Encoding and Provisioning Data in different Data Models for Quantum Computing

Markus Zajac[1]

[1]*Supervised by Prof. Uta Störl*
FernUniversität in Hagen, Universitätsstraße 47, 58097 Hagen, Germany

Abstract

Quantum computers promise polynomial or exponential speed-up in solving certain problems in comparison to classical computers. In practical use, however, there are currently a number of fundamental technical challenges. One of them concerns the loading of data into quantum computers and the corresponding encoding, since they cannot access database systems directly. In this paper, a hybrid data management architecture in which databases can serve as data sources for quantum algorithms is presented. Furthermore, a first encoding approach for data organized in tree structures is given.

Keywords

Data management for quantum computing, hybrid quantum computing, data encoding, data loading

1. Brief Introduction from a Database Perspective

Quantum computers have the potential to perform certain calculations much faster than classical computers. They can be used in various application areas, such as optimization, machine learning or search algorithms, to name just a few examples [1, 2]. Depending on the problem, a polynomial or exponential acceleration can be assumed compared to a classical computer [3]. It is foremost a mathematical superiority. This is because general challenges must be overcome when embedding quantum computers in database landscapes or software architectures. The main reason for embedding is that data from data-driven use cases and parameters to be processed to compute solutions on quantum computers are nowadays managed in database systems. The following two challenges are associated with embedding:

Challenge 1: Quantum computers cannot directly access data and information from database systems [4]. However, quantum algorithms assume that their data is already accessible in a suitable form [5].

Challenge 2: Data, which exist in different structures and models, must be encoded accordingly before they can be used on a quantum computer. The efficient encoding of data is also a challenge [5, 6] and a subject of research [1]. The reason is the high time consumption of the corresponding encoding routines, which is exponential in the worst case [4, 7].

Problem definition: The overall problem is to provide classical data in different models for use on quantum computers. The provision goes hand in hand with the encoding of the data.

In general, the following can be said about today's quantum computers: The current generation of quantum computers (NISQ–Noisy Intermediate-Scale Quantum) are characterized as noisy and limited in scalability [8, 9]. This means that the number of qubits (the smallest information units of a quantum computer) is limited, the qubits themselves are error-prone, and operations on qubits deviate a bit from the expected result. However, research is being carried out on improved hardware [10].

Regardless of these current limitations, quantum-supported database concepts are being developed, but these are to be understood as concepts and are out of the question for operational use today. However, this shows that there is interest in quantum technologies and their use in the database context. Some of these works are the following. The future support of database management systems by quantum computing is mentioned in [11]. In [12] mention is made of a hypothetical "quantum database" and *Quantum Query Language*. In [13], possible applications of the Grover algorithm in database systems for set operations are described. The advantage of the Grover algorithm in NoSQL databases was theoretically proved in [14].

In contrast, the overall solution to be worked out in the PhD thesis should be feasible and not purely conceptual or visionary. This is presented in the remainder of the paper. Section 2 firstly discusses related work on data provision and encoding. Section 3 then presents the research direction of the PhD thesis thesis. Section 4 presents the conclusion.

VLDB 2023 PhD Workshop, co-located with the 49th International Conference on Very Large Data Bases (VLDB 2023), August 28, 2023, Vancouver, Canada

markus.zajac@fernuni-hagen.de (M. Zajac)
0000-0002-9338-9259 (M. Zajac)

2. Related Work

It can be generally stated that research on the use of quantum computers around database systems and data-driven applications is rather rare. As outlined in Section 1, work on this may well be visionary. However, in order to concretely use data in a quantum computer, that data must be loaded by encoding it into the state of the qubits [4]. In paper [6] it is generally mentioned that loading data into a quantum computer is a hard problem. The fact that databases can provide data for quantum computing is almost not discussed in the literature. The same applies to possible data exchange procedures between database systems and quantum computers. The paper [15] deals with the orchestration of hybrid quantum applications in the cloud using BPMN-Workflows and uses a database as input for a specific use case. In [5], there is mention of the need to transform data from databases for quantum computing.

The next important topic concerns encoding. Encoding represents the transformed classical data by means of qubits [5]. Only then can the data be processed on a quantum computer. In [4, 16, 17], various encoding procedures are explained, which are understood as patterns. These patterns are self-contained, reusable building blocks that are to be reused in the construction of quantum algorithms. The fit of these patterns to real-world data-driven use cases is not discussed. The use case data itself can be organized in different data models or structures. For example, they can be available in relations, in documents (as semi-structured data), or as labeled/unlabeled trees/graphs. The papers do not address how these logical structures can be encoded.

3. Research Direction

Based on the challenges of Section 1 and the related work in Section 2, the following research questions arise for the PhD thesis:

Q1: How is a framework for exchanging data between database systems and quantum computers?

Q2: How can data in different data models and structures be encoded using qubits?

Comments on Q1: Previous work [15] uses a BPMN workflow engine for enterprises to orchestrate a variety of tasks. In addition, various other tools and programming languages are used, which form a complex mesh in the end. The framework of this thesis is intended to reduce complexity. A uniform programming language and alternative workflow options are to be evaluated for this purpose. The *service layer pattern*[1] can be considered as an example for this. In addition, event-driven workflows are to be supported and economical data exchange between database systems and quantum computers is to be explored (cf. Subsection 3.1). Works describing communication between database systems and quantum computers considering these aspects are not known so far. Initial design of this architecture is described in Subsection 3.1. However, it is a work in progress and will be continuously developed.

Comments on Q2: For developed encoding methods a runtime and space complexity analysis is performed. The first analyzes the runtime of the encoding methods, the second the demand for qubits. This task is described as challenging [5]. In this way, methods can be compared with each other. In Subsection 3.2, an encoding approach for a tree structure is conceptually developed, which, however, has not yet been experimentally verified.

3.1. Data Exchange Framework–A Hybrid Data Management Architecture

In this Section, we introduce the Hybrid Data Management Architecture, which acts as a data exchange framework. A first conceptual idea of the architecture was published in [18]. Figure 1 shows the schematic structure of the architecture. It can be roughly divided into two parts: On the left, a classical system with *Applications*, *Coordinator*, and *Databases*, and on the right, *Quantum Circuits* that process data using quantum hardware or simulators. Quantum hardware may well be provided as a service in the cloud.

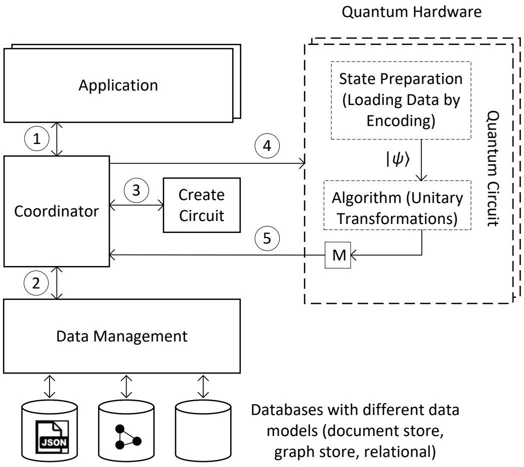


Figure 1: Hybrid Data Management Architecture

Firstly we consider the classical system using the example of a workflow. For example, an *Application* should solve an optimization problem or classify data, to name

---

[1] Architecture Patterns with Python. https://www.cosmicpython.com/book/chapter_04_service_layer.html

just two examples. The *Coordinator* is responsible for data exchange and communication between the components of the classical system and the quantum computer. With the help of the *Coordinator*, the following procedure results: **(1)** An *Application* submits a request to the *Coordinator*. The *Coordinator* initially verifies whether a problem should be solved by means of a quantum algorithm on a quantum computer in order to process the request. The *Coordinator* notifies the *Application* of the decision. We assume in the following that a quantum computer will be used. Otherwise, the data is retrieved from the *Databases* and made directly available to the *Application*. The problem will be solved classically. **(2)** The required data is extracted from the *Databases* and pre-processed if necessary. **(3)** To solve a problem, a corresponding *Quantum Circuit* (see below) is generated. **(4)** The generated circuit is transmitted to a quantum computer and execution is initiated. **(5)** The *Coordinator* fetches the achieved result of the algorithm and provides it to the *Application*. The *Application* itself is then in charge of post-processing.

Next, we look at the structure of a *Quantum Circuit*. A *Quantum Circuit* can be roughly divided into the areas of *State Preparation* and *Algorithm* [4]. The *State Preparation* block is responsible for encoding, which means that data and, if necessary, parameters are loaded and encoded in a quantum state. This quantum state forms the starting point for the actual algorithm in the *Algorithm* block, which can manipulate the initial state. The execution of an *Algorithm* ends with the measurement of the final quantum state, which represents the result.

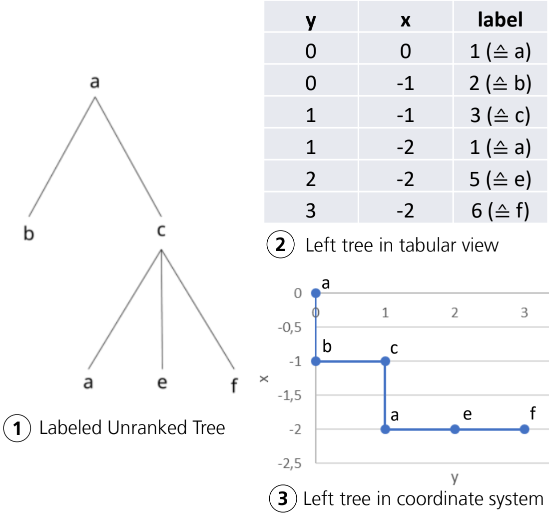
Finally, we would like to discuss some aspects of the realization of the data exchange in Figure 1.

Event-driven workflows: Certain use cases assume that the relevant data is exchanged on demand. In this case, the *Coordinator* creates and transmits the circuit on request. In other use cases, changes may occur to the data in the *Databases* and new data may be added. In such a case, the circuit should also be updated immediately. In this scenario, a quantum algorithm operates on data that is always up-to-date. Both variants are understood as events. For (or instead of) the *Coordinator*, alternative approaches are being tested in this context, such as publish/subscribe messaging.

Data economy: Considering data restrictions in the architecture is crucial to quantum computing. More data requires more qubits and longer runtimes of state preparation routines. We propose to store data constraints in profiles to define only a reasonable minimum amount of data to be processed on a quantum computer. For a use case, different profiles can be used to manage, for example, different data value ranges or different node sets (to partition graphs from graph databases). In the NISQ era, this is of particular interest since hardware limitations allow only quantum circuits of certain size.

3.2. Encoding Approach using the Example of Labeled Unranked Trees

In this section, we sketch an idea of how labeled unranked trees could be encoded. Figure 2 shows a tree that is to be encoded. The source tree is converted into a relational form. The column y of the table represents the neighboring nodes, the column x represents the corresponding level. For the purposes of illustration, we represent the entries of the table in a coordinate system. There, the connections between the stages and neighboring nodes per level can be seen directly. Based on this, tree-pattern queries can be performed. David [19] describes other time-consuming problems on such trees. Suitable quantum algorithms can achieve a quadratic speedup.



y	x	label
0	0	1 (≙ a)
0	-1	2 (≙ b)
1	-1	3 (≙ c)
1	-2	1 (≙ a)
2	-2	5 (≙ e)
3	-2	6 (≙ f)

① Labeled Unranked Tree

② Left tree in tabular view

③ Left tree in coordinate system

Figure 2: Various representations of a labeled unranked trees. 1) Source tree 2) The same tree in tabular representation 3) Translation of the table into a coordinate system.

Using the coordinate system, we explain the concept behind encoding the tree in a quantum state (Figure 3). For encoding, each level is first encoded using the basic encoding method [17, 5] (A). For this, each level must be represented as a bit string b. This contains all the information of a level (like labels and node links). The basic encoding method encodes each level into a quantum state $|b_l\rangle$. To take advantage of a quantum computer, these states are used to generate a so-called superposition [5] (B), which encodes the tree in one state. A superposition state is the starting point of a quantum search algorithm.

The encoding approach can be regarded as general. First, logically related information is encoded via the base encoding before a superposition is generated. Intermediate formats (like a table in Figure 2) can be used to compose related information.

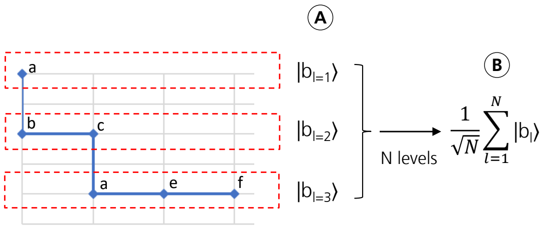


Figure 3: Encoding of the tree in the coordinate system from Figure 2. A) The bit encoding / base encoding is applied for each level. B) The total tree encoded in superposition.

4. Conclusion

In this paper, the research direction of the PhD thesis was presented. It covers two topics. Firstly, a hybrid data management architecture that makes classical data available to quantum computers using a suitable encoding. Secondly, the development of efficient encoding methods for classical data in different data models. The concept of data management architecture was briefly presented. Also, a first encoding approach for tree structures was provided.

Acknowledgments

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

References

[1] E. H. Houssein, Z. Abohashima, M. Elhoseny, W. M. Mohamed, Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision, Expert Syst. Appl. 194 (2022) 116512.

[2] V. Hassija, V. Chamola, A. Goyal, S. S. Kanhere, N. Guizani, Forthcoming applications of quantum computing: peeking into the future, IET Quantum Communication 1 (2020) 35–41.

[3] M. Schuld, F. Petruccione, Quantum Computing, Springer International Publishing, 2021, pp. 79–146.

[4] M. Weigold, J. Barzen, F. Leymann, M. Salm, Encoding patterns for quantum algorithms, IET Quantum Communication 2 (2021) 141–152.

[5] M. Kieferová, Y. Sanders, Assume a Quantum Data Set, Harvard Data Science Review 4 (2022).

[6] S. Herbert, Quantum computing for data-centric engineering and science, Data-Centric Engineering 3 (2022).

[7] N. Gleinig, T. Hoefler, An efficient algorithm for sparse quantum state preparation, in: DAC, IEEE, 2021, pp. 433–438.

[8] F. Leymann, J. Barzen, The bitter truth about gate-based quantum algorithms in the NISQ era, Quantum Science and Technology 5 (2020) 044007.

[9] M. Weigold, J. Barzen, F. Leymann, D. Vietz, Patterns for Hybrid Quantum Algorithms, in: SummerSOC, volume 1429 of *Communications in Computer and Information Science*, Springer, 2021, pp. 34–51.

[10] H. Riel, Quantum computing technology, in: 2021 IEEE International Electron Devices Meeting (IEDM), 2021.

[11] S. Groppe, Semantic Hybrid Multi-Model Multi-Platform (SHM3P) Databases, in: Proceedings of the International Semantic Intelligence Conference 2021 (ISIC 2021), volume 2786 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 16–26.

[12] I. Hamouda, A. M. Bahaa-Eldin, H. Said, Quantum databases: Trends and challenges, in: 2016 11th International Conference on Computer Engineering & Systems (ICCES), 2016, pp. 275–280.

[13] S. Jóczik, A. Kiss, 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*, Springer International Publishing, 2020, pp. 13–23.

[14] H. Amellal, A. Meslouhi, A. El Allati, 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.

[15] B. Weder, J. Barzen, F. Leymann, M. Zimmermann, Hybrid Quantum Applications Need Two Orchestrations in Superposition: A Software Architecture Perspective, in: 2021 IEEE International Conference on Web Services (ICWS), 2021, pp. 1–13.

[16] M. Weigold, J. Barzen, F. Leymann, M. Salm, Expanding Data Encoding Patterns For Quantum Algorithms, in: 2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C), IEEE, 2021-03, pp. 95–101.

[17] M. Weigold, J. Barzen, F. Leymann, M. Salm, Data encoding patterns for quantum computing, in: Proceedings of the 27th Conference on Pattern Languages of Programs, PLoP '20, The Hillside Group, 2022.

[18] M. Zajac, U. Störl, Towards quantum-based Search for industrial Data-driven Services, in: Proceedings of the 2022 IEEE International Conference on Quantum Software (QSW), IEEE, 2022.

[19] C. David, Complexity of Data Tree Patterns over XML Documents, in: MFCS, volume 5162 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 278–289.