JURNAL ILMU KOMPUTER
Volume 2 Nomor 2, Desember, 2024
ISSN: 3031-125X (ONLINE)
Penerbit: Program Studi Teknik Informatika S-2 Universitas Pamulang

Technical Comparison Between Classical and Quantum Architectures: Quantum Error Challenges and Qubit Stability

*Bonie Wijaya[1], Muhammad Fahrizal[2] ,Muhrodi[3], Dhamma Nagara[4], Yan Everhard[5]

1,2,3,4,5 Master's Degree, Faculty of Information Technology, Budi Luhur University, Jakarta

Email: [1]2411600014@student.budiluhur.ac.id, [2]2411600030@student.budiluhur.ac.id, [3]2411600048@student.budiluhur.ac.id, [4]2411600097@student.budiluhur.ac.id, [5]yan.everhard@budiluhur.ac.id

ABSTRACT

In ~~T~~the age of evolving computational technologies, classical architecture (traditional digital computing) and quantum architecture have emerged as two prominent approaches, offering diverse computational solutions. Classical computing bases its operations on transistors and binary logic gates, while quantum computing leverages the principles of quantum mechanics to perform information processing. This article provides a technical comparison between the two architectures, encompassing essential characteristics, algorithms, processing models, problem-solving capabilities, and challenges faced.-This article highlights the key challenges in quantum computing, namely quantum errors and qubit stability, which significantly impact its reliability and practical implementation. The method used in this research is a literature review study, analyzing various reference sources such as journals, articles, and research reports. With the growing influence of quantum computing in specific sectors, this study is expected to provide a clearer view of the potential and limitations of both architectures, as well as the steps needed to overcome these challenges. The main conclusion of this study is that quantum computing has the potential to revolutionize certain fields but still faces challenges in terms of stability and error correction.

Keywords: classical computing; quantum computing; processor architecture; quantum mechanics; quantum algorithms; future technologies.

1. INTRODUCTION

Computation has been the foundation of the modern digital revolution, starting with the development of classical computers in the mid-20th century. Classical computers, or digital or binary computers, rely on transistors as their essential components and binary logic gates to process data as 0s and 1s. John von Neumann introduced the architecture underlying this computing model in 1945, now known as the von Neumann architecture. Modern classical machines have significantly evolved, especially with the introduction of supercomputers and distributed computing networks capable of processing billions of instructions per second [13], [14], [31].

However, despite the rapid development of classical computing technology, there are technical limitations. As transistors get smaller, quantum effects start to play a role, causing instability in the hardware. This problem, combined with the ever-increasing computational demands of extensive data analysis and complex modeling, raises the need for a new paradigm.

Quantum computing, first proposed by physicists such as Richard Feynman and David Deutsch in the 1980s, is one promising solution. Quantum computing exploits unique phenomena of quantum mechanics, such as superposition and entanglement, to process information in the form of qubits (quantum bits). Unlike classical bits, which can only be in a 0 or 1 state, qubits can be in any combination of both states simultaneously, enabling parallel computing on a scale that is impossible with classical architectures [27][28].

Despite its enormous potential, quantum computing ~~also~~ faces significant technical challenges, particularly related to quantum errors and qubit stability. Quantum systems are highly susceptible to environmental disturbances, which can lead to errors in computation. These challenges are a significant obstacle to realizing practical and reliable quantum computers for large-scale use.

This article aims to review the technical comparison between classical and quantum architectures. In addition, we will explore the main challenges in quantum computing, especially those related to quantum errors and qubit stability, and how these challenges may affect the reliability and application of quantum systems in the future.

A. CLASSICAL ARCHITECTURE

1) Basic Principles

Classical computing works based on Boolean logic, where information is represented in binary form (0 and 1). The computation process is carried out through a series of logic gates that operate bits (basic data units) using transistors as the main building blocks. The speed and efficiency of classical computers are determined by the number of transistors, clock speed, and architectural models such as von Neumann or Harvard architecture [2] [5].

2) Main Components

a. **CPU (Central Processing Unit)**: The core of classical computing that executes program instructions [2].

b. **Memory and Storage**: Information is stored in binary form in primary memory and secondary storage [2] [3].

c. **System Bus**: The connection between the CPU, memory, and input/output devices [2] [3].

3) Performance and Efficiency

a. Moore's law limits computing speed and there is an increasing number of transistors per chip, as well as physical limitations in heat and power management [4] [5].

b. Implementation of techniques such as **parallelism** and **multithreading** to increase efficiency [4] [5].

c. Capable of handling deterministic problems efficiently, but less effective for non-deterministic problems that require exponential time and resources [4] [5].

4) Application in Various Fields

In the realm of classical computing, supercomputers such as Japan's Fugaku have proven their extraordinary ability to handle very intensive computational tasks. Fugaku, once ranked as the world's fastest supercomputer, is used to simulate global weather and climate at a level of detail never before achieved. Fugaku's computing power still allows scientists to create complex atmospheric and ocean models, predict weather patterns with greater accuracy, and provide more accurate early warnings of natural disasters such as hurricanes and typhoons [31].

In addition, classical computing also plays a vital role in Big Data analysis an increasingly relevant field in today's digital age. With its ability to process vast volumes of data, classical computing allows us to identify hidden patterns, trends, and correlations that are invisible to the naked eye. The valuable insights generated from big data analysis can be used in a variety of fields, from business and marketing, where companies can understand consumer behavior and develop more effective strategies, to healthcare, where researchers can analyze genomic data to understand diseases and develop more personalized treatments [32].

B. QUANTUM ARCHITECTURE

1) Basic Principles

Quantum computing is based on quantum mechanical principles such as superposition, entanglement, and interference. The basic unit of quantum computing is the qubit, which can represent 0 and 1 simultaneously through superposition. This allows for much more efficient parallel processing for certain types of problems compared to classical architectures.

2) Main Components

a. **Qubit:** The basic unit in quantum computing that can be in multiple states at once.

b. **Quantum Gates:** Quantum gates that manipulate qubits by applying linear transformations according to the principles of quantum mechanics.

c. **Quantum Circuit:** A combination of various quantum gates to process information and produce output.

3) Performance and Efficiency

a. Capable of performing exponentially parallel computations through **superposition** and **entanglement**.

b. Capable of performing exponentially parallel computations through **superposition** and **entanglement**.

c. Technical challenges include **error correction**, **decoherence**, and qubit stability [9].

4) Application in Vairous Fields

On the other hand, quantum computing opens ~~up~~ new opportunities for solving problems that are difficult or even impossible to solve with classical computers. In drug development, companies such as IBM and Google are leveraging quantum computers to simulate molecular interactions more accurately and efficiently. This allows scientists to understand how drugs interact with biological targets at the atomic level, thereby speeding up the discovery of new drugs and reducing research and development costs. In addition, quantum computing also has excellent potential in cryptography. With its ability to break conventional encryption algorithms currently used to secure sensitive data, quantum computing is driving the development of new, more potent encryption methods, such as quantum cryptography, which leverages the principles of quantum mechanics to create security systems that are more resistant to attacks from future quantum computers [3], [21], [29], [33].

2. RESEARCH METHODS

Classical computing employs the best available algorithms for factoring large numbers, such as the General Number Field Sieve (GNFS). However, classical processors are not fast enough to efficiently handle the factorization of large numbers used in modern cryptographic systems like RSA. These processors face both physical and mathematical limitations, particularly in handling problems with exponential complexity. In contrast, with qubit-based processors and Shor's algorithm, quantum computing offers a more efficient solution for factorization by reducing the time complexity from exponential to polynomial. This quantum advantage presents a potential threat to classical cryptographic systems. However, quantum processors are still in their infancy and require significant improvements in stability and the number of qubits to become practical. One of the most compelling factors is the performance difference between classical and quantum computers in factoring 2048-bit numbers, a computational challenge that is crucial to modern cryptography, particularly RSA encryption.

A. DATA SOURCE AND TOOLS

Our study involved a technical comparison between classical computing (utilizing the GNFS algorithm) and quantum computing (using Shor’s algorithm). The performance data for GNFS were derived from prior research and simulation results, while the theoretical performance of Shor's algorithm was assessed using existing quantum processors. The tools and resources used in this research include [software used], [datasets], and [specific quantum simulators], allowing for accurate simulation of both classical and quantum algorithm performance.

B. CRITERIA FOR COMPARISON

The comparison between classical and quantum computing focused on the following key criteria:

1) **Time Complexity:** We analyzed the theoretical and simulated time complexity required to factor 2048-bit numbers.
2) **Computational Resources:** This included CPU and memory usage for GNFS in classical systems, and the qubit requirements for running Shor's algorithm in quantum systems.
3) **Challenges:** We compared the challenges faced by both computing paradigms, such as the stability of qubits in quantum computing and the physical limits of transistor-based classical processors.

C. ALGORITHM PERFORMANCE

The GNFS algorithm's performance was evaluated using (tool), while Shor's algorithm was tested through a (quantum simulator) under specific parameters. The main metrics for comparison included execution time, computational complexity, and error rates, providing a comprehensive overview of the current state of classical versus quantum factorization.

The future of quantum computing presents both challenges and opportunities, particularly as a potential breakthrough in solving problems that classical computing cannot handle efficiently. Several key areas will shape the trajectory of quantum computing:

1) **Scalability of Qubits:** For quantum computing to reach its full potential, quantum processors will need to increase the number of stable qubits significantly. Current quantum computers are limited in the number of qubits they can maintain without substantial error, which hinders their scalability for practical applications like factoring large numbers.
2) **Quantum Error Correction:** One of the most significant challenges facing quantum computing is the high error rates caused by qubit instability and noise. Quantum error correction techniques must improve to reduce these errors and enable reliable, large-scale quantum computations.

3) **Quantum Supremacy:** Quantum supremacy refers to the point at which quantum computers can outperform classical computers for specific tasks. While isolated demonstrations of quantum supremacy in experimental settings have been made, further advancements are needed to make this a reality for practical and widespread computational problems, such as breaking RSA encryption.

Although quantum computing is still in its early stages, its future looks promising, especially in fields like cryptography, optimization, and simulation of complex systems. However, classical computing will continue to dominate for the foreseeable future, especially in tasks that do not require quantum efficiency or are constrained by the current limitations of quantum hardware. In the long term, hybrid systems that integrate classical and quantum computing may offer the most practical solutions, leveraging the strengths of both paradigms for maximum computational power.

3. LITERATURE REVIEW

The classical quantum computing debate primarily revolves around the future of cryptography. If quantum computing achieves large-scale, error-free operation, it could render current cryptographic systems, particularly RSA and similar schemes, obsolete. This possibility has sparked significant research into post-quantum cryptography, which aims to develop encryption algorithms resistant to quantum attacks. While classical encryption remains dominant for now, this could change as quantum hardware evolves.

In summary, classical computing powers today's digital infrastructure, but quantum computing promises to more efficiently address challenges like large-number factorization. However, this potential will only be realized if significant technical hurdles are overcome. The future likely lies in a hybrid approach where classical and quantum systems complement each other in specific applications.

Table 3.1 Technical Comparison Table

Aspect	**Classical Computing**	**Quantum Computing**
Data Units	Bit (0 or 1)	Qubit (superpositio n of 0 and 1

		simultaneously)
Logic	Boolean logic gates	Quantum gates (Hadamard, Pauli, etc.)
Algorithm	Deterministic, linear	Probabilistic, exponential parallel
Processing	Limited serial and parallel	Exponential parallel (due to superposition)
Speed	Depends on clock speed	Depends on the number of qubits and coherence time
Main Challenges	Physical limitations of transistors, power	Quantum errors, qubit stability

A. PERFORMANCE COMPARISON BETWEEN CLASSICAL AND QUANTUM COMPUTING

The main differences between the two technologies above can be seen in several key issues, namely:

1) Factoring Large Numbers

Being a prime factor is a task that becomes exponentially harder for classical computers as the size of the number increases. This problem is very relevant in cryptography, especially for RSA encryption, which is used in many digital security systems.

2) Classical Computing

a. Best Algorithm

Classical computers use factoring algorithms such as the GNFS (General Number Field Sieve) algorithm, which is a two-stage algorithm, as quoted from the website

In the first stage, the algorithm selects a large smoothness limit B and finds two integers, x, and y, such that x2 = y2 (mod n), where n is the integer to be factored. The algorithm then creates a polynomial of degree B using these two integers and uses sifting to find a set of B-smooth numbers. These B-Smooth numbers are used in the second stage of the algorithm.

In the second stage, the algorithm creates a matrix using the B-Smooth numbers found in the first stage. This matrix has dimensions (r+s) xr, where r is the number of prime factors of the smooth number, and s is a parameter that depends on the size of the factored number. The algorithm then uses Gaussian elimination to solve the matrix and find the factors of the factored integers [36].

This is the fastest factoring algorithm for large numbers, but it is still exponential in time complexity.

Performance: To factor a number with 2048 bits (as used in the RSA encryption system), a modern classical computer would take thousands to millions of years to complete the factoring. This is because of the exponential complexity that increases with the size of the number.

Time Complexity: For factoring very large numbers (such as 2048 bits), the complexity of GNFS is exponential, i.e.

$$\exp\left(\left((64/9)^{1/3}+o(1)\right)(\log n)^{1/3}(\log\log n)^{2/3}\right)$$

$$= L_n\left[1/3, (64/9)^{1/3}\right]$$

This makes it inefficient for huge numbers.

3) Quantum Computing

a. Best Algorithm

Quantum computers, using Shor's Algorithm, are able to solve factoring problems with polynomial time complexity, which is much more efficient than classical approaches [36].

b. Performance

With Shor's Algorithm, theoretical quantum computers are able to factor numbers of the same length in hours or days, much faster than classical computers. This is due to the quantum advantage in solving problems involving operations on superposition and entanglement of qubits [36].

Table 3.2 Comparison of Shor's Factorization Algorithm and General Number Field Sieve (GNFS)

Function	Average CPU Utilization (%)	Average Core Temperature (°C)	Running time (hours)
Polynomial	25	65	0.40
Sieving	100	82	1.33
Relations	100	82	0.02
Matrix	100	82	0.06
Square root	100	82	0.01

As shown in Table 3.2, where the research is taken from [36], the implementation of GNFS, although efficient, still requires a long time (around 1 hour and 50 minutes for a 100-digit number) and significant computing resources. This study also highlights that most of the functions in the GNFS implementation support multithreading.~~,~~ However, the main challenge is the high temperature generated during the computing process, which can damage the system if not handled properly. Therefore, a sophisticated cooling system is needed. In addition, this study also shows the potential for using GPUs, especially with CUDA technology from Nvidia, to significantly improve GNFS performance [9].

One of the main challenges in developing quantum computers is the stability of qubits and quantum errors. Qubits, which are the basic units of information in quantum computers, are highly susceptible to environmental disturbances such as temperature fluctuations, electromagnetic radiation, and other interference. This instability leads to significant quantum errors, and while quantum computers promise incredible computing power, the challenges of maintaining qubit stability and managing these errors have not been fully solved.

In the context of Shor's algorithm, which runs on large quantum computers, the issue of qubit stability becomes even more crucial [37]. This algorithm can reduce the computation required to extract the private key from an asymmetric cypher such as RSA, which protects almost all internet traffic and encrypted data today [34]. However, to achieve this capability, quantum computers must have a large number of stable qubits so that the computation process can proceed without errors that disrupt the final result.

The importance of qubit stability and error management is also evident in the practical implications of Shor's algorithm for cryptography. Asymmetric cryptography is currently the foundation of Internet security. If Shor's algorithm can run on a stable quantum computer, the computational complexity required to crack a private key will be drastically reduced, threatening data security. Almost all internet traffic, from financial transactions to confidential communications, uses asymmetric encryption that is threatened by the power of quantum computers. This means that if qubit stability can be achieved and quantum errors minimized, quantum computers have the potential to break encryption that is considered secure by classical computing systems.

For this reason, the development of quantum error correction algorithms is essential. These algorithms are designed to overcome errors that occur during quantum operations and improve the stability of qubits. In the long term, these challenges must be overcome to ensure that quantum computers are not only able to run algorithms like Shor's but also to do so with precision and without errors that could invalidate the computation's results.

As a solution, the cybersecurity world is currently developing post-quantum cryptography, which is designed to resist attacks from quantum computers. This effort is a proactive step in dealing with potential threats from quantum computers, which ultimately depends on solving the challenges of qubit stability and quantum error control.

4) **Challenges of Classical Computing**

a. **Moore's Law Limits:** Increasing the number of transistors per chip is becoming increasingly difficult due to physical limitations such as transistor sizes approaching atomic scale [8],[9].

b. **Power and Heat Management**: Computing speed increases with increasing power consumption and heat production [8],[9].

5) **Quantum Computing Challenges**

In quantum computing, the main challenges faced are **quantum errors** and **qubit stability** [8],[9]. Both of these problems are closely related to the nature of quantum mechanics, which is highly susceptible to external disturbances. The following is a more detailed discussion of each challenge:

a. **Qubit Stability**: Qubits are very sensitive to external disturbances, which causes **decoherence problems**

b. **Quantum Error Correction**: Still under development to correct errors that often occur in qubit operations[13][24][25].

c. **Physical Complexity**: Quantum hardware development requires extreme cooling technologies and special conditions to maintain system stability[5][22][33].

6) **Quantum Error**

Quantum errors are problems that arise when qubits experience disturbances from their environment during a computation. Unlike classical bits that have only two fixed states (0 or 1), qubits are in a superposition of multiple states at the same time[5][13][24]. This makes qubits much more vulnerable to external disturbances, which can corrupt the information stored in them. There are several types of quantum errors to consider:

a. **Physical Complexity**: Quantum hardware development requires extreme cooling technologies and special conditions to maintain system stability [5][22][33].
b. **Bit Flip Error** A bit-flip error occurs when a qubit that should be in state ∣0⟩|0\rangle∣0⟩ accidentally flips to ∣1⟩|1\rangle∣1⟩, or vice versa. This can be caused by disturbances from the environment, such as electromagnetic radiation or temperature fluctuations [18], [19].
c. **Phase flip error** occur when the quantum phase of a qubit is flipped, although its bit value remains unchanged. For example, a qubit in the superposition 12(∣0⟩+∣1⟩)\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)21 (∣0⟩+∣1⟩) can flip into 12(∣0⟩−∣1⟩)\frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)21(∣0⟩−∣1⟩). These errors are so complicated to detect and correct because they involve a phase change [18], [19].
d. **Depolarizing Error**: In a depolarization error, the qubit loses its correlation with the initial state and falls into a mixed state, where it is no longer in an apparent superposition. This causes a loss of information and makes the computational results invalid [18], [19].
e. **Decoherence**: is the most severe type of quantum error that occurs when a qubit interacts with its environment on a macroscopic scale. In this process, the quantum superposition of the qubit collapses, turning the qubit into one of the classical states before the computation is complete. Decoherence is one of the most fundamental challenges in maintaining the stability of quantum systems [18], [19].

7) Quantum Error Correction

Quantum Error Correction (QEC) techniques have been developed. Unlike error correction in classical computers, QEC faces more significant challenges because it is impossible to directly observe the state of a qubit without disturbing its superposition [26],[28-29]. Here are some of the main approaches to quantum error correction:

a. **Qubit Redundancy**: One of the main methods in QEC is to use **redundant qubit codes**. For example, instead of using a single qubit to store information, a number of qubits are used in parallel to ensure that errors can be detected and corrected without directly observing the individual states of the qubits [26],[28-29].
b. **Shor Code** : One of the most famous schemes is **Shor's Nine-Qubit Code** , which uses 9 qubits to protect 1 qubit of information. This scheme is capable of correcting both bit-flip and phase-flip errors simultaneously, although it requires significant quantum overhead [26],[28-29].
c. **Topological Quantum Error Correction**: This technique uses quantum topological properties to protect quantum information from errors. For example, **the toric code** proposed by Alexei Kitaev creates a quantum system that is more stable against physical errors by encoding information in a topological quantum form, which is more resistant to external disturbances [26],[28-29].

8) Qubit Stability

Qubit stability is a crucial factor in determining the success of quantum computing. In quantum architecture, qubit stability refers to how long a qubit can maintain its quantum state (either superposition or entanglement) before being affected by external perturbations [1],[5],[7]. There are two main aspects of qubit stability:

a. Coherence Time (Coherence Time)

Coherence time refers to the duration that a qubit can remain in superposition without experiencing decoherence. The longer the coherence time, the more quantum operations can be performed before the qubit loses its information. In current qubit systems, the coherence time is often minimal, in the range of milliseconds or even microseconds, depending on the type of qubit used (e.g., trapped ion qubits or superconducting qubits) [1],[3],[5],[7],[21],[29],[33].

b. Error Rates

The error rate is how often a qubit produces an incorrect result during a quantum operation. In current quantum systems, the error rate is still relatively high, around 1% per quantum gate. Reducing this error rate is one of the main challenges in bringing quantum computing to a practical level. To achieve more stable and scalable quantum supremacy, the error rate must be on a very low scale, around 0.0001% or less [1],[3],[5],[7],[21],[29],[33].

c. Decoherence

Decoherence is one of the leading causes of qubit stability loss and can be triggered by various factors, such as interactions with the environment (especially at high temperatures or fluctuating electromagnetic fields). Efforts to reduce decoherence include cooling quantum systems to near absolute zero temperature and minimizing external disturbances [1],[3],[5],[7],[21],[29],[33].

9) Approaches to Improve

To improve qubit stability and extend coherence time, some approaches being explored include:

a. **Extreme Cooling**: Many quantum systems (such as superconducting qubits) require temperatures near **absolute zero** to minimize the thermal interference that causes decoherence[3][5][37].
b. **Topological Qubit Technology**: Topological quantum approaches focus on using topological particles (such as anyons) to create qubits that are more resistant to external disturbances and quantum errors. This is expected to significantly improve quantum stability[5][37].
c. **Near-Qubit Environment Reduction**: Keeping qubits in an electromagnetically isolated space and ultra-high vacuum to reduce interference from the surrounding environment[5][37].
d. **Quantum Error Correction Technology Improvement**: Developing more efficient error correction algorithms to detect and correct errors without sacrificing too many computing resources [18][24][25].

B. FUTURE PROSPECTS

1. Classical Computing

It is still dominant for everyday applications, with a focus on increasing parallelism and architectural optimization. However, it is important to remember that "Cryptography is the foundation of information security," as emphasized by Stallings (2017). Thus, advances in classical computing must also be accompanied by developing more robust cryptographic algorithms to protect data from evolving threats [34].

2. Quantum Computing

It can potentially revolutionize fields such as cryptography, molecular simulation, and artificial intelligence. However, widespread adoption requires advances in stability and error correction. In a YouTube video by Cleo Abram titled Quantum Computers, explained with MKBHD**,** In the context of data security, quantum computing could threaten current encryption methods, necessitating the development of quantum-resistant cryptography to protect privacy and information security[1][33][34]. In the pharmaceutical field, the ability of quantum computing to simulate molecular interactions with precision could accelerate the discovery of new, more effective and personalized drugs, thereby improving quality of life and reducing healthcare costs[2][33][37].

Furthermore, quantum computing is projected to enhance the capabilities of artificial intelligence in tasks such as pattern recognition and natural language processing, with significant implications for sectors including transportation, manufacturing, and customer service[4][9]. Amazon Bracket simplifies access to quantum computing resources, enabling users to experiment with quantum algorithms using familiar tools like Jupyter notebooks. This setup allows developers to create, visualize, and collaborate on quantum projects more efficiently[6][15]. By modeling complex molecular processes, the quantum simulation will enable breakthroughs in pharmacology (life-saving drug development) and materials science (chemicals essential to industry, battery chemistry, and more). Breakthroughs in quantum optimization could also impact the financial sector (risk analysis and financial portfolio optimization)[10]. Quantum Computing as a Service

(QCaaS) is viewed as a solution attuned to the philosophy of service-orientation that can offer QC resources and platforms, as utility computing, to individuals and organizations who do not own quantum computers[11], the concept of Quantum Software as a Service (QSaaS) through a quantum API gateway. It outlines the challenges and opportunities of providing quantum computing capabilities via cloud-based services. The authors emphasize the need for standardized interfaces to facilitate access to quantum resources and enhance collaboration between developers and quantum service providers[12][16].

4. CONCLUSION

Classical computing and quantum computing have their ~~own~~ strengths and weaknesses. Classical computing remains the dominant solution for general problems, while quantum computing promises breakthroughs in solving problems beyond classical computers' reach. The future of computing will likely involve a combination of these two approaches, with specific applications taking advantage of the strengths of each architecture. With advances in qubit stability and quantum error correction, the potential for quantum computing to become a powerful and practical computing technology is increasing. However, until these challenges are fully overcome, quantum computing applications are still limited to specific problems and cannot wholly replace classical computing. Although quantum computing offers the potential for extraordinary performance in specific problems, classical computing is still the preferred choice for practical applications today due to its stability and scalability. However, with further development, quantum computers may overcome these limitations and be widely used.

5. REFERENCES

[1] Harrow, A. W., & Montanaro, A. (2017). Quantum computational supremacy. *Nature*, 549(7671), 203–209.

[2] Ahmad, A., Khan, A. A., Waseem, M., Fahmideh, M., & Mikkonen, T. (2022). Towards process-centered architecting for quantum software systems. In *2022 IEEE International Conference on Quantum Software* (pp. 26–31). IEEE.

[3] Ahmad, A., Waseem, M., Liang, P., Fehmideh, M., Khan, A. A., Reichelt, D. G., & Mikkonen, T. (2023). Engineering software systems for quantum computing as a

service: A mapping study. *arXiv preprint arXiv:2303.14713*.

[4] Ali, S., Yue, T., & Abreu, R. (2022). When software engineering meets quantum computing. *Communications of the ACM*, 65(4), 84–88.

[5] Andrikopoulos, V., & Lago, P. (2021). Software sustainability in the age of everything as a service. *Next-Gen Digital Services: A Retrospective and Roadmap for Service Computing of the Future*, 35–47.

[6] Biswas, R., Jiang, Z., Kechezhi, K., Knysh, S., Mandra, S., O'Gorman, B., Perdomo-Ortiz, A., et al. (2017). A NASA perspective on quantum computing: Opportunities and challenges. *Parallel Computing*, 64, 81–98.

[7] Bobier, J.-F., Langione, M., Tao, E., & Gourevitch, A. (2021). What happens when 'if' turns to 'when' in quantum computing. *Boston Consulting Group*.

[8] Bouguettaya, A., Singh, M., Huhns, M., Sheng, Q. Z., Dong, H., et al. (2017). A service computing manifesto: The next 10 years. *Communications of the ACM*, 60(4), 64–72.

[9] Bouguettaya, A., Singh, M., Huhns, M., Sheng, Q. Z., Dong, H., et al. (2022). Global cloud computing revenue by segment 2015-2022 — Statista. Retrieved from https://stagingfr.statista.com/statistiques/540499/worldwide-cloud-computing-revenue-by-segment/

[10] Cao, Y., Romero, J., & Aspuru-Guzik, A. (2019). Potential of quantum computing for drug discovery. IBM Journal of Research and Development, 62(6), 1-11.

[11] Castelvecchi, D. (2017). IBM's quantum cloud computer goes commercial. *Nature*, 543(7644).

[12] Chang, W.-L., & Vasilakos, A. V. (2021). *Fundamentals of Quantum Programming in IBM's Quantum Computers*. Springer.

[13] Chong, F. T., Franklin, D., & Martonosi, M. (2017). Programming languages and compiler design for realistic quantum hardware. *Nature*, 549(7671), 180–187.

[14] Cusumano, M. A. (2018). The business of quantum computing. *Communications of the ACM*, 61(10), 20–22.

[15] De Stefano, M., Pecorelli, F., Di Nucci, D., Palomba, F., & De Lucia, A. (2022). Software engineering for quantum programming: How far are we? *Journal of Systems and Software*, 190, 111326.

[16] Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods,

and analytics. International Journal of Information Management, 35(2), 137-144.

[17] Garcia-Alonso, J., Rojo, J., Valencia, D., Moguel, E., Berrocal, J., & Murillo, J. M. (2021). Quantum software as a service through a quantum API gateway. *IEEE Internet Computing*, 26(1), 34–41.

[18] Gemeinhardt, F., Garmendia, A., & Wimmer, M. (2021). Towards model-driven quantum software engineering. In *2021 IEEE/ACM 2nd International Workshop on Quantum Software Engineering* (pp. 13–15). IEEE.

[19] Gonzalez, C. (2021). Cloud-based QC with Amazon Braket. *Digitale Welt*, 5, 14–17.

[20] https://softwaredominos.com/home/science-technology-and-other-fascinating-topics/integer-factorization-algorithms-a-comparative-analysis/ (30/09/2024 16:00 WIB)

[21] Humble, T. S., McCaskey, A., Lyakh, D. I., Gowrishankar, M., Frisch, A., & Monz, T. (2021). Quantum computers for high-performance computing. *IEEE Micro*, 41(5), 15–23.

[22] Keen, M., Ackerman, G., Azaz, I., Haas, M., Johnson, R., Kim, J., & Robertson, P. (2006). Patterns: SOA foundation-business process management scenario. *IBM Redbooks*.

[23] Leymann, F. (2019). Towards a pattern language for quantum algorithms. In *Quantum Technology and Optimization Problems: First International Workshop, QTOP 2019* (pp. 218–230). Springer.

[24] Lin, S. F., Sussman, S., Duckering, C., Mundada, P. S., Baker, J. M., et al. (2022). Let each quantum bit choose its basis gates. In *2022 55th IEEE/ACM International Symposium on Microarchitecture* (pp. 1042–1058). IEEE.

[25] Murali, P., McKay, D. C., Martonosi, M., & Javadi-Abhari, A. (2020). Software mitigation of crosstalk on noisy intermediate-scale quantum computers. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* (pp. 1001–1016).

[26] National Academies of Sciences, Engineering, and Medicine. (2019). Quantum Computing: Progress and Prospects. The National Academies Press, Washington, DC.

[27] Nistala, P., Nori, K. V., & Reddy, R. (2019). Software quality models: A systematic mapping study. In *2019 IEEE/ACM International Conference on*

Software and System Processes (pp. 125–134). IEEE.

[28] Petersen, K., Vakkalanka, S., & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64, 1–18.

[29] Pirandola, S., Andersen, U. L., Banchi, L., Berta, M., Bunandar, D., Colbeck, R., ... & Wallden, P. (2020). Advances in quantum cryptography. Advances in Optics and Photonics, 12(4), 1012-1236.

[30] Raymer, M. G., & Monroe, C. (2019). The US National Quantum Initiative. *Quantum Science and Technology*, 4(2), 020504.

[31] Riedel, M., Kovacs, M., Zoller, P., Mlynek, J., & Calarco, T. (2019). Europe's quantum flagship initiative. *Quantum Science and Technology*, 4(2), 020501.

[32] Shah M. H, Syed T. Z, Firoz M, Biprodip Pal. (2014) A Compare between Shor's Quantum Factoring Algorithm and General Number Field Sieve (Tabel I. Execution Time Environment Details Of GGNFS Software Package).

[33] *State of Quantum Computing: Building a Quantum Economy*. (2023, January). World Economic Forum. Retrieved from https://www.weforum.org/reports/state-of-quantum-computing-building-a-quantum-economy/

[34] *The World is Heading for a 'Quantum Divide': Here's Why It Matters*. (2023, January). World Economic Forum. Retrieved from https://www.weforum.org/agenda/2023/01/the-world-quantum-divide-why-it-matters-davos2023/

[35] Weigold, M., Barzen, J., Leymann, F., & Vietz, D. (2021). Patterns for hybrid quantum algorithms. In *Service-Oriented Computing: 15th Symposium and Summer School, SummerSOC 2021* (pp. 34–51). Springer.

[36] Yashiro, H., Miyoshi, T., Seko, H., Sato, T., Sueyoshi, T., & Tomita, H. (2021). Overview of the Fugaku Supercomputer System. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 1-12.

[37] Zhao, J. (2020). Quantum software engineering: Landscapes and horizons. *arXiv preprint arXiv:2007.07047*.