Grover's Algorithm Using IBM's Fake Providers

Frank Acasiete[1]*, Anderson Buarque[1]
[1]QuIIN, SENAI CIMATEC University; Salvador, Bahia, Brasil

In this work we will show Grover's algorithm using Fake Providers which are images of real quantum computers to give us an idea of their behavior in the implementation in Qiskit and we will show the fidelities with respect to the quantum simulator which gives us the result closest to the analytical one.
Keywords: Quantum Computing. Quantum Circuit. Quantum Algorithm.

A significant milestone in the implementation of quantum computers was established by Bennett [1], who introduced the concept of reversible computing as a model. Furthermore, Toffoli [2] made a substantial contribution by developing a universal set of gates for this model in classical computation.

Shor's Algorithm [3] has garnered considerable interest within the academic community. It has proven capable of efficiently solving challenging problems, such as the Integer Factorization Problem and the Discrete Logarithm Problem, when executed on a quantum computer.

In a notable experiment [4], Grover's search algorithm was successfully implemented on a quantum system with four states. This represented an important step in practically demonstrating the efficiency of quantum algorithms compared to their classical counterparts.

Preliminaries

In this section, we'll discuss Grover's Algorithm and Fake Providers. These are the essential tools we'll use to achieve the results presented in this work.

Grover's Algorithm

Grover's algorithm can find the marked element with high probability using only $O(\sqrt{N})$ function evaluations, compared to classical search which evaluations [5]. Grover's algorithm is a powerful tool in quantum computing, especially in search and optimization problems where efficiency is crucial.

The main barrier to demonstrating acceleration with Grover's algorithm is that the quadratic speedup achieved is too modest to overcome the significant overhead of near-term quantum computers.

In Grover's algorithm, we consider a function f: $\{0, 1,..., N-1\} \rightarrow \{0, 1\}$ as input. In the "unstructured database" analogy, the function's domain represents the indices of a database, and $f(x) = 1$ if and only if the data pointed to by index x satisfies the search criteria. We assume that only one index, called ω, and our goal is to identify this index.

To access the function f, we use a subroutine, also known as an oracle, in the form of a unitary operator U_ω, which acts as follows:

$$U_\omega|x\rangle = -|x\rangle \text{ for } x = \omega, \text{ when, } f(x) = 1,$$

$$U_\omega|x\rangle = |x\rangle \text{ for } x \neq \omega, \text{ when, } f(x) = 0.$$

This uses the N-dimensional state space H, which is provided by a register with $n = \lceil log_2 N \rceil$ qubits. This is often written as:

$$U_\omega|x\rangle = (-1)^{f(x)}|x\rangle. \quad (1)$$

In this algorithm, we use an N-dimensional state space H, which is provided by a register with $n=\lceil log_2 N \rceil$ qubits. In this way, the operator $U\omega$ acts on the quantum states represented by the qubits, applying a negative sign (-1) to the state $|x\rangle$ if $f(x) = 1$, and keeping the state $|x\rangle$ unchanged if $f(x) = 0$.

Received on 21 March 2026; revised 26 May 2026.
Address for correspondence: Frank Acasiete. Av. Orlando Gomes, 1845 - Piatã, Salvador – BA – Brazil, Zipcode: 41650-010. E-mail: frank.quispe@fbter.org.br.

J Bioeng. Tech. Health 2026;9(7):730-734

The oracle $U\omega$ has its behavior defined by the equation:

$$U_{\omega}: |x\rangle|a\rangle = |x\rangle|a \oplus f(x)\rangle. \quad (2)$$

where $|x\rangle \in H^N$, $|x\rangle \in H^2$, and $\oplus$ represents the XOR operation.

Grover's algorithm produces the value ω with a probability of at least 1/2 using $O(\sqrt{N})$ applications of $U\omega$. The Grover matrix, represented by G, is defined by:

$$G|x\rangle = (\tfrac{2}{N} - 1)|x\rangle + \tfrac{2}{N}\sum_{i \neq x}|i\rangle. \quad (3)$$

The Grover evolution operator, U_G, is given by:

$$U_G = GU_{\omega} \quad (4)$$

The algorithm's initial condition is:

$$|\psi_0\rangle = |G\rangle|-\rangle, \quad (5)$$

where $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{N}$ and $G|x\rangle = \frac{1}{\sqrt{N}}\sum^{N-1}_{u=0}|u\rangle$. Grover's algorithm instructs us to iteratively apply U_G approximately $\lfloor \frac{\pi}{4}\sqrt{N} \rfloor$ times.

After these iterations, we measure the first register in the computational basis, and the result will be x with a probability greater than or equal to $1 - \frac{1}{N}$.

Fake Providers

The fake providers module allows developers to create realistic simulations of IBM quantum computers without needing physical access to a real device. It does this by using "snapshots" of actual quantum systems, which capture information like qubit connectivity and the types of errors that can occur [6].

Imagine you're building a new program for a quantum computer. Instead of testing your program directly on a real device (which can be expensive and hard to access), you can use fake providers to create a simulation. This simulation will mimic the behavior of a real quantum computer, including the common errors and inaccuracies found in these systems.

Why Use Fake Backends?

- Testing and Development: It lets you test and debug quantum algorithms before running them on a real device.
- Learning: It's an excellent tool for learning about quantum computing and understanding how errors affect quantum calculations.
- Optimization: It helps optimize quantum circuits before executing them on a real device.

Limitations

- Accuracy: Simulations might not be perfectly accurate, as the snapshots are from real systems that can change over time.
- Complexity: Simulating large-scale quantum systems can be computationally expensive.

While fake providers use snapshots of real quantum systems for its simulations, qiskit-aer offers a variety of simulators because it's a more general library. The fake providers system uses the most recent calibration results.

Results

We now show the results of the quantum simulation of Grover's algorithm for 4 qubits (Figure 1) for two cases of 1 and 2 marked elements, ω_1=0111 and ω_2 = 1101 (Figure 2). We will compare it with 3 fakes providers and finally present the fidelity of the results using the Hellinger metric:

$$h^2 = \tfrac{1}{2}\sum_x(\sqrt{p_x} - \sqrt{q_x})^2,$$

where p is the probability distribution of the q simulation and is the quantum computer.

Figure 3 presents Fake Osaka and Fake Auckland yield quite similar results. Notably, Fake LagosV2, despite having the lowest fidelity value, still provides good outcomes. This demonstrates

Figure 1. Grover's algorithm circuit for 4 qubits.

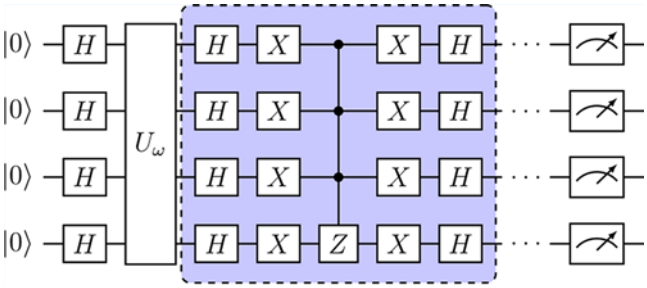
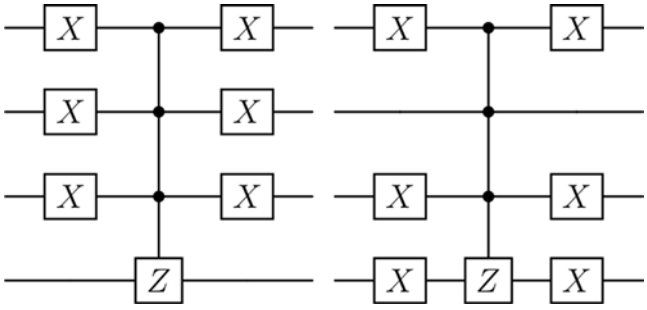


Figure 2. Circuits of the oracles of $\omega_1 = 0111$ and $\omega_2=1101$.



that Fake Providers exhibit behavior akin to real quantum computers. In Figure 4, we observe similar behavior, specifically for the case of two marked elements. Furthermore, Table 1 shows the fidelities of these providers when compared to the quantum simulator.

As mentioned, with these backends, it's possible to implement our quantum models and see how they behave. With these, we could observe, for example, quantum superiority in certain problems. Another useful feature would be that, since they already include the error of real quantum computers, we can use this as an exercise to reduce circuits through equivalences, improving our results and using this as a reference for future developments.

We also have to take into account the high cost of using real quantum computers today (with 10 minutes of free monthly use on IBM's QPUs), which limits research and development in this area.

Figure 3. Probability distribution of Grover's Algorithm with 4 qubits and one marked element

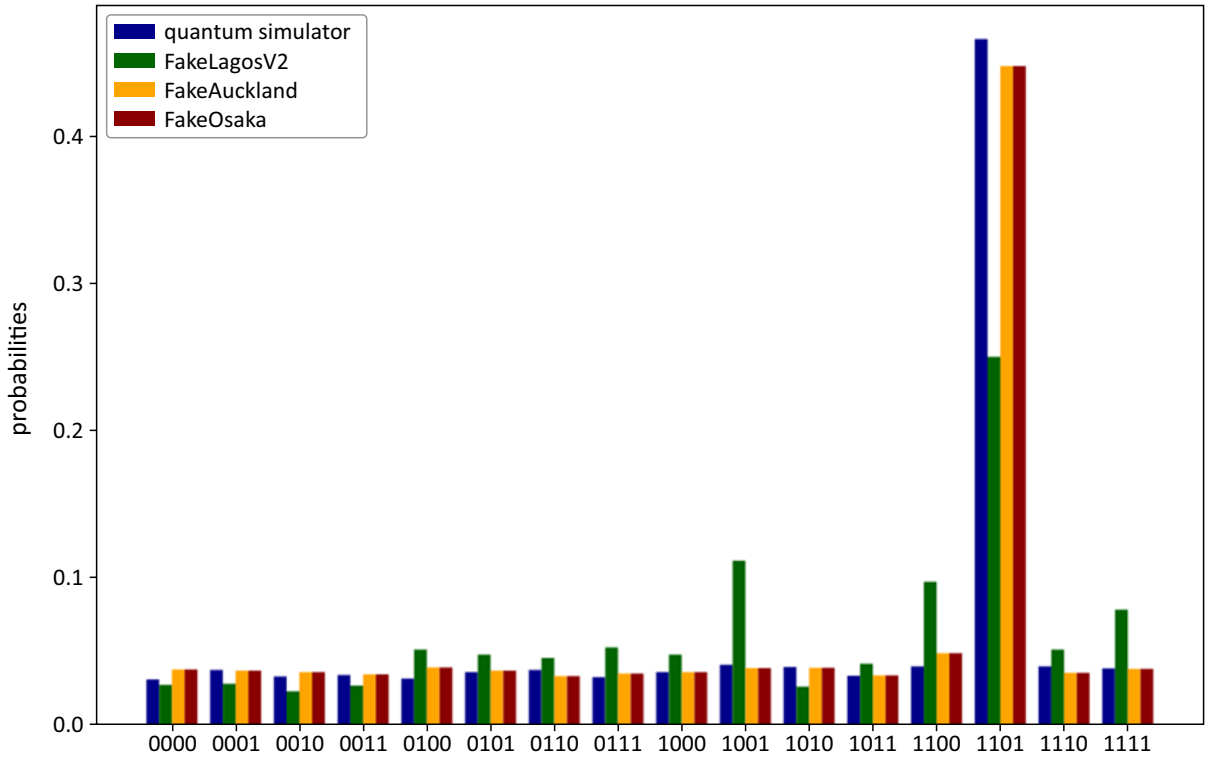


Figure 4. Probability distribution of Grover's Algorithm with 4 qubits and two marked elements.

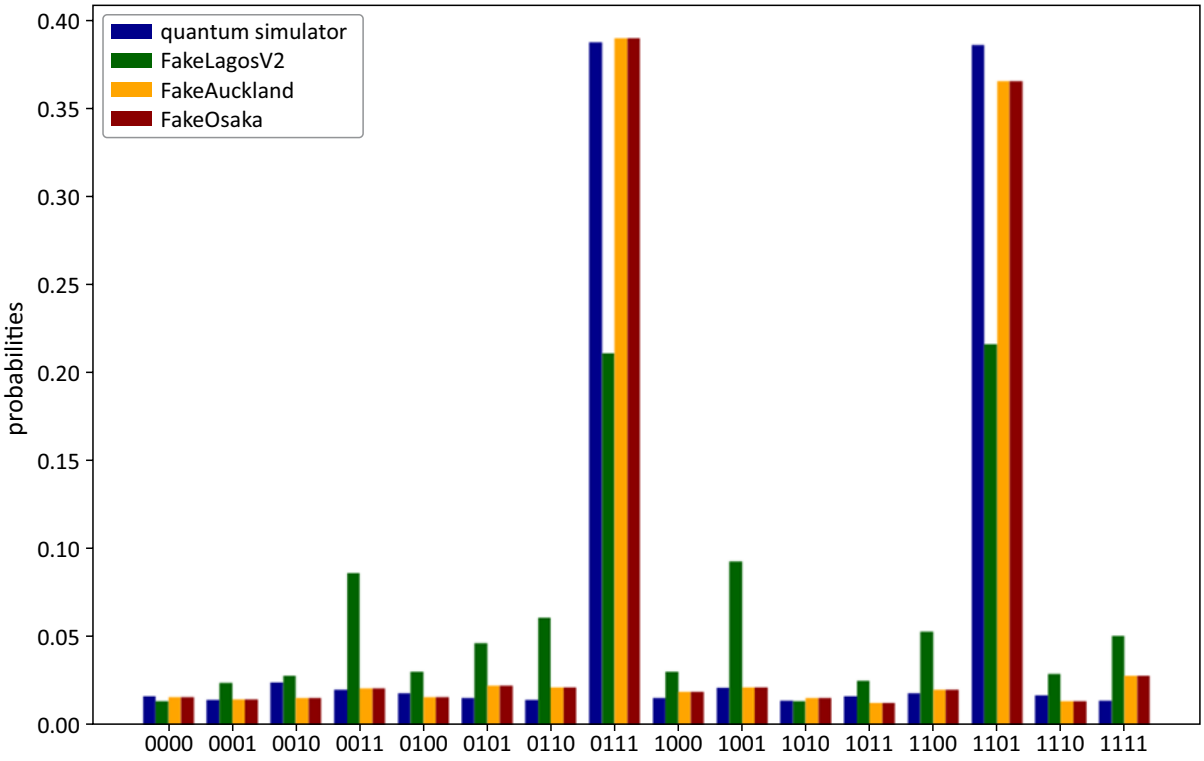


Table 1. Fidelity of Grover's algorithm with 4 qubits, using IBM's Fake Providers.

N° marked	Fake LagosV2	Fake Auckland	Fake Osaka
1	0.958	0.999	0.998
2	0.922	0.996	0.993

Conclusion

With these results, we can see that Fake providers are a tool for observing the behavior of quantum computers to determine if our algorithm implementations are performing as desired.

Although these are only snapshots of real quantum computers, most of which have already been retired, they show us how these computers have evolved over the years and that the error rate is decreasing, leading to the possibility of one day having a real, logical quantum computer.

Acknowledgement

This work was entirely funded by Project QIN-AFCCT-FC-CC-2024-6-11-1, supported by QuIIN-Quantum Industrial Innovation, the EMBRAPII CIMATEC Competence Center in Quantum Technologies. This funding was secured through MCTI Call number 053/2023, established with EMBRAPII.

References

1. Bennett CH. Logical reversibility of computation. IBM J Res Dev. 1973;17(6):525-532.
2. Toffoli T. Reversible computing. In: Proceedings of the 7th Colloquium on Automata, Languages and Programming; 1980. Berlin: Springer-Verlag; 1980. p. 632-644.
3. Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. 1999;41.
4. Chuang IL, Gershenfeld N, Kubinec M. Experimental implementation of fast quantum searching. Phys Rev Lett. 1998;80(15).

5. Grover LK. A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing; 1996. New York: Association for Computing Machinery; 1996.

6. IBM Quantum. qiskit_ibm_runtime.fake_provider module [Internet]. IBM; [cited 2025 Jun 20]. Available from: https://docs.quantum.ibm.com/api/qiskit-ibm-runtime/fake_provider