HYBRID DATA MANAGEMENT ARCHITECTURE FOR PRESENT QUANTUM COMPUTING

Markus Zajac Databases and Information Systems FernUniversität in Hagen 58097 Hagen, Germany markus.zajac@fernuni-hagen.de Uta Störl Databases and Information Systems FernUniversität in Hagen 58097 Hagen, Germany uta.stoerl@fernuni-hagen.de

March 20, 2024

ABSTRACT

Quantum computers promise polynomial or exponential speed-up in solving certain problems compared to classical computers. However, in practical use, there are currently a number of fundamental technical challenges. One of them concerns the loading of data into quantum computers, since they cannot access common databases. In this vision paper, we develop a hybrid data management architecture in which databases can serve as data sources for quantum algorithms. To test the architecture, we perform experiments in which we assign data points stored in a database to clusters. For cluster assignment, a quantum algorithm processes this data by determining the distances between data points and cluster centroids.

Keywords Data management for quantum computing · Hybrid quantum computing · Data loading · Data encoding

1 Introduction from the 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, 8, 6]. Depending on the problem, a polynomial or exponential acceleration can be assumed compared to a classical computer [20]. It is foremost a mathematical superiority. However, for practical use, certain hurdles must be addressed.

First of all, the current generation of quantum computers (NISQ) can be termed as noisy and limited in scalability [12, 24]. This means that the number of qubits is limited, and the qubits themselves are error-prone. Next difficulty concerns loading data into the quantum computer. From the perspective of a database, data is managed in various database models that need to be processed to solve a particular problem or query. In principle, it may be interesting to perform this processing on a quantum computer for certain problems or queries. We introduce some computationally intensive queries later on. Quantum computers, however, cannot access databases directly [23]. Moreover, today's quantum algorithms assume its input data is already in the desired form [10]. The data to be processed has therefore first to be encoded in a suitable way in order to be used on a quantum computer at all. The encoding must reflect the structure of the data, for example, when data is organized hierarchically (in trees) or semi-structured (in documents). Efficient encoding of data is a challenge [10, 7] as well as a future research direction [8]. Last but not least, workflows and interfaces for data exchange between classical systems (such as applications but also databases) and quantum computers are an issue. An important design goal of such hybrid systems should be the reduction and manageability of complexity. Our contributions are therefore the following:

 In this vision paper, we describe a hybrid system that enables the exchange of data between applications and databases and quantum computers and call it Hybrid Data Management Architecture (HDMA). The architecture acts as a framework for researching and piloting appropriate encoding methods for data in different data structures and models. It is also a framework for designing future data-centric applications for quantum computing.

2. We validate an early prototype of this architecture on an example.

The remainder of the paper is structured as follows: In Section 2 we discuss related work. In the following, we describe the HDMA (Section 3) and in Section 4 its proof of concept. In doing so, we use a quantum distance estimation algorithm. Section 5 provides a summary and outlines next steps.

2 Related Work

We review work on computationally intensive database queries, quantum technologies in the database environment, data encoding, and data exchange.

Computationally intensive queries. We first addressed the question of whether data models and corresponding queries exist that are of interest for quantum computing. We first look at queries that are computationally intensive and therefore represent interesting objects of investigation to determine whether quantum computers can generally enable their acceleration. As an example, data organization in labeled trees (like XML trees) can be given here. David [2] describes an NP-Complete problem involving data tree patterns as a query language for such trees. Gottlob et al. [5] discuss the complexity of *XPath* queries (used for node selection) on such trees. These queries exhibit a polynomial runtime, and for some the polynomial degree can be 4 or 5. Another NP-hard problem is the keyword search in (graph) databases, which is related to the so-called *Group Steiner Tree* problem [14]. This non-exhaustive list of examples shows that interesting query candidates exist for quantum computing.

Quantum technologies in the database environment. In [1] quantum algorithms (such as *Grover*) and heuristics (such as *Variational Quantum Algorithms* or *Quantum Annealing*) for optimizing queries and transaction plans are presented. The development of hybrid algorithms for database problems is mentioned as a future research direction. Yuan et al. [27] also conduct a literature review on quantum computing for database problems (such as database search, database manipulation, or query optimization). In addition, the paper outlines a vision of a quantum-based multi-modal database. Jóczik and Kiss [9] describe some possible applications for database systems based on *Grover's* algorithm. In [13], the idea of a so-called data center with *Quantum Random Access Memory* (QRAM) is presented. Classical or quantum information should be able to be uploaded and downloaded to the center. This uploaded information can be processed, e.g. by suitable quantum algorithms. However, the authors assume that QRAM has been constructed in a fault tolerant manner and has been error corrected. Today, however, noise resilience and scalability represent major unsolved hurdles [17]. An implementation of a sufficiently large memory therefore seems to be impossible today [15].

In summary, the papers present visionary ideas, future research directions, and reviews of the literature. Our contribution is to develop not only concepts for the use of quantum technologies in the database environment, but also a hybrid architecture that can be implemented today to successively realize these concepts. The architecture is a framework for exploring and testing appropriate encoding methods for data in various data structures and models, and implementing certain queries using quantum computing. To the best of our knowledge, there is no other work that addresses a framework for realizing the above database queries using database systems or classical technologies and quantum technologies.

Data encoding. Encoding represents the transformed classical data by means of qubits [10]. Only then can the data be processed on a quantum computer. In [23, 25, 26], various encoding procedures are explained, which are understood as patterns. These patterns are reusable self-contained building blocks that can be reused in the construction of quantum algorithms.

As mentioned earlier, data can be organized in different data models and structural information must be preserved when encoding. A first idea of how to encode data organized in labeled trees was presented in [28]. We are not aware of any other work on encoding specific database data models. In general, the encryption methods we are developing are also intended to serve as a blueprint for future improved quantum hardware. For these, roadmaps are given, accompanied by appropriate research [19].

Data exchange. Another aspect concerns the exchange of data between classical components and quantum computers. Data wrangling, pre-processing, job management, post-processing, and feeding back the results to applications or a databases are individual processes here. The work of Weder et al. [22] deals with orchestration using *BPMN* workflows. The described approach with many steps and modeling aspects seems complex to us. It can be generally stated that more complex systems (such as so-called *data mesh architectures*) tend towards complexity and cause considerable additional expenses for their operation [11].

To reduce complexity, we propose a decentralized approach with a lightweight, asynchronous communication mechanism by adopting the microservice paradigm. Microservices are typically used to manage the increase in complexity of systems [4].

3 Architectural Description

This Section introduces the Hybrid Data Management Architecture (HDMA). We derive these from the introductory Section 1 and the considered work of Section 2. Figure 1 shows the schematic structure of the architecture. The classic components include a series of loosely coupled services that follow the microservice paradigm. These are the following services: *Decision Service, Circuit Service, Data Service, Backend Service* and *Result Manager*. Some services communicate with a *Gate-based Quantum Computer* that processes data to solve a task. A *Gate-based Quantum Computer* is characterised by its ability to be used for a wide range of problems. This type of quantum computer is based on quantum gates for encoding and processing data, which form a *Quantum Circuit*. Today, quantum computers are usually provided as cloud services. Instead of quantum computer, a simulator can also be used.



Figure 1: Hybrid Data Management Architecture [Source: Own representation]

We will first provide a brief overview of the services before focusing on communication. As motivation, we imagine an *Application* that would initiate a computationally intensive query or intends to solve another computationally intensive problem. The decision as to whether a problem is calculated using a *Gate-based Quantum Computer* or completely classically is made by the *Decision Service*. The data required to solve a problem, as well as the results of the calculations, are stored in *Databases* and managed by the *Data Service*. The *Circuit Service* generates *Quantum Circuits* for the *Gate-based Quantum Computer* that contain the data and processing instructions. The transmission of *Quantum Circuits* to *Gate-based Quantum Computer* is the responsibility of the *Backend Service*. The *Result Manager* determines whether previously transmitted *Quantum Circuits* have been executed and notifies the *Application* and / or *Data Service* if a result is available. Based on the services and the *Gate-based Quantum Computer*, we define the following reference procedure in which messages are exchanged asynchronously between the services.

- **Message 1.** An *Application* notifies the *Decision Service* of the problem to be solved. This first verifies whether a problem should be calculated using a *Gate-based Quantum Computer* or not and notifies the *Application* of the decision. We assume in the following that a *Gate-based Quantum Computer* is to be used. Otherwise, the problem is solved classically and the *Application* communicates directly with the *Data Service* for the purpose of data exchange (**Message 9**).
- Message 2. The *Data Service* is notified in order to extract the data required to solve the problem.
- **Message 3.** After all data is available, the *Circuit Service* is notified to generate a corresponding *Quantum Circuit*. If appropriate, several circuits are created.
- Message 4. Once the circuits have been successfully created, the *Backend Service* is triggered to transfer them to a *Gate-based Quantum Computer*.

- **Messages 5-6.** The *Backend Service* then transmits the circuits to a *Gate-based Quantum Computer* using a quantum cloud service and initiates their execution (**Message 5**). At the same time, the *Result Manager* is notified, which monitors the status of the execution (**Message 6**).
- **Messages 7-8.** Once the *Result Manager* determines that the execution of a circuit has been completed, it retrieves the result. The status and result are retrieved via a quantum cloud service (**Message 7**). At the same time, the *Application* and / or *Data Service* can be notified (**Message 8**). The result can be sent directly to the *Application*, stored in a *Database* or both. Usually, the result obtained must be post-processed. In particular, other data records stored in *Databases* can be used for this purpose. This step is performed by the *Application*.

Next, we take a brief look at the structure of a *Quantum Circuit*. A *Quantum Circuit* can be roughly divided into the areas of *Encoding*, *Unitary Transformation* and *Measurement* [21, 23]. The *Encoding* 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 quantum algorithm in the *Unitary Transformation* 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 address two aspects that are to be supported by the HDMA.

Circuit generation. In some cases, the circuits are generated on demand, in others automatically, e.g. when data changes. The first method can be useful if, for example, different small portions of historical data or predefined problem instances are to be processed. In the second case, data changes (e.g. new or updated data records) must be detected automatically and the circuit generation must then be triggered automatically at certain times. In this scenario, a quantum algorithm works with data set at a given time or the most recent time.

Data restrictions / data economy. Considering data restrictions in the architecture is central 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 given use case, different profiles can be used, e.g. to manage different data value ranges or different graph partitions. In the latter case, different circuits could be generated for the different node sets (each partition forms the input for a calculation task). Some use cases (which especially process historical data) allow different circuits to be executed independently of each other. In the NISQ era, this is of particular interest. Due to the hardware limitations, the width and depth of quantum circuits should be reduced as much as possible [3]. In general, validation of use cases respectively the aforementioned computationally intensive queries is only possible with small problem instances in the NISQ era.

4 Experimentation

In this Section, we describe the experiment to test the hybrid architecture. For the development of the first prototype, we use *Python*, *FastAPI*, *Docker* and *Qiskit* [18] as implementation technologies. The objective is to verify the functioning of the reference procedure described in Section 3. To validate the procedure, we resort to well-known quantum routines. We first consider the following Table 1.

Table 1: Test Data						
	ID	Feature1	Feature2	Cluster		
Centroid A	0	-0.5	0.5	blue		
Centroid B	1	0.2	-0.2	green		
Data Point	2	0.15	-0.15	?		
Data Point	3	-0.45	0.45	?		

This represents some relational data. Each row (tuple) represents a data point that has two properties (called features) *Feature1* and *Feature2*. Each tuple also has a unique ID. The first two tuples are data points, each representing a cluster centroid. The remaining data points are now to be assigned to a cluster. We aim to achieve this with the support of a routine or algorithm for distance estimation as used in the Quantum K-Means algorithm [16, 3].

According to the reference procedure in Section 3, we implement the following services and functionalities in the prototype:

- 1. The *Decision Service* receives a request from the *Application* for the assignment of data points to clusters and decides to solve this problem using a *Gate-based Quantum Computer*. The *Application* is informed of this.
- 2. Next, a defined amount of data records is extracted via the Data Service.
- 3. The *Circuit Service* creates several circuits. We create a circuit for each pair of data point and centroid. In this first experiment, we have opted for this simple approach. First, the data provided is encoded. In our case, these are the features and the ID of the data points. We need the latter for post-processing (cf. step 5). For the encoding of the features we use the so-called *angle embedding* [3, 16]. For this we calculate per tuple the two angles $\theta = (Feature1 + 1)\frac{\pi}{2}$ and $\varphi = (Feature2 + 1)\frac{\pi}{2}$, whereby angle values between 0 and π are allowed. For the encoding of the IDs we use the basis encoding in this example [26]. Next the distance estimation algorithm follows, which works on the encoded information (cf. Subsection 4.1). Note: For the encoding of the IDs, we have chosen basis encoding in this first example. Other implicit encoding types (as in [15]) are also of interest. In addition, the identification of records in a distributed database system requires more comprehensive identifiers than just an ID as in this example. This was reported in [29].
- 4. The generated circuits are transmitted to a *Gate-based Quantum Computer* via the *Backend Service* and their execution is triggered. In our case, it is the quantum cloud service from IBM Quantum¹, which we used for our tests.
- 5. In the final step, the *Result Manager* checks the execution status and retrieves the result after a circuit has been executed. The result is then forwarded to the *Application*. The corresponding cluster of each data point can be derived from the overall result (comparison of all points with both centroids, cf. Subsection 4.2). The IDs of these data points are included in the individual results. The *Application* is responsible for post-processing the overall result. Based on the IDs, the Table 1 can be supplemented with the information of the correct cluster. In a relational database, we can imagine that this table is indexed by ID, so that records can be efficiently identified by ID for an update.

4.1 Circuit

The circuit created in Step 3) can be roughly divided into two sections *Encoding* and *Algorithm*. Figure 2 shows the circuit as an example for a data point centroid pair. The encoding is marked. To encode the features, so called $U_3(\theta, \varphi, 0)$ gates can be used. The parameters of the gates are the previously calculated angles. *Pauli X* gates can be used for encoding the IDs. In this case, an ID is represented by a corresponding bit string.

The core of the actual algorithm is the estimation of the distance between a data point and a centroid. The basis of the algorithm is described in [16]. We supplement it with the encoding of the IDs. The distance is reflected in the measurement result and how often a particular measurement result occurs. We will deal with this in Subsection 4.2.



Figure 2: Distance Estimation Algorithm following [16] with additional encoding of the Tuple IDs to utilize them in post-processing. Parameterized with Data Point ID=3 and Centroid A. Drawn by *Qiskit*.

¹IBM Quantum. https://quantum.ibm.com/, 2023

4.2 Results

In this Subsection we review the results, which are shown in the Table 2.

Table 2: Experiment Results						
	Data Point	Bit sequence	Determined	Calculated		
	ID	for $\mathbf{c}[2] = 1$	Frequency	Frequency		
Centroid A	2	1 10	43	≈ 47		
Centroid B	2	-	0	pprox 0.513		
Centroid A	3	-	0	pprox 0.061		
Centroid B	3	1 11	43	≈ 46		

Each data point (**Column 1**) is compared to the two centroids to determine the proximity in each case. In quantumbased distance estimation, the estimated distance between points can be derived from the frequency of a particular state that we obtain from the measurements. For this purpose, we use the *ibmq_qasm_simulator* with 1000 shots (number of repetitions) per centroid/data point comparison. According to the Figure 2, we measure $q_0 \rightarrow c[2], q_3 \rightarrow c[1]$, and $q_4 \rightarrow c[0]$ and obtain in this way a bit sequence representing the measured state (**Column 2**). c[1]c[0] corresponds to the binary representation of the ID of the data point. We are interested in the case c[2] = 1 (the particular state) and the frequency of its occurrence (**Column 3**). In this case, a low frequency correlates with a low estimated distance between a point and a centroid (if the value for the determined frequency is zero, then it means that the probability of measuring the particular state is low). **Column 4** contains the calculated frequencies (for 1000 repetitions). We need these to compare the determined values (in Column 3) with predicted values and thus verify the correctness of the architectural workflow. The calculated probability for c[2] = 1 is: $\frac{1}{2} - \frac{1}{2} |\langle q_1 | q_2 \rangle|^2$ [16, 3]. The calculated frequencies correspond approximately to the determined values and prove the correct functioning of the architecture implementation.

Based on the estimated distances, the source table can be supplemented. The supplemented entries are shown in Table 3.

Table 3: Updated Test Data						
	ID	Feature1	Feature2	Cluster		
Data Point	2	0.15	-0.15	green		
Data Point	3	-0.45	0.45	blue		

The architecture thus allows data from a database to be processed by a quantum algorithm to assign data points to the correct clusters. The correct generation of the corresponding circuits was ensured by the previously mentioned comparison of determined and calculated frequencies.

5 Summary and Future Work

In this paper, we have presented a Hybrid Data Management Architecture to enable data exchange between databases and quantum computers. We have experimentally proven a correct functioning of the architecture. In the experiments, quantum circuits were created and executed based on relational data. The circuits implement an algorithm for the distance estimation between data point and centroid and includes the encoding of the classical data. However, the architecture is not limited to this use case.

Next, we would like to leverage the architecture to implement more data-centric applications. In doing so, we would like to encode data organized in hierarchical structures (for example, labelled trees for which we have outlined an initial idea [28], or graphs) in order to execute selected queries on the encoded data using appropriate quantum algorithms. The implemented routines are also accompanied by time and space complexity analysis to deduce any advantages over purely classical solutions.

Acknowledgement

This work has been funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) grant #385808805. We would like to thank Stefanie Scherzinger from University of Passau for many prolific discussions as well as helpful suggestions.

References

- Umut Çalikyilmaz, Sven Groppe, Jinghua Groppe, Tobias Winker, Stefan Prestel, Farida Shagieva, Daanish Arya, Florian Preis, and Le Gruenwald. Opportunities for Quantum Acceleration of Databases: Optimization of Queries and Transaction Schedules. *Proc. VLDB Endow.*, 16(9):2344–2353, 2023.
- [2] Claire David. Complexity of Data Tree Patterns over XML Documents. In *Proc. MFCS'08*, volume 5162, pages 278–289. Springer, 2008.
- [3] Stephen DiAdamo, Corey O'Meara, Giorgio Cortiana, and Juan Bernabé-Moreno. Practical Quantum K-Means Clustering: Performance Analysis and Applications in Energy Grid Classification. *IEEE Transactions on Quantum Engineering*, 3:1–16, 2022.
- [4] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch-Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. Microservices: Yesterday, Today, and Tomorrow. In *Present and Ulterior Software Engineering*, pages 195–216. Springer, 2017.
- [5] Georg Gottlob, Christoph Koch, and Reinhard Pichler. The complexity of XPath query evaluation. In *Proc. PODS'03'*, pages 179–190. ACM, 2003.
- [6] Vikas Hassija, Vinay Chamola, Adit Goyal, Salil S. Kanhere, and Nadra Guizani. Forthcoming applications of quantum computing: peeking into the future. *IET Quantum Communication*, 1(2):35–41, 2020.
- [7] Steven Herbert. Quantum computing for data-centric engineering and science. *Data-Centric Engineering*, 3, 2022.
- [8] Essam H. Houssein, Zainab Abohashima, Mohamed Elhoseny, and Waleed M. Mohamed. Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision. *Expert Syst. Appl.*, 194:116512, 2022.
- [9] Szabolcs Jóczik and Attila Kiss. Quantum Computation and Its Effects in Database Systems. In *New Trends in Databases and Information Systems*, volume 1259, pages 13–23. Springer, 2020.
- [10] Mária Kieferová and Yuval Sanders. Assume a Quantum Data Set. Harvard Data Science Review, 4(1), 2022.
- [11] Tim Kraska, Tianyu Li, Samuel Madden, Markos Markakis, Amadou Ngom, Ziniu Wu, and Geoffrey X. Yu. Check Out the Big Brain on BRAD: Simplifying Cloud Data Processing with Learned Automated Data Meshes. *Proc. VLDB Endow.*, 16(11):3293–3301, 2023. ISSN 2150-8097.
- [12] Frank Leymann and Johanna Barzen. The bitter truth about gate-based quantum algorithms in the NISQ era. *Quantum Science and Technology*, 5(4):044007, 2020.
- [13] Junyu Liu, Connor T. Hann, and Liang Jiang. Data centers with quantum random access memory and quantum networks. *Phys. Rev. A*, 108:032610, Sep 2023.
- [14] Ioana Manolescu and Madhulika Mohanty. Full-Power Graph Querying: State of the Art and Challenges. Proc. VLDB Endow., 16(12):3886–3889, 2023. ISSN 2150-8097.
- [15] Olivia Di Matteo, Vlad Gheorghiu, and Michele Mosca. Fault-Tolerant Resource Estimation of Quantum Random-Access Memories. *IEEE Transactions on Quantum Engineering*, 1:1–13, 2020.
- [16] Oumayma Ouedrhiri, Oumayma Banouar, Said Raghay, and Salah el Hadaj. Comparative study of data preparation methods in quantum clustering algorithms. In NISS (ACM), pages 28:1–28:5. ACM, 2021.
- [17] Koustubh Phalak, Avimita Chatterjee, and Swaroop Ghosh. Quantum Random Access Memory For Dummies. *CoRR*, abs/2305.01178, 2023.
- [18] Qiskit contributors. Qiskit: An Open-source Framework for Quantum Computing, 2023.
- [19] Heike Riel. Quantum computing technology. In 2021 IEEE International Electron Devices Meeting (IEDM), 2021.
- [20] Maria Schuld and Francesco Petruccione. Quantum Computing, pages 79–146. Springer International Publishing, 2021. ISBN 978-3-030-83098-4.
- [21] Maria Schuld and Francesco Petruccione. Representing data on a quantum computer. In *Machine Learning with Quantum Computers*, pages 147–176. Springer, 2021. ISBN 978-3-030-83098-4.

- [22] Benjamin Weder, Johanna Barzen, Frank Leymann, and Michael Zimmermann. Hybrid Quantum Applications Need Two Orchestrations in Superposition: A Software Architecture Perspective. In 2021 IEEE International Conference on Web Services (ICWS), pages 1–13, 2021.
- [23] Manuela Weigold, Johanna Barzen, Frank Leymann, and Marie Salm. Encoding patterns for quantum algorithms. *IET Quantum Communication*, 2(4):141–152, 2021.
- [24] Manuela Weigold, Johanna Barzen, Frank Leymann, and Daniel Vietz. Patterns for Hybrid Quantum Algorithms. In SummerSOC, volume 1429 of Communications in Computer and Information Science, pages 34–51. Springer, 2021.
- [25] Manuela Weigold, Johanna Barzen, Frank Leymann, and Marie Salm. Expanding Data Encoding Patterns For Quantum Algorithms. In 2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C), pages 95–101. IEEE, 2021-03.
- [26] Manuela Weigold, Johanna Barzen, Frank Leymann, and Marie Salm. Data encoding patterns for quantum computing. In *Proceedings of the 27th Conference on Pattern Languages of Programs*, PLoP '20. The Hillside Group, 2022.
- [27] Gongsheng Yuan, Jiaheng Lu, Yuxing Chen, Sai Wu, Chang Yao, Zhengtong Yan, Tuodu Li, and Gang Chen. Quantum Computing for Databases: A Short Survey and Vision. In VLDB Workshops, volume 3462 of CEUR Workshop Proceedings. CEUR-WS.org, 2023.
- [28] Markus Zajac. Encoding and Provisioning Data in different Data Models for Quantum Computing. In *PhD@VLDB*, volume 3452 of *CEUR Workshop Proceedings*, pages 45–48. CEUR-WS.org, 2023.
- [29] Markus Zajac and Uta Störl. Towards quantum-based Search for industrial Data-driven Services. In *Proceedings* of the 2022 IEEE International Conference on Quantum Software (QSW). IEEE, 2022.