Reliability of IBM's Public Quantum Computers

Raquel Pérez-Antón, Alberto Corbi Bellot, José Ignacio López Sánchez, Daniel Burgos

Universidad Internacional de La Rioja (UNIR)

ABSTRACT

One of the challenges of the current ecosystem of quantum computers (QC) is the stabilisation of the coherence associated with the entanglement of the states of their inner qubits. In this empirical study, we monitor the reliability of IBM's public-access QCs network on a daily basis. Each of these state-of-the-art machines has a totally different qubit association, and this entails that for a given (same) input program, they may output a different set of probabilities for the assembly of results (including both the right and the wrong ones). Although we focus on the computing structure provided by the "Big Blue" company, our survey can be easily transferred to other currently available quantum mainframes. In more detail, we probe these quantum processors with an ad hoc–designed computationally demanding quaternary search algorithm. As stated, this quantum program is executed every 24 hours (for nearly 100 days) and its goal is to put to the limit the operational capacity of this novel and genuine type of equipment. Next, we perform a comparative analysis of the obtained results according to the singularities of each computer and over the total number of executions. In addition, we subsequently apply (for 50 days) an improvement filtering to perform noise mitigation on the results obtained proposed by IBM. From our continuous and long term tests, we derive that tangible room still exists regarding the improvement of quantum calculators in order to guarantee enough confidence in the returned outcomes.

LA UNIVERSIDAD EN INTERNET

KEYWORDS

quantum computers, quality control, IBM Q, quantum search algorithms

I. Introduction

Quantum computing is a very promising and radically incipient area of knowledge in contrast with the current limitations of classical computing. For instance, classic transistors have a finite physical volume, and we are already approaching the 1 nm limit . Theory predicts that after surpassing this physical dimension, manipulating the flow of the electric current (without the loss of electrons) may be operationally impossible [1]. That is why, additional computational technologies may be needed to expand the ability to solve some (new) complex problems or perform extremely convoluted calculations.

According to the scientific community, quantum computing may be the solution for addressing these issues. However, despite all the efforts in this discipline, the challenges are still overwhelming. Perhaps, the biggest one has to do with the stability of the quantum states and the way they are organized (entangled). This characteristic is known as coherence and depends on the number of qubits in a quantum computer (QC). Coherence in quantum computing can be defined as the conservation of the superposition state of a system over time. This property causes the system to be extremely sensitive to interferences from the environment, and it can be destroyed by simple mechanical vibrations, electromagnetic disturbances, sound waves, tiny seismic temblors, or even adverse weather effects. When this takes places, the quantum wave functions collapse as if they were being measured, and the system loses its multistate nature. This unavoidable process is known as decoherence. The maintenance of coherence (or avoidance of decoherence) in this type of hardware is essential for the correct implementation and execution of quantum instructions and the derivation of the expected (and accurate) results.

Therefore, quantum error correction (QEC) is necessary in quantum computing to protect information from errors caused by decoherence and other sources of noise at the quantum level (see [2]–[5]). However, while classic error correction uses the redundancy process to counteract errors (storing the information several times in such a way that if the copies are not the same later, you can choose the option that is generally present) this possibility is no longer feasible in quantum computing according to the no-cloning theorem [6]. Even though this theorem seems to present an obstacle to formulating a theory of QEC, some alternative strategies still exist. One of these has to do with the spread of qubits through highly entangled states of several neighboring physical qubits in such a way that a state inversion event could be detected without the need for consulting the exact value of the examined qubit (which would destroy the information). These qubit aggregates (making up a compound logical qubit) are resistant to errors in the final computer. Clearly, this means that if a program requires 10 qubits to run, in practice, it will need 10 logical qubits, which can be translated into hundreds or thousands of the original physical ones. These systems, called *noisy intermediate scale quantum computers* or NISQ, are expected to provide the advantages necessary to meet the required QEC [7], [8].

All errors can be corrected if the imperfections of quantum operations and measurements are below a certain threshold and the correction can be applied repeatedly [9], [10]. However, these error thresholds also depend on the details of the physical system and quantifying them requires careful analysis of both the hardware and software implementation [11].

Although some studies have addressed the quantification of these error baselines for different platforms and QC configurations (see [12]), it is a relatively new area which needs further clarification through experimentation. For this reason, in this work, the reliability of the coherence of IBM's public

* Corresponding authors:
E-mail address: mail@mail.com (First A. Author), mail@mail.com (Second B. Author), mail@mail.com (Third C. Author).

quantum computers has been examined. The choice of the Big Blue's network of QCs has not been arbitrary. Two factors have influenced this decision. On one hand and for several years, IBM has provided, free of charge, some of its QC infrastructure for research and study. On the other, each piece of equipment is designed differently and with a contrasting qubit number, arrangement, and entangling layout.
In more detail, our experiment consisted of executing for almost 100 days, 1024 times each day, the same quaternary search algorithm (described in Section IV) on 8 IBM public quantum computers. For the sake of completeness, the characteristics of IBM's public quantum processors as well as the environmental conditions in which they are designed to operate are tackled in Section II. Our results are then presented and discussed in Section V. Finally, some conclusions are drawn in Section VII.

II. IBM's public quantum processors

In 2016, IBM deployed and made publicly available the first 5-qubit cloud quantum processor. This was followed by others that were organized into families according to their number of qubits. Each family was named after a bird. Thus, we have the 5-qubit processors, which formed the Canary family, the 16-qubit processors such as Albatross, Penguin with 20 qubits, etc. In addition, within each family, the processors are named after a city, so within the Canary family are London, Rome, Vigo, etc. Melbourne is a 16-qubit processor included in the Albatross family.

Category	Qubits	Processors
Canary	5	Tenerife, Yorktown, Ourense, London, Vigo, Rome, Burlington, Valencia, Santiago
Albatross	16	Melbourne
Penguin	8–16	Austin, Tokyo, Poughkeepsie, Johannesburg, Singapore, Almaden, Boeblingen
Hummingbird	+16	Raleigh

Table 1: List of names and categories according to the number of qubits of IBM computers [Chow & Gambetta, 2020]

A. Main Characteristics and Components of the IBM Q Equipment

As it is shown in Table 2, the characteristics of each quantum computer in which the experiment (detailed in Section 4) has been carried out, as well as their corresponding numbers of qubits. The type of gates that were used for the design and construction of their circuits is also included. In more detail, u1, u2, and u3 are the three parameters that allow the building of any single qubit gate and have a duration of one unit of time [13]. In addition, the error rate that each door could develop at the time of the measurement is also referenced. This ratio will increase as time passes if the QC is not properly calibrated.
Specifically, IBM performs these calibrations twice a day on each quantum processor and conveniently keeps the users informed so that they can take them into account when eventually launching their programs. Calibration consists of carrying out a series of experiments to obtain precise information about the physical behaviour of each qubit. The values of the parameters that characterize a qubit are different for each qubit within the processor and among different processors, and these can even vary over time. It is possible to identify the qubit's proper frequency by sweeping through a range of frequencies and observing absorption signals. The qubit's frequency is the energy difference between the ground state and the excited state. Aside from calibration, these processors must remain in specific environmental conditions. They also need a temperature close to absolute zero 0 K (−273.144 °C) to better account for the Heisenberg's uncertainty principle [14].
IBM public quantum computer hardware uses the characteristic known as superconductivity. The materials with this property can carry electrical currents without the resistance or loss of energy under specific circumstances. From an architectural point of view, superconductors are wrapped in the form of Josephson joints [15]. These structures are formed using two sheets of aluminum, which, under normal environmental circumstances, would behave like classical electrical circuits. However, in the subatomic world, they operate as quantum gates.

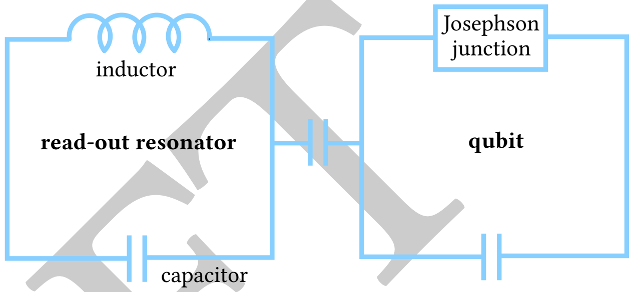

Fig. 1. Superconductivity diagram with Josephson joint (1 qubit).

The transition between the possible states of the qubits is generated by applying a certain level of energy, and because of the tunneling effect, the particle crosses the barrier (with some probability). The state of the qubit can be read by observing the energy of each aluminum sheet.

B. Quantum Computer Connectivity

As stated above, each IBM public quantum processor has a different physical architecture. Nevertheless, the logic of any given quantum algorithm can be applied independently of the subjacent hardware. Even so, it is important to know the internal structure of these computers for several reasons. To begin with, the circuit will use, at most, all the qubits of the processor only once since the algorithms are according to the principles of their construction [16]. Furthermore, the application of a logic gate to several qubits requires, for greater efficiency, that they be physically interconnected in their architecture. Therefore and a priori, the more qubits a processor has and the greater their interconnection, the better the expected results.

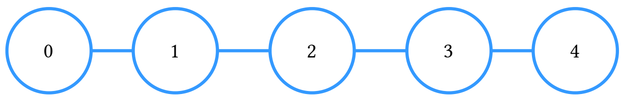

Fig. 2. Qubit arrangement for the Rome quantum computer. The structure of this processor is iterative, linking all the qubits of the processor in an orderly fashion.

As can be seen in Fig. 3, Fig. 2 and Fig. 4, each IBM public QC refers to a different connectivity or type of entanglement depending on the number of qubits and their arrangement. Each circle represents a qubit, and from this possible entanglement lines emerge towards the contiguous qubits. The evolution in connectivity and lattice design is one of IBM's ongoing investigations, as shown in [17]. The debugging of errors in the

Name	Qubits	Error Rate Door CNOT	Basic Doors	Single-qubit u2 Error Rate
Melbourne	15	2.384e-2/1.000e+0	id, u1, u2, u3, cx	4.632e-4/3.482e-2
London	5	9.411e-3/1.430e-2	u1, u2, u3, cx, id	3.369e-4/4.624e-4
Burlington	5	9.009e-3/2.075e-2	u1, u2, u3, cx, id	3.568e-4/6.144e-4
Essex	5	8.434e-3/1.406e-2	u1, u2, u3, cx, id	3.929e-47.155e-4
Ourense	5	6.851e-3/2.976e-2	u1, u2, u3, cx, id	2.961e-4/9.845e-4
Vigo	5	7.772e-3/1.470e-2	u1, u2, u3, cx, id	3.673e-4/8.616e-4
Yorktown	5	1.280e-2/2.203e-2	u1, u2, u3, cx, id	6.106e-4/7.950e-4

Table 2: Qubit error rate and the characteristics of each active public IBM Q Experience processor.

gates and exposure to crosstalk is linked to the connectivity among the qubits. Therefore, the new processors are built on improvements in previous structural experiences.

III. Design of the Quaternary Search Algorithm

In classical computing, a binary tree is a data structure widely used in dynamic memory programming. Each node of the complete tree can have a left and a right child, where its complexity in the search for ordered elements in the best case is as follows:

$$(O(\log_2(N)), \tag{1}$$

where N is the number of nodes, T is the time, and n is the data to search. The algorithm presented here uses the data structure of a tree, but in this case, it exploits the intrinsic characteristics of the entanglement of qubits, thus managing a quaternary tree. Each node has four children. The complexity associated with searching in this structure will be considerably reduced in the best case if we transform it into a quaternary tree [18]:

$$O\left(\frac{1}{4\pi}\cdot \ln(N)-1\right), \tag{2}$$

Besides, our search algorithm forces an iterative entanglement to verify the consistency and stability of the qubits. Therefore, if we take as a reference Grover's basic search algorithm [19], the number of iterations is equal to:

$$\sum U_f = (n_{\text{qubits}})-1 \tag{3}$$

where U_f is the so-called oracle (i.e. the unitary operator) and f is a Boolean function. According to Eq. (3), each iteration performs the addition of amplitudes until it approaches 1. As we can see, n_{qubits} will need $n-1$ iterations to find the element of the list regardless of whether the first or last element is found. On the contrary, if we search a quaternary tree using the initial state of entanglement of two qubits $\{00, 01, 10, 11\}$, as shown in Fig. 5, the number of oracles to be used will be equal to the number of levels in the tree:

$$\sum U_f = \sum L, \tag{4}$$

where L is the number of levels in the tree and U_f is the number of oracles. Furthermore, each oracle will return the maximum possible amplitude, so unlike Grover's algorithm we only need to apply the oracle once in each iteration. In our case, we have reduced the application of oracles to two iterations. The result of each oracle is concatenated with the next one until they are finalized in a leaf of the tree. The entanglements of the qubits in each of the oracles are detailed next:

- $L1 = \{00, 01, 10, 11\}$, where the entanglement is formulated by $q[3]$ and $q[2]$. We generate a vector of states in the 4-D Hilbert space (H^4). The expression $q[i]$ stands for qubit q and i is the position of the qubit in the circuit.
- $L2 = \{0000, 0001, 0010, 0011, 0100, 0101, \ldots\}$, where the entanglement will be formulated by $q[3]$, $q[2]$, $q[1]$, and $q[0]$, if and only if the element has not been found in $L1$. We generate a vector of states in the 16-D Hilbert space.

To achieve this, first the Pauli gate X has been applied to $q[2]$, with the aim of changing the initial state. Then the qubits $q[3]$ and $q[2]$ have been interlaced applying the following Hadamard gates that perform a rotation π around the XZ axis:

$$H\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad H\,|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\,,$$

or as a matrix:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{5}$$

The state of entanglement of two qubits is determined by the Einstein–Podolsky–Rosen (EPR) pair [20]:

$$|q[2]\,q[3]\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}\,.$$

Therefore, this state $|q[2]\,q[3]\rangle$ cannot be decomposed into pure states since no combination of complex coefficients fulfills both descriptions. Therefore, as an alternative to assembling pure states, it is possible to describe mixed states through the matrix or density operator ρ, explained in [21]. We then define the assembly of pure states as the set $\{\rho_i \mid \psi\}$ where ρ_i are all possible states of and thus $\sum \rho_i = 1$. Then the density operator or density matrix is the result of the entanglement of several qubits:

$$\rho = \sum_i \rho_i\,|\psi_i\rangle\langle\psi_i|\,,$$

or in matrix form:

$$\rho = \frac{1}{N} I\,,$$

where N is the number of possible states in its measurement and I is the identity matrix. For example, the density matrix for a single qubit will be $\rho = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. In our case, with 2 qubits, the generated initial density matrices will be:

$$\{\rho \mid q[2]\,q[3]\} = \frac{1}{4}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

And for 4 qubits:

$$\{\rho \mid q[2]\,q[3]\,q[1]\,q[0]\} = \frac{1}{16}\begin{pmatrix} 1 & \cdots & 0 \\ \vdots & 1 & \vdots \\ 0 & \cdots & 1 \end{pmatrix},$$

where the set $\{\rho \mid q[0] \ldots q[n]\}$ defines the state probabilities of the entangled qubits. In our case, we want to find state 01 at level $L1$ and then state 0111 at level $L2$. Hence, we change the sign of the amplitude of the states we are looking for such that:

$$U_{01}\,|01\rangle = -\,|01\rangle\,,$$

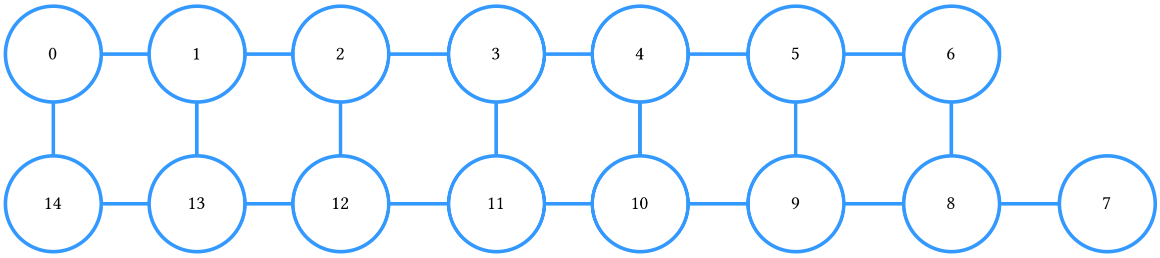

Fig. 3. Qubit arrangement for the Melbourne quantum computer. This architecture is used on processors with more than 16 qubits.

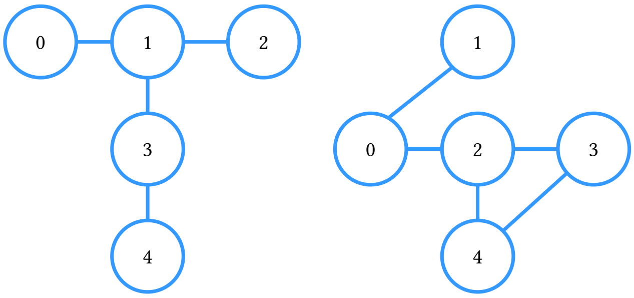

Fig. 4. Qubit arrangement for the London, Burlington, Essex, Ourense (left) and Yorktown (right) quantum computers. These architectures are a composite of Melbourne and Rome since they combine the lattice structure with the iterative one.

$$U_{01}\,|q[2]\,q[3]\rangle = |{-q[2]\,q[3]}\rangle\ ,$$

$$U_{0111}\,|0111\rangle = -\,|0111\rangle\ ,$$

$$\begin{aligned}U_{0111}\,|q[2]\,q[3]\,q[1]\,q[0]\rangle &= -\,|q[2]\,q[3]\,q[1]\,q[0]\rangle \\ \forall \quad & q[2]\,q[3]\,q[1]\,q[0] \neq 0111\end{aligned}\ .$$

After that, the unit transformation of the oracle U is applied. The oracles that must be applied to the algorithm are scalar according to the number of search qubits. In addition, we will only need one oracle for each level, thus reducing the Grover's algorithm:

$$L1 = (I-2)\,|\omega_0\rangle\langle\omega_0|) \qquad L2 = (I-2)\,|\omega_1\rangle\langle\omega_1|)\,.$$

The oracles for 2- and 4-qubit entanglement are:

$$U_{\omega_0} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{6}$$

$$U_{\omega_1} = \begin{pmatrix} 0_{0000,0000} & \cdots & 0_{0000,1111} \\ \vdots & 1_{0111,0111} & \vdots \\ 0_{1111,0000} & \cdots & 0_{1111,1111} \end{pmatrix}, \tag{7}$$

where $\omega_0 = 01$ and $\omega_1 = 0111$. The resulting algorithm forces the qubits to iteratively intertwine during execution. In other words, the entanglement result of the first two qubits of the first level $L1$ given by Eq. (6) of the tree will continue to entangle with the second level $L2$ given by Eq. (7) and so on (if we expand the tree and, consequently, the number of qubits). Therefore, it should be noted that the qubits are not all initialized entangled, but rather additions are made to the source entanglement. One of the necessary conditions to generate a quantum circuit is the lack of breaks in the code. In this quaternary search algorithm, searching for an element of the first level L1 given by Eq. (6) would be impossible. All oracles must be evaluated, and the result obtained is stored in a sheet of the second level L2 given by Eq. (7).

After that, we apply Grover's star operator:

$$G_f = (2\,|\omega_0\rangle\langle\omega_0| - I)\ ,$$

to increase the amplitude of the element to be found. Recall that Grover performs a search on unordered items, but in our case, they are ordered. Therefore, in principle, applying this oracle should ensure a successful outcome for the desired element. The operator defined in Grover's algorithm to increase the amplitude consists of applying the inverse of U_f, in our case $L1^{-1}$) and $L2^{-1}$. Considering the following assertions:

$$L1 = (I - 2\,|\omega_0\rangle\langle\omega_0|) \qquad L2 = (I - 2\,|\omega_1\rangle\langle\omega_1|)\ ,$$

$$L1^{-1} = (2\,|\omega_0\rangle\langle\omega_0| - I) \qquad L2^{-1} = (2\,|\omega_1\rangle\langle\omega_1| - I)\ ,$$

$$|S_0\rangle = |q[2]\,q[3]\rangle \qquad |S_1\rangle = |q[2]\,q[3]\,q[1]\rangle\ ,$$

where $|S_0\rangle$ and $|S_1\rangle$ are auxiliary notations for the entanglement states, then the resulting equation of the algorithm is:

$$\begin{aligned} L1\,|S_0\rangle\,L1^{-1} + L2\,|S_1\rangle\,L2^{-1} = \tfrac{1}{4\sqrt{4}}\Big((4-4)\textstyle\sum_{S_0\neq\omega_0}|S_0\rangle + 8\,|\omega_0\rangle\Big) \\ +\tfrac{1}{16\sqrt{16}}\Big((16-4)\textstyle\sum_{S_1=\omega_1}|S_1\rangle + 44\,|\omega_1\rangle\Big) \end{aligned} \tag{8}$$

A graphical and step-by-step representation of the execution of this quaternary search algorithm is shown in Fig. 6. The Qiskit framework [22] generates this graphical timeline automatically from the Python code available at GitHub. IBM has brought quantum computing closer to the public by giving access to its processors and providing a series of intuitive tools for conducting experiments. In addition, it announced in 2020 the quantum educator program which introduces training in this discipline in the classroom. To complete the teaching material, IBM offers the open-source textbook *Learn Quantum Computing Using Qiskit*. Thanks to the initiatives of the Big Blue, many students are able to train in this discipline, which would otherwise be impossible for them [23]. The code in Listing 1 shows the OpenQASM code behind the graphical representation.

Both in Fig. 6 and Listing 1, we see the zero entry of the 4 qubits used defined as q0_0,q0_1,q0_2 and q0_3. After that, the following methodology is used on the timeline:

- Steps a and b: We initialise the qubit q0_3 to one, applying the Pauli gate X (U3) and then we interlace the states of the qubits q0_2 and q0_3 by applying the Hadamard operator (U2).
- Step c: We apply the gate CZ (Pauli Z (U1) conditioning factor) where the state we want to find is activated, in our case 01.
- Steps d, e, f, g and h: We combine the Pauli Z (U1) and X (U3) doors to perform the unitary operator and its inverse.
- Step i: We introduce the second-level qubits of the trees q0_0 and q0_1 initialising them to one applying the Pauli gate X (U3).
- Steps j, k, l, m and n: We combine the Pauli Z (U1) and X(U3) doors to perform the unitary operator and its inverse.
- Steps o, p, q and r: We measure the output of each qubit.

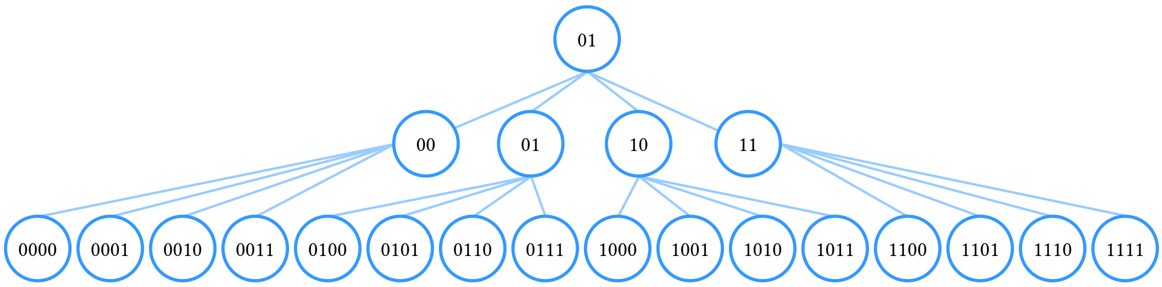

Fig. 5. Graph representation of the complete quantum quaternary tree.

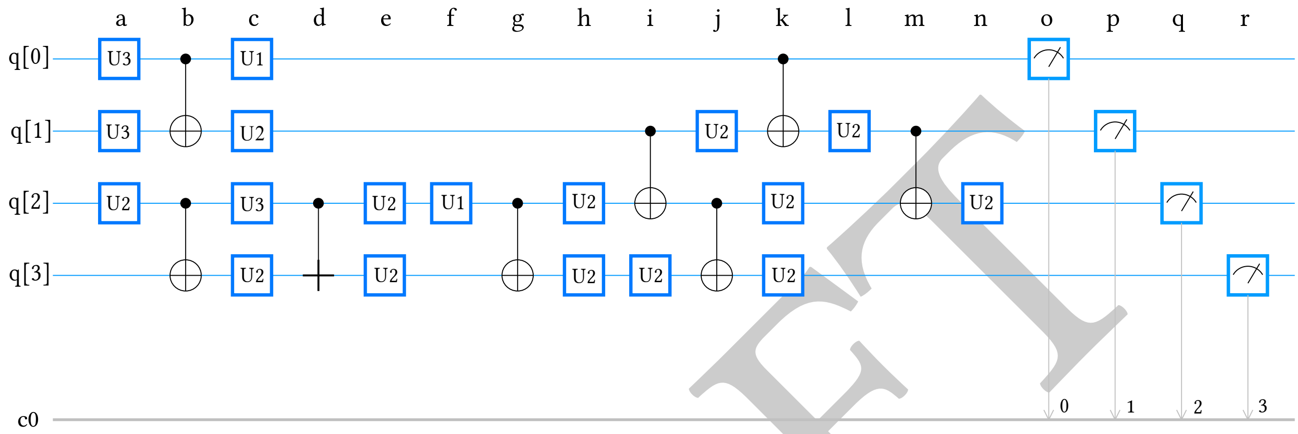

Fig. 6. Graphical representation (from the IBM OpenQASM 2.0 specification [22]) of the quaternary search algorithm used to probe the stability of quantum hardware. The nomenclature used by OpenQASM 2.0 indicates each turn on the qubit made by the unitary matrix or gate applied in the U form U(theta, phi, lam); where U2 corresponds to the Hadamard gate, U3 to Pauli X gate and U1 to Pauli Z gate.

IV. Data collection phase

As explained in Section 1, the algorithm described above has been run, on a daily basis for almost 100 days. On each execution job, each of the IBM quantum computers (introduced in Section II.B) performed the quaternary search 1024 times. Some data is missing because of occasional and prolonged maintenance/offline periods. In addition, we have considered the calibration timetable of each piece of hardware by launching our probing code both before and after this housekeeping phase. However, as we can see in Fig. 7, no type of improvement has been observed in the search for the desired data just after the daily calibration.

Each run job produced, among other outcomes, a daily histogram (like the example shown in Fig. 7) with the probabilities of the sought result. The data of all the probabilities of the elements have been recorded to be later analysed.

Through a graphical interface or manual code insertion with a simple Jupyter notebook written in the high-level Python language, real qubits have been used, the algorithm has been run online, and experiments have been carried out on these processors. Next, we detail the daily results of the behaviour of each remote quantum computer, in addition to making a generic evaluation of all of them.

As seen in Fig. 2, Fig. 3 and Fig. 4, each computer registers a certain initial interlacing architecture. However, the execution process is supposed to generate all the necessary interleaves for any given execution. Table 3 shows the relationship between the original entanglements for each computer and the proposed quaternary search algorithm.

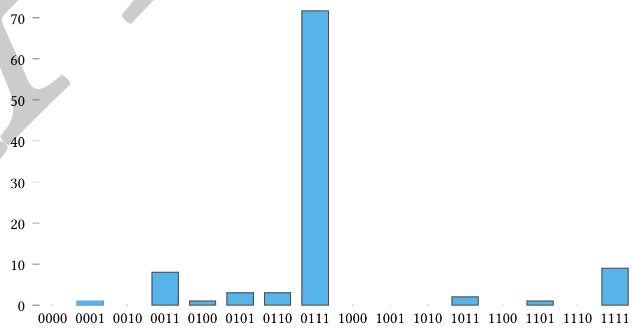

Fig. 7. Histogram of results obtained in the Yorktown computer on June 10, 2020 (after calibration). The 0111 value is the correct one.

V. Results

Next, we detail the behaviour of each piece of remote hardware when running the quantum program described in Section III. It is possible to establish two main categories: those who show a stable trend (and low variability) over time and those who exhibit erratic performance.

A. Quantum computers with a stable trend over time

As shown in Fig. 9, the Yorktown computer present a stable trend for the probability (70%) of the right outcome (0111). Furthermore, the difference between all the obtained values obtained does not exceed 5%.

Listing 1 OpenQASM code of the quaternary search algorithm presented in this research work (equivalent to Fig. 6). Comments (lines beginning with #) signal the start of the carried out steps.

```
include "qelib1.inc";
qreg q[15];
creg c0[4];
barrier q[0], q[1];
barrier q[0], q[1];
barrier q[0], q[1];
barrier q[0], q[1];
barrier q[0], q[1];
barrier q[0], q[1];
barrier q[0], q[1];
barrier q[0], q[1];
#----------------------- a
u3(3.14, 3.14, 3.14) q[0];
u3(1.57, 3.14, 0) q[1];
cx q[0], q[1];
#----------------------- b
barrier q[0];
u1(3.14) q[0];
u2(0, 3.14) q[1];
u2(0, 3.14) q[2];
u3(3.14, 0, 3.14) q[3];
#----------------------- c
cx q[2], q[3];
u3(1.57, 6.28, 3.14) q[2];
u2(0, 3.14) q[3];
cx q[2], q[3];
u2(0, 3.14) q[2];
barrier q[2], q[3];
u1(3.14) q[2];
u2(0, 6.28) q[3];
cx q[2], q[3];
u2(0, 3.14) q[2];
#----------------------- d
cx q[1], q[2];
u2(0, 6.28) q[1];
cx q[0], q[1];
barrier q[0];
u2(0, 3.14) q[1];
u2(3.14, 3.14) q[2];
u2(0, 3.14) q[3];
barrier q[3];
u2(0, 6.28) q[3];
cx q[2], q[3];
u2(0, 3.14) q[2];
cx q[1], q[2];
u2(0, 3.14) q[2];
u2(0, 3.14) q[3];
barrier q[3];
#------------------ o p q r
measure q[0] -> c0[0];
measure q[1] -> c0[1];
measure q[2] -> c0[2];
measure q[3] -> c0[3];
```

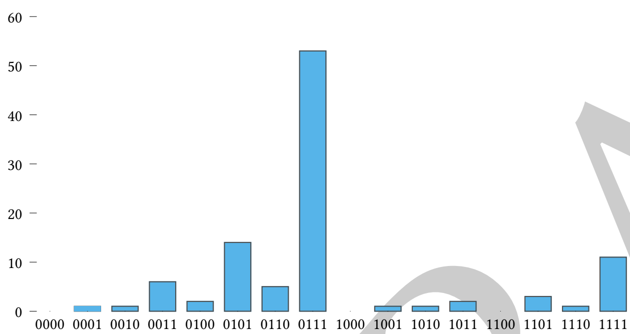

Fig. 8. Histogram of results obtained in the Burlintong computer on May 29, 2020 (after calibration). The 0111 value is the correct one.

B. Unified evaluation of all the quantum computers

In Fig. 10, it is shown a global (and averaged over time) view of the performance of the 8 QCs. As we can see, none of them exceed the 65% average probability. Furthermore, the computer with the highest number of qubits (Melbourne) does not offer the greatest performance, which seems to indicate the importance of entanglement.

VI. Mitigation of Crosstalk

The term Noisy Intermediate Scale Quantum (NISQ) refers to prototype systems with 5-20 qubits that are now available for wide public use [8]. In NISQ systems, a major source of noise such as diphony corrupts quantum states when multiple gates or instructions are executed simultaneously [24]. Noise or crosstalk mitigation in IBM quantum computers is done physically on the hardware through daily calibrations. However, this calibration is impossible for us to carry out and therefore, in this work other mitigation methods are explored through the application of software. One of the basic software tools proposed by IBM for noise mitigation in its quantum computers is the application of filters through noise matrices [25]. IBM's proposal to reduce noise is carried out

	$q[3]$ & $q[2]$	$q[0]$ & $q[1]$	($q[3]$ & $q[2]$) & ($q[3]$ & $q[2]$)
Melbourne	Yes	Yes	Yes
Rome	Yes	Yes	Yes
London	No	Yes	$(q[2])-(q[0]-q[1])$
Burlington	No	Yes	$(q[2])-(q[0]-q[1])$
Essex	No	Yes	$(q[2])-(q[0]-q[1])$
Ourense	No	Yes	$(q[2])-(q[0]-q[1])$
Vigo	No	Yes	$(q[2])-(q[0]-q[1])$
Yorktown	Yes	Yes	Yes

Table 3: Entanglement relationship between the intrinsic architecture of each quantum computer and the entanglement forced by the search algorithm (initial processor interleaving architecture). However, there are possible partial entanglements indicated in the third column.

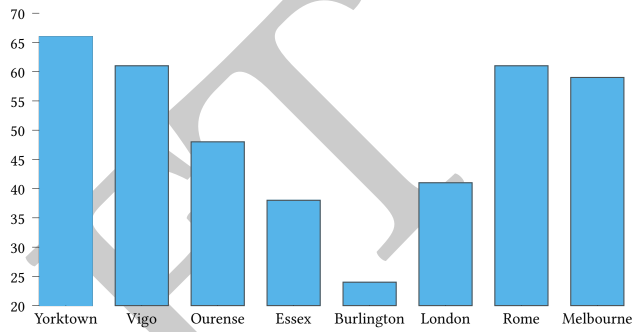

Fig. 9. Average probability of the desired result (0111) through the ~90 days of running the algorithm on several remote quantum computers.

through Qiskit's open `CompleteMeasFitter` class and consists of applying software filters on the initial probabilistic results. These filters are based on the creation of a noise matrix, which houses the deviations from the basic states. Therefore, any other state in superposition will be helped by a weighting in these deviations. Detailed development of this methodology can be found in the open IBM Q Experience documentation as *Measurement Error Mitigation*.

The noise mitigation software has been applied to the same algorithm described in the Section III for 50 days. The reason for running this process over several days is based on the variability of daily calibrations that IBM performs at its facilities. That is, the noise matrix applied to the algorithm is different in each execution, as are the probabilistic results of the quaternary search.

The hypothesis test statistic applied for this case is Wilcoxon [26], since the test variables are adjusted to its methodology. The hypothesis proposed suggests a significant improvement in the initial results by applying noise mitigation. As can be seen in the box diagram in Fig. 11, the improvement exceeds 85% of the expected value. Therefore, we can conjecture that the application of noise mitigation in measurements proposed by IBM gives the expected results.

VII. Conclusions

In this work, the reliability in time of a specific series of public access quantum processors has been studied through the repeated and transversal execution of the same state-of-the-art quantum algorithm. In addition, a quantum decoherence filtering proposed by IBM has been applied with a significant improvement in the results. The objective was to empirically

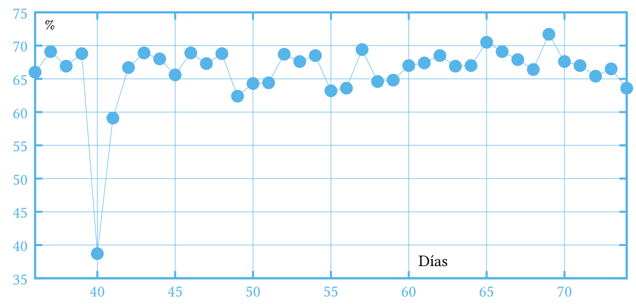

Fig. 10. Results for the Yorktown IBM equipment. Each data point refers to the probability of the desired result (0111) through the days of running the algorithm on this remote quantum computer.

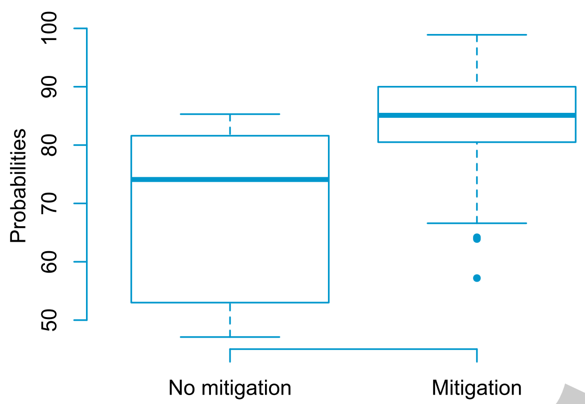

Fig. 11. Result of the application of the Wilcoxon statistic on the noise mitigation values in the quantum computers of IBM Santiago, Bogotá and Yorktown.

demonstrate their current suitability for executing a resource and computational hungry quantum program. The results obtained (probability level of the correct sequences), although sufficient for some specific runs/dates, do not seem to be able to keep up with a homogeneous trend over time. Therefore, in the case of this type of quantum program, quantum computing may not guarantee an adequate level of reliability to make relevant decisions in the business, healthcare or academic worlds. On the other hand, its suitability for research, education and the study and advancement of this technology itself has been amply demonstrated.

Future research may, to begin with, expand the number of daily executions of the algorithm. Furthermore, it could also modify the algorithm and assess the level of acceptance of qubit entanglement on the results. It would also be interesting to analyse the level of error of the quantum gates in greater detail.

VIII. Acknowledgment

This research is partially funded by the Universidad Internacional de la Rioja (UNIR) through the Research Institute for Innovation & Technology in Education (UNIR iTED). The authors would also like to express their gratitude towards IBM for allowing public access to their hardware and for fostering the free use of their remarkable equipment.

References

[1] J. Gao, M. A. Thompson, *et al.*, *Combined quantum mechanical and molecular mechanical methods*, vol. 712. ACS Publications, 1998.

[2] A. Furusawa, J. L. Sørensen, S. L. Braunstein, C. A. Fuchs, H. J. Kimble, E. S. Polzik, "Unconditional quantum teleportation," *science*, vol. 282, no. 5389, pp. 706–709, 1998.

[3] M. Barrett, J. Chiaverini, T. Schaetz, J. Britton, W. Itano, J. Jost, E. Knill, C. Langer, D. Leibfried, R. Ozeri, *et al.*, "Deterministic quantum teleportation of atomic qubits," *Nature*, vol. 429, no. 6993, pp. 737–739, 2004.

[4] T. Aoki, G. Takahashi, T. Kajiya, J.-i. Yoshikawa, S. L. Braunstein, P. Van Loock, A. Furusawa, "Quantum error correction beyond qubits," *Nature Physics*, vol. 5, no. 8, pp. 541–546, 2009.

[5] M. A. Thornton, "Introduction to quantum computation reliability," in *2020 IEEE International Test Conference (ITC)*, 2020, pp. 1–10, IEEE.

[6] W. K. Wootters, W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.

[7] S. S. Tannu, M. K. Qureshi, "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*, 2019, pp. 987–999.

[8] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[9] J. Cramer, N. Kalb, M. A. Rol, B. Hensen, M. S. Blok, M. Markham, D. J. Twitchen, R. Hanson, T. H. Taminiau, "Repeated quantum error correction on a continuously encoded qubit by real-time feedback," *Nature communications*, vol. 7, no. 1, pp. 1–7, 2016.

[10] M. Otten, S. K. Gray, "Recovering noise-free quantum observables," *Physical Review A*, vol. 99, no. 1, p. 012338, 2019.

[11] P. Schindler, J. T. Barreiro, T. Monz, V. Nebendahl, D. Nigg, M. Chwalla, M. Hennrich, R. Blatt, "Experimental repetitive quantum error correction," *Science*, vol. 332, no. 6033, pp. 1059–1061, 2011.

[12] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, C. Monroe, "Experimental comparison of two quantum computing architectures," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3305–3310, 2017.

[13] IBM, "Ibm quantum experience," 2016.

[14] W. Heisenberg, "Schwankungserscheinungen und quantenmechanik," *Zeitschrift für Physik*, vol. 40, no. 7, pp. 501–506, 1927.

[15] B. D. Josephson, "Supercurrents through barriers," *Advances in Physics*, vol. 14, no. 56, pp. 419–451, 1965.

[16] D. P. DiVincenzo, "The physical implementation of quantum computation," *Fortschritte der Physik: Progress of Physics*, vol. 48, no. 9-11, pp. 771–783, 2000.

[17] J. Chow, J. Gambetta, "Quantum takes flight: Moving from laboratory demonstrations to building systems," *IBM Research Blog*, 2020.

[18] P. Høyer, J. Neerbek, Y. Shi, "Quantum complexities of ordered searching, sorting, and element distinctness," *Algorithmica*, vol. 34, no. 4, pp. 429–448, 2002.

[19] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[20] A. Einstein, B. Podolsky, N. Rosen, “Can quantum-mechanical description of physical reality be considered complete?,” *Physical review*, vol. 47, no. 10, p. 777, 1935.

[21] J. P. Hecht, *Fundamentos de Computación Cuántica orientados a la criptología teórica.* 2012.

[22] D. C. McKay, T. Alexander, L. Bello, M. J. Biercuk, L. Bishop, J. Chen, J. M. Chow, A. D. Córcoles, D. Egger, S. Filipp, *et al.*, “Qiskit backend specifications for openqasm and openpulse experiments,” *arXiv preprint arXiv:1809.03452*, 2018.

[23] M. Tilves, “Ibm lleva la computación cuántica a las aulas,” *Silicon.es*, 2020.

[24] P. Murali, D. C. McKay, M. Martonosi, A. Javadi-Abhari, “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*, 2020, pp. 1001–1016.

[25] IBM, “Measurement error mitigation,” 2020.

[26] J. j. Litchfield, F. Wilcoxon, “A simplified method of evaluating dose-effect experiments,” *Journal of pharmacology and experimental therapeutics*, vol. 96, no. 2, pp. 99–113, 1949.

Raquel Pérez-Antón is a PhD Candidate in computer science at the Universidad Internacional de La Rioja (UNIR), Graduated in Computer Engineering from the Universidad Internacional de La Rioja (UNIR), Technical Engineer in Management Computer Science from the Universidad de Alicante (UA), Master's degree in secondary teaching staff specializing in Mathematics and Computer Science from the Universidad Internacional de Valencia (VIU). She currently works as a secondary school teacher in the Higher Degree in Multiplatform Applications Development (DAM) teaching the Programming, Services and Processes modules, and in the Higher Degree in Web Application Development (DAW) as director of end-of-cycle projects.

Alberto Corbi obtained his PhD in Physics at the Universidad de Valencia (UV) and the Institute for Corpuscular Physics (belonging to the Spanish Council for Scientific Research). He also works as a senior researcher at the Research Institute for Innovation & Technology in Education (UNIR iTED) and as an assistant professor at the Engineering School, which are both part of the Universidad Internacional de La Rioja (UNIR). With a background in ocean-atmosphere interaction (M.Sc. obtained at the Universidad Católica de Valencia – UCV), he is currently involved in a variety of research fields: eLearning standards, systems interoperability, server-sided programming languages and solutions medical physics, radiological protection/survey, education-enhanced science education, monitoring of physical activities through inertial sensors (with a special focus on martial arts), social implications of technology (with emphasis on social networks), eHealth advancement (with an accent on Alzheimer's disease characterization and clinical information standards) and environmentalism. He has published over 20 research papers on all the aforementioned subjects, and he is a frequent speaker and knowledge disseminator at radio stations, podcast shows, scientific workshops, general press, academic settings, and outreach events.

Daniel Burgos is the Vice-rector for International Research, director of the UNESCO Chair on eLearning and of the ICDE Chair on Open Educational Resources, at Universidad Internacional de La Rioja (UNIR). He is also director of the Research Institute for Innovation & Technology in Education (UNIR iTED). His work is focused on Adaptive, Personalised and Informal eLearning, Learning Analytics, eGames, and eLearning Specifications. He has published over 150 scientific papers, 20 books and 15 special issues on indexed journals. He has developed +55 European and Worldwide R&D projects. He holds degrees in Communication (PhD), Computer Science (Dr. Ing), Education (PhD), Anthropology (PhD), Business Administration (DBA), and Artificial Intelligence & Machine Learning (postgraduate, at MIT).

Jose Ignacio Sánchez López obtained his PhD in Chemistry at the Universidad de Murcia (UM), while working for the chemical industry as a Torres Quevedo researcher. Previously, he was awarded with a three-year research grant from the Regional Agency for Science and Innovation from Murcia (Fundación Séneca) and had participated in several university-industry research projects. He has held various management positions in the industry as R&D and laboratory director and has participated as co-investigator and PI in Regional, National and European projects. As an Associate Professor at the Engineering School (ESIT, https://www.unir.net/facultades/esit), part of the Universidad Internacional de La Rioja (UNIR, https://www.unir.net), he teaches in environment and prevention of occupational hazards. He has published 17 scientific papers in the areas of chemistry and cognitive performance and co-invented national and international patents. He also attends and exhibits regularly at international conferences.