, 2019.

[Wa18] Wang, Junping; Zhang, Wensheng; Shi, Youkang; Duan, Shihui; Liu, Jin: Industrial Big Data Analytics: Challenges, Methodologies, and Applications. CoRR, abs/1807.01016, 2018.

[We20] Weigold, Manuela; Barzen, Johanna; Leymann, Frank; Salm, Marie: Data Encodinbg Patterns for Quantum Computing. In: Proceedings of the 27th Conference on Pattern Languages of Programs. PLoP '20. The Hillside Group, 2020.

[We21] Weigold, Manuela; Barzen, Johanna; Leymann, Frank; Salm, Marie: Expanding Data Encoding Patterns For Quantum Algorithms. In: 2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C). IEEE, pp. 95–101, 2021.

[ZS22] Zajac, Markus; Störl, Uta: Towards quantum-based Search for industrial Data-driven Services. In: Proceedings of the 2022 IEEE International Conference on Quantum Software (QSW). IEEE, 2022.