Applicability of Quantum Computing on Database Query Optimization

Manuel Schönberger
Technical University of Applied Sciences Regensburg
Regensburg, Germany
manuel.schoenberger@oth-regensburg.de

ABSTRACT

We evaluate the applicability of quantum computing on two fundamental query optimization problems, join order optimization and multi query optimization (MQO). We analyze the problem dimensions that can be solved on current gate-based quantum systems and quantum annealers, the two currently commercially available architectures.

First, we evaluate the use of gate-based systems on MQO, previously solved with quantum annealing. We show that, contrary to classical computing, a different architecture requires involved adaptations. We moreover propose a multi-step reformulation for join ordering problems to make them solvable on current quantum systems. Finally, we systematically evaluate our contributions for gate-based quantum systems and quantum annealers. Doing so, we identify the scope of current limitations, as well as the future potential of quantum computing technologies for database systems.

CCS CONCEPTS

• **Computer systems organization → Quantum computing**; • **Theory of computation → Database query processing and optimization (theory)**.

KEYWORDS

quantum computing, quantum computation, query optimization, multiple query optimization, join order optimization

ACM Reference Format:

Manuel Schönberger. 2022. Applicability of Quantum Computing on Database Query Optimization. In *Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22), June 12–17, 2022, Philadelphia, PA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3514221.3520257

1 INTRODUCTION

A fundamental problem in query optimization research is determining the optimal join order [12]. Multi query optimization (MQO) seeks to determine a globally optimal set of execution plans for a set of queries such that execution costs are minimized through sharing and reusing the results of common subexpressions [8, 18]. Approaches for both problems have been investigated for decades, including genetic algorithms [1, 19] and mixed integer linear programming (MILP) [22]. An approach based on quantum computing has been proposed for MQO [5, 21]. For some small instances, it can find optimal solutions faster on a quantum annealer than classical solvers.

Quantum processing units (QPU) work with quantum bits, or qubits [13]. Qubits, making use of the quantum superposition phenomenon, are not limited to be in either one of the states 0 or 1 at a given moment in time. This allows to encode significantly more information than with classical bits, providing QPUs with a computational advantage over classical CPUs. However, quantum computing, in particular its application to database problems, is still a largely unexplored topic. To the best of our knowledge, MQO is the only database problem investigated so far. At the same time, quantum computers, having matured from the state of prototypes in laboratories, are now available to non-expert users, and can even be booked as cloud services (see for instance, IBM's QPUs [7]). Due to the increased accessibility, we envision their integration into the DBMS architecture, as depicted in Figure 1.

Unfortunately, existing classical algorithms can typically not directly be deployed on QPUs: While the actual implementation effort is limited [17], efficient problem reformulations that account for QPU properties need to be found, which is challenging. Qubits are still a very scarce resource: At the time of writing, the latest D-Wave quantum annealer features over 5,000 qubits [11] whereas IBM-Q offers gate-based QPUs with up to 65 qubits [7]. In addition, QPUs only provide limited connectivity between qubits. To increase qubit connectivity, quantum annealers use chains of physical qubits to represent logical qubits [2]. Problems with high connectivity requirements generate longer chains, leaving only a fraction of the available physical qubits usable as logical qubits. This reduces the qubit advantage of quantum annealers over gate-based QPUs. It is further an open debate whether quantum annealing can truly provide speedups unreachable with classical systems [15].

High connectivity requirements also impact gate-based QPUs, which execute quantum circuits comparable to classical ones [13]. Specifically, they increase the likeliness of decoherence errors caused by a loss of quantum information to the environment [16]. Since qubit connectivity is increased by inserting additional gates into the circuit during embedding [3, 20], the execution time of the extended circuit may exceed the limited coherence time of the QPU.

The prospects of using quantum computing to achieve speedups are nevertheless promising. The goal of this work is to investigate the applicability of quantum computing to database query optimization w.r.t. the problem dimensions solvable on current QPUs. Our original contributions are: (a) we investigate the applicability of

SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA

ACM ISBN 978-1-4503-9249-5/22/06.
https://doi.org/10.1145/3514221.3520257

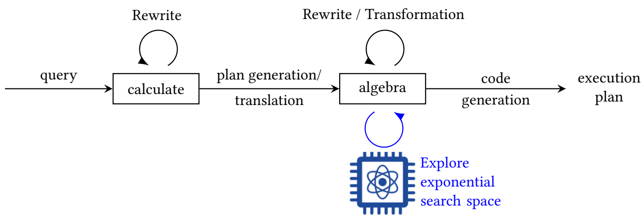

Figure 1: Envisioning a QPU as a co-processor in database query optimization (adapting from [12]).

current gate-based QPUs on MQO; (b) we reformulate join ordering problems for current QPUs; (c) we compare the approach for gate-based QPUs and quantum annealers.

2 APPROACH

Quantum annealers solve quadratic unconstrained binary optimization (QUBO) problems [11], which can also be solved running hybrid quantum-classical algorithms on gate-based QPUs [10]. Hybrid algorithms run only partially on QPUs, and are augmented by CPU computations. We consider two algorithms: the variational quantum eigensolver (VQE) [14] and the quantum approximate optimization algorithm (QAOA) [6]. We study their applicability on a variety of MQO problems, reformulated following Ref. [21], with up to 24 plans. While QAOA was previously investigated for MQO in Ref. [5], the scaling behavior for transpiled circuits has so far not been analyzed. We transpile the quantum circuits for the qubit topology of the IBM-Q Mumbai QPU with 27 qubits [7] and analyze the transpiled circuits w.r.t. depth. This, and the number of required qubits, are crucial parameters to judge technical feasibility of the approach for years (likely decades) to come.

A lower bound for the number of required qubits is given by the total number of alternative plans [21]. The number of quadratic contributions to the QUBO formulation is also crucial, since a large number of quadratic terms requires a high qubit connectivity. For MQO, this is mainly influenced by the number of alternative plans per query [21]. We therefore study different classes of MQO problems with varying numbers of plans per query.

For join ordering, we propose a novel multi-step reformulation that enables QPU usage. We reformulate the problems as MILP problems [4], as done in Ref. [22], which considers left-deep join trees and supports Cartesian products. We consider the basic MILP formulation that minimizes the cardinalities of intermediate join results, but do not account for any of the extensions to the MILP model proposed in Ref. [22], since these require significantly more qubits. We then eliminate inequality constraints and we discretize resulting continuous variables based on arbitrary precision. This allows us to cast the problems as binary integer linear programming (BILP) formulations, for which an efficient transformation into QUBO form is known [9]. We analyze our approach for a current gate-based QPU and for the qubit topology of the D-Wave Advantage system.

The MILP formulation allows to specify an arbitrary number of threshold values for approximating the cardinalities of intermediate join results. Each value requires a number of corresponding variables. Since each BILP variable is encoded by one qubit, increasing

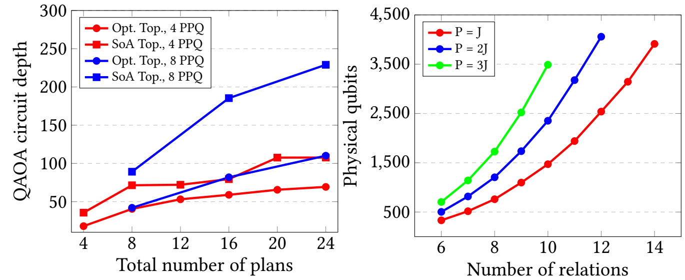

Figure 2: Resource scaling for MQO (left) and join ordering (right) on state-of-the-art (SoA) QPUs.

the approximation precision therefore increases the number of required qubits. We derive an upper bound (not derived here) for the number of required qubits n: For T relations, J joins, P predicates, and R threshold values,

$$n \leq 2TJ + (3P + R)(J - 1) + T + R \sum_{j=1}^{J-1} \left(\left\lfloor \log_2 \left(\frac{c_j}{\omega} \right) \right\rfloor + 1 \right)$$

qubits are required at most. Hereby, ω denotes the discretization precision, and c_j gives the maximum logarithmic cardinality possible for the intermediate result serving as an operand for the j-th join.

3 FIRST RESULTS

Overall, due to qubit limitations, only significantly smaller MQO problems are solvable on gate-based QPUs compared to prior results for quantum annealing shown in Ref. [21]. Further, Figure 2 compares QAOA quantum circuit depths for an ideal qubit topology against a physically realizable IBM-Q QPU. Due to the depth increase after transpilation, some circuits (e.g., for 24 plans and 8 plans per query) already come close to exceeding the coherence time, which further limits the scalability of the approach in addition to the qubit numbers.

Figure 2 also shows quantum annealing results for join ordering problems: It depicts the physical qubits required for all problems where an embedding can be reliably found (i.e., in at least 50% of the cases), the embedding algorithm times out otherwise. We found embeddings for problems with up to 14 relations. In comparison, only small-scale join ordering instances can be solved on IBM-Q.

Integrating QPUs into existing architectures requires a great adaption effort, including the design of efficient problem reformulations. Here, we proposed such a reformulation for the join ordering problem. Current QPU limitations prevent us from scaling up problem dimensions. However, with steadily maturing QPUs, these limitations may soon become less restrictive. As such, it is now the right time to investigate the potential of quantum computing on database problems to enable the use of future QPUs on practical problems.

Acknowledgements. I thank my supervisors Wolfgang Mauerer (Technical University of Applied Sciences Regensburg) and Stefanie Scherzinger (University of Passau) for guidance and support throughout this research. This research was supported by the German Federal Ministry of Education and Research (BMBF), funding program "quantum technologies—from basic research to market", grant number 13N15645.

REFERENCES

[1] Murat Ali Bayir, Ismail H. Toroslu, and Ahmet Cosar. 2006. Genetic algorithm for the multiple-query optimization problem. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37, 1 (Dec. 2006), 147–153.

[2] P. I. Bunyk, Emile M. Hoskinson, Mark W. Johnson, Elena Tolkacheva, Fabio Altomare, Andrew J. Berkley, Richard Harris, Jeremy P. Hilton, Trevor Lanting, Anthony J. Przybysz, and Jed Whittaker. 2014. Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Transactions on Applied Superconductivity* 24, 4 (April 2014), 1–10.

[3] Lukas Burgholzer, Rudy Raymond, and Robert Wille. 2020. Verifying results of the IBM Qiskit quantum circuit compilation flow. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, Denver, CO, USA, 356–365.

[4] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. 2014. *Integer programming*. Graduate Texts in Mathematics, Vol. 271. Springer International Publishing, Cham.

[5] Tobias Fankhauser, Marc E. Solèr, Rudolf M. Füchslin, and Kurt Stockinger. 2021. Multiple query optimization using a hybrid approach of classical and quantum computing. (July 2021). arXiv:2107.10508

[6] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. (Nov. 2014). arXiv:1411.4028

[7] IBM Quantum. 2021. Cloud access to quantum computers provided by IBM. https://quantum-computing.ibm.com

[8] Tarun Kathuria and S. Sudarshan. 2017. Efficient and provable multi-query optimization. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. ACM, New York, NY, USA, 53–67.

[9] Andrew Lucas. 2014. Ising formulations of many NP problems. *Frontiers in Physics* 2 (Feb. 2014), 5.

[10] Jarrod R. McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. 2016. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics* 18, 2 (Feb. 2016), 023023.

[11] Catherine McGeoch and Pau Farré. 2020. *The D-Wave Advantage system: An overview.* Technical Report 14-1049A-A. D-Wave Systems Inc.

[12] Guido Moerkotte. 2020. Building query compilers. https://pi3.informatik.uni-mannheim.de/~moer/querycompiler.pdf

[13] Michael A. Nielsen, Isaac Chuang, and Lov K. Grover. 2002. Quantum computation and quantum information. *American Journal of Physics* 70, 5 (April 2002), 558–559.

[14] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications* 5, 1 (July 2014), 4213.

[15] John Preskill. 2018. Quantum computing in the NISQ era and beyond. *Quantum* 2 (Aug. 2018), 79.

[16] Eleanor Rieffel and Wolfgang Polak. 2011. *Quantum computing: A gentle introduction.* MIT Press, Cambridge, MA.

[17] Manuel Schönberger, Maja Franz, Stefanie Scherzinger, and Wolfgang Mauerer. 2022. Peel | pile? Cross-framework portability of quantum software. In *19th IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE, Honolulu, HI, USA.

[18] Timos K. Sellis. 1988. Multiple-query optimization. *ACM Transactions on Database Systems* 13, 1 (March 1988), 23–52.

[19] Michael Steinbrunn, Guido Moerkotte, and Alfons Kemper. 1997. Heuristic and randomized optimization for the join ordering problem. *The VLDB Journal* 6, 3 (Aug. 1997), 191–208.

[20] Swamit S. Tannu and Moinuddin K. Qureshi. 2019. Not all qubits are created equal: A case for variability-aware policies for NISQ-era quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, New York, NY, USA, 987–999.

[21] Immanuel Trummer and Christoph Koch. 2016. Multiple query optimization on the D-Wave 2X adiabatic quantum computer. *Proceedings of the VLDB Endowment* 9, 9 (May 2016), 648–659.

[22] Immanuel Trummer and Christoph Koch. 2017. Solving the join ordering problem via mixed integer linear programming. In *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, New York, NY, USA, 1025–1040.