M. Klein, D. Krupka, C. Winter., M. Gergeleit, L. Martin (Hrsg.): INFORMATIK 2024,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2024 573

Quantum versus Classical Computation: Automatic Decision-Making Approaches

Markus Zajac [1], Valerie Restat [1], and Uta Störl [1]

Abstract: The limitations of today's quantum hardware require the development of hybrid applications in which parts of the calculations are to be carried out on quantum computers. The decision of whether outsourcing particular calculations actually offers an advantage and is worthwhile is not a trivial one. Decisions are problem-specific and are made manually these days. In this work-in-progress paper, we briefly outline some decision-making approaches and describe the idea of a rule-based approach in more detail. These approaches could form the basis for the implementation of a decision service that is integrated into a hybrid software architecture.

Keywords: Hybrid Quantum Computing, Decision Making, Quantum-Classical Splitting

1 Introduction

The development of so-called hybrid applications [GGW21] and the design of quantum software architectures for hybrid applications [Yu23] are required to overcome the limitations of today's quantum computers. This means that calculations or problems are outsourced to quantum computers, for which quantum computers can achieve a significant speed-up compared to classical systems [We20]. Pre- and post-processing is still carried out classically (this procedure is referred to below as the hybrid approach). Nowadays, however, it is necessary to manually decide which calculations or problems should actually be calculated on quantum computers [We20; Yu23].

Before calculations are actually outsourced to quantum computers, it should first be examined whether the hybrid approach offers any advantage over purely classical systems. We intend to realize this examination through a decision service. This is embedded in a hybrid software architecture [ZS23] and should decide as autonomously as possible on request, whether the hybrid approach (advantage over the purely classical approach exists) or the purely classical approach should be used (no advantage with the hybrid approach) to perform calculations. Although the service exists as a component of the architecture, its decision logic must still be implemented.

Our contribution: We consider three approaches to decision making as a basis for the decision service and describe our idea of the rule-based approach.

[1] FernUniversität in Hagen, Databases and Information Systems, 58097 Hagen, Germany,
markus.zajac@studium.fernuni-hagen.de, https://orcid.org/0000-0002-9338-9259;
valerie.restat@fernuni-hagen.de, https://orcid.org/0000-0002-5960-5886;
uta.stoerl@fernuni-hagen.de, https://orcid.org/0000-0003-2771-142X

doi:10.18420/inf2024_45

The remainder of the paper is structured as follows: In Section 2, we briefly present three approaches to decision-making. We describe the rule-based approach in Section 3. Section 4 contains a summary and future work.

2 Approaches to Decision Making

In this Section, we briefly present three possible approaches to decision-making. As a prerequisite, we assume a dataset with at least these features, which we use in the following approaches: *problemDesignation*, $t_a, t_{pc}, t_q, t_s, t_{ph}$. The feature *problemDesignation* names a problem or a variant of a problem or a routine or a calculation in general for which it is to be checked whether classical or quantum hardware should be used. The other features specify the time complexities according to Table 1.

Function term	Type of algorithm
$t_a(n) = O(a(n))$	Time complexity of the classical algorithm t_a.
$t_{pc}(n) = O(pc(n))$	Time complexity of pre- and post-processing routines t_{pc}.
$t_c(n) = O(a(n) + pc(n))$	Total time complexity t_c in the classical approach to solve p.
$t_q(n) = O(q(n))$	Time complexity of the quantum algorithm t_q.
$t_s(n) = O(s(n))$	Time complexity for the state preparation routine t_s of a quantum algorithm to encode data and information.
$t_{ph}(n) = O(ph(n))$	Time complexity of pre- and post-processing routines t_{ph}.
$t_h(n) = O(q(n) + s(n) + ph(n))$	Total time complexity t_h in the hybrid approach to solve p.

Tab. 1: The table defines function terms according to the Big-O notation. $t(n) = O(g(n))$ (or $t(n) \in O(g(n))$) defines the worst-case time complexity of an algorithm type for an input size n. The first half of the table contains entries relating to the purely classical approach; the second half relates to the hybrid approach. Both approaches solve the same problem p.

Rule-based approach. Rule-based systems generally use a set of predefined rules to make decisions and thus solve problems or issues. The underlying knowledge can be organized in terms of if-then statements [RF22]. This approach can be used for the implementation of the decision service, for example by comparing the time complexity of a classical and a hybrid approach in the case of a service request. If the hybrid approach offers an advantage, then it is recommended, otherwise the classical approach is recommended. The main advantages are the rapid implementation of such a system, the unambiguousness of the rules and the compatibility with machine learning approaches. The disadvantages of this approach, however, are the administration of many exception rules, time-consuming subsequent adjustments of many rules and the lack of learning ability of such a system (compared to machine learning approaches).

Supervised learning. Supervised learning is a type of machine learning in which a model is trained with labeled data [CCD08]. An implementation for the decision service could look like this: Various requests serve as input. The features correspond to those mentioned above. *problemDesignation* is a categorical feature. One hot encoding would be conceivable here. The others are ordinal features[2]. Ordinal encoding would thus be an option. The actual strength of machine-learning-based approaches only comes into play when further information is added, so that the decision becomes too complex for a rule-based approach. Other conceivable features would be: *space complexity*, *quantum hardware* and *problem instance*. In addition, a label is necessary. This label would be the decision as to whether the calculations should be executed on classical hardware or on quantum computers. Thus, a lot of data is required for which the decision is already known. After training the model, a classification could be made for future requests. Depending on the classification, the calculations would be carried out on the corresponding hardware. The advantage of this approach is that decisions can be made completely automatically. No manual intervention and therefore no domain knowledge is required. However, a large amount of labeled data is required for this.

Unsupervised learning. In contrast to supervised learning, unsupervised learning does not require labels [Ty22]. Instead, it is used to search for patterns and correlations in the data. For the decision service, the data could be grouped into clusters. The input data would be the same as the supervised approach, but without the associated labels. After the training phase, in which the clusters are created, manual labeling by domain experts would be required. They would decide which clusters would correspond to execution on classical hardware and which would correspond to execution on quantum computers. After this one-time labeling, the processing for new requests can be automated again. When a new request comes in, it is assigned to one of the clusters by the algorithm. Since the labels of the clusters are already available, the calculations are executed on classical hardware or quantum computers according to label of the assigned cluster.

General process: Regardless of the approach, the process that leads to the decision is always the same. A request to the decision service is made. In a pre-processing step, this request is converted into the features presented above. These are then passed to the rules or the trained model that provides the decision.

Only the way this decision is made differs. As described, this involves varying amounts of effort: A set of rules must first be created for the rule-based approach. For supervised learning, a comprehensive training dataset is required for which decisions are already available. An extensive training dataset is also required for unsupervised learning. However, no labels are needed here. Instead, a one-time manual labeling of the clusters is necessary.

We currently do not have a sufficiently large amount of training data available. For this reason, we initially opt for the rule-based approach, which is described in more detail below.

[2] Since the following hierarchy applies: $O(1) < \cdots < O(n) < \cdots < O(n^2) < \cdots < O(n!)$.

3 Rule-based Approach

In Section 2 we have opted for the rule-based approach, which we describe in this section. The basis of this approach is the asymptotic behavior of the algorithms. This allows us to make an estimate of the scaling of the time complexity of a classical and hybrid solution, to relate these to each other, and to make a decision for or against a hybrid approach. We define time complexity as the number of iterations of an algorithm. The prerequisite for this approach is that the time complexity of all classical and hybrid algorithms involved in a solution can be determined or assumed.

According to Table 1, $t_c(n) = O(a(n) + pc(n))$ describes the overall time complexity in the classical approach and $t_h(n) = O(q(n) + s(n) + ph(n))$ in the hybrid approach. To make a decision, we check whether the following inequality applies to a problem (cf. footnote 2):

$$O(q(n) + s(n) + ph(n)) < O(a(n) + pc(n)) \tag{1}$$

If the inequality is fulfilled, then the time complexity grows faster in the classical approach and is therefore disadvantageous compared to the hybrid approach. The latter would then appear to offer an asymptotic advantage in terms of the number of operations. Efficient encoding [LB20] as well as efficient pre- and post-processing are therefore of central importance to ensure that any advantages of a quantum computer do not become insignificant. It should also be noted that, from today's perspective, classical computers are likely to remain more advantageous for small problem inputs [HHT23]. The consideration of corresponding problem instances is therefore of great interest.

To implement the rule-based approach as the basis for the decision service, we first set up a suitable database with our dataset. This assigns algorithm types and time complexities to different problems. To determine the complexities of time, the research literature is consulted, as well as other existing sources[3]. Once a database has been created, the service examines whether the inequality is fulfilled or not when a request is made and recommends a classical or hybrid approach. The decision-making process based on comparing time complexities seems simple. However, additional features such as *space complexity*, *quantum hardware* and *problem instance* increase the complexity of the decision. This pushes the boundaries of the rule-based approach and argues in favor of an ML-based approach.

4 Summary and Future Work

In this paper, we have outlined three approaches to decision making. They are intended to assist in automatic decision-making processes as to whether certain problems or calculations should be calculated purely classically or hybridly.

We conceptually described a rule-based approach in order to create an implementation basis for the decision service. The approach compares the time complexity of a hybrid and a

[3] https://quantumalgorithmzoo.org/

classical solution to a problem. The next step is to implement and evaluate this approach. In a further step, the *space complexity* should also be taken into account in the decision-making process.

We also intend to implement and evaluate the described machine learning approaches in the future. For this, a suitable data base must first be created and training carried out.

References

[CCD08] Cunningham, P.; Cord, M.; Delany, S. J.: Supervised Learning. In: Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval. Pp. 21–49, 2008.

[GGW21] Gemeinhardt, F.; Garmendia, A.; Wimmer, M.: Towards Model-Driven Quantum Software Engineering. In: Q-SE@ICSE. Pp. 13–15, 2021.

[HHT23] Hoefler, T.; Häner, T.; Troyer, M.: Disentangling Hype from Practicality: On Realistically Achieving Quantum Advantage. Commun. ACM 66 (5), pp. 82–87, 2023.

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

[RF22] Reddy, B.; Fields, R.: From past to present: a comprehensive technical review of rule-based expert systems from 1980 - 2021. In: ACM Southeast Regional Conference. Pp. 167–172, 2022.

[Ty22] Tyagi, K. et al.: Chapter 3 - Unsupervised learning. In: Artificial Intelligence and Machine Learning for EDGE Computing. Pp. 33–52, 2022.

[We20] Weder, B. et al.: The Quantum software lifecycle. In: APEQES@ESEC/SIGSOFT FSE. Pp. 2–9, 2020.

[Yu23] Yue, T. et al.: Challenges and Opportunities in Quantum Software Architecture. In: Software Architecture. Pp. 1–23, 2023.

[ZS23] Zajac, M.; Störl, U.: Hybrid Data Management Architecture for Present Quantum Computing. In: ICSOC Workshops. Vol. 14518. Lecture Notes in Computer Science, pp. 174–184, 2023.