Quantum Computing: Present and Prospects

Rodrigo Gil-Merino*, Enrique Alba†, José Francisco Chicano†, Zakaria Abdelmoiz Dahi*‡, Gabriel Luque†
*Dept. Lenguaje y Ciencias de la Computación, E.T.S.I. Informática, Universidad de Málaga, Spain
†ITIS Software, Edificio Ada Byron, Universidad de Málaga, Spain,
‡Dept. IFA, Fac. NTIC, Constantine 2 University, Algeria
Emails:gilmerino@uma.es, eat, chicano, gabriel@lcc.uma.es, zakaria.dahi@uma.es, univ-constantine2.dz

***Abstract*—Quantum computing (QC) promises more powerful computers than classical ones and faster solutions to complex problems. Currently, there are two main paradigms in QC: quantum gate computers and quantum annealers. Both technologies are well established and face similar problems: scalability of the number of qubits, robustness of QC, and access to quantum facilities. In this work we review the basis and the state-of-the-art of these two technologies, analyse the aforementioned problems, and describe the near future challenges they face.**

***Index Terms*—Quantum Computing, Quantum Algorithms, Quantum Annealer, Quantum Error Correction, Quantum Computer Languages, Quantum Physics, Artificial Intelligence**

I. Introduction

QC is a new paradigm in computer science. It requires a fresh perspective both in programming and in hardware architecture, since several quantum physical effects must be taken into account when designing quantum software and hardware. The promise land of QC is a set of quantum algorithms (QAs) that outperform classical computation when running on quantum computers. For some classes of computational problems, this already starts becoming a reality, as with the case of the Shor's algorithm to factorize large numbers [1] or the Grover's algorithm to find an element in a unsorted list [2].

Nevertheless, although we understand the quantum nature of QC, we are still far from having multi-purpose personal quantum computers at home. There are different reasons for this: scaling a quantum computer to a large number of qubits, avoiding or significantly decrease quantum errors, increasing quantum fault tolerance, and/or developing quantum software platforms to allow programmers an easy access to all the capabilities of quantum machines.

This work is an attempt to compile the present success and challenges of QC. Our main contributions are the following:

- We describe the current difficulties to scale current quantum computers in terms of the number of qubits needed for practical applications.
- We analyse the current and near future strategies to decrease quantum errors depending on quantum architectures.
- We compare the present and prospects of the two main QC approaches: quantum gate computers and quantum annealing computers,

We organised the rest of the paper in the following sections. In Section II we review the concepts of quantum physics needed to understand QC: superposition, entanglement, parallelism, quantum tunnelling, and measurement. In Section III we expound the transition from classical computing to the two main paradigms in QC: quantum gates-based computers and quantum annealing-based computers. In Section IV we describe the state-of-the-art of quantum computers in number of qubits and its evolution. In Section V we explain the undesired interactions of quantum computers with the environment that induce quantum errors during computational processes and the current framework for quantum error correction and fault tolerant strategies. In Section VI we sketch several possibilities to currently access quantum computers. In Section VII we describe the challenges in QC future research. In Section VIII we explain the some industrial challenges in industrial applications. In the final Section IX we provide our conclusions.

II. Basic concepts for Quantum Computing

The basic blocks of classical computing are the binary units of data called *bits*. The two possible values of a bit are usually represented as 0 and 1, which translates to different voltage levels in a digital electronic circuit.

The basic units of data in QC are the *quantum bits* (*qubits*). Physically, they are usually subatomic particles or photons. Thus, the behaviour of the qubits is better described under the odd properties of quantum physics [3]. In the following, we review four important properties related to QC: superposition, entanglement, parallelism, quantum tunnelling, and measurement.

A. Superposition

The possible states of a qubit are 0, 1 and, additionally, a "mix" of the two. Mathematically, the qubit is a vector in a Hilbert space[1], and the "mix" is expressed as a linear combination of the two basis vectors 0 and 1 of the Hilbert space. Using the *Dirac notation*[2], a qubit $|\psi\rangle$ can be written as $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, where $|0\rangle$ and $|1\rangle$ are the proper vectors, and α, $\beta \in \mathbb{C}$ are the coefficients of the linear combination, called *amplitudes*. The values $|\alpha|^2$ and $|\beta|^2$ represent, according to the laws of quantum physics, the probabilities of finding the system in the state $|0\rangle$ or $|1\rangle$ after a measurement of the

[1]A Hilbert space is an extension of a vector space with the inner product defined as the vector product within the field $\mathbb{C}$ of complex numbers.

[2]The Dirac's notation represents row vectors, called *bra*, as $\langle u|$ and column vectors, called *ket*, as $|v\rangle$, so that the vector product between $\vec{u}$ and $\vec{v}$ is written as $\langle u|v\rangle$.

qubit, respectively, and satisfy $|\alpha|^2 + |\beta|^2 = 1$. This means that, in general, we can only describe the states of a quantum system (here a qubit) in terms of probabilities associated to their amplitudes. Extending the idea to n qubits, we have 2^n possible states that can be combined into a new superposed state, written as:

$$|\phi\rangle = \alpha_0 |0...00\rangle + \alpha_1 |0...01\rangle + ... + \alpha_{2^n-1} |1...11\rangle \quad (1)$$

where $\alpha_i \in \mathbb{C}$ and $\sum |\alpha|^2 = 1$.

Superposition is a key property in QC. In a classical system composed by n bits, we can store values from 0 to $2^n - 1$, but the system processes one of those 2^n values, and only one, at a time. In a quantum system, on the contrary, n qubits are able to process all the 2^n states at the same time. Thus, from an information theory point of view, the qubit is able to store more information than the classical bit. Superposition gives to QC a potential speed-up compare to classical computing.

B. Entanglement

Sometimes we cannot describe the quantum state of a particle within a set independently of the others, we say there is entanglement between the particles. An illustrative case is a pair of particles forming a quantum state of spin zero. The knowledge of the spin of one of the particles "forces" the value of the spin of the other. This pair of particles is *entangled* [4]. In the case of QC, we can explain entanglement as an special case of superposition of multiple qubits.

In general, in a quantum computer not all qubits are entangled. The entanglement of the qubits depends on the physical interconnections between them, and on the physical architecture of the quantum computer. This is an important issue when considering the purpose of the computer and the QAs it will run, as we will see in Section III-B.

C. Parallelism

Along quantum computations, a multiple qubit system $|\phi\rangle$ evolves through unitary transformations[3] [5], which are equivalent to rotations in a Hilbert space. The quantum system $|\phi\rangle$ is a superposition of all the base states of that Hilbert space. Any transformation applied to the system will transform all base states at the same time. The simultaneous transformation of all the possible states is the quantum parallelism. Parallelism is the essence of QC, and the potential source for faster algorithms compared to classical computers [6].

D. Quantum Tunnelling

Quantum tunnelling is the ability of a particle of getting across a potential barrier that is forbidden by classical physics. It is a quantum mechanics effect related to the Heisenberg uncertainty principle. The uncertainty principle establishes that there are pairs of physical quantities of quantum systems that cannot be known with accuracy at the same time, like *position* and *momentum*. Mathematically, the reason for this is that the two quantities are *conjugate variables* in the Hilbert space or, in other words, they are Fourier transforms of one another in that Hilbert space.

This mathematical view helps in understanding that, if a variable associated to a particle is known, then a conjugate variable to the first one will be in a superposition of all its possible states. This means that if, for example, we know the total energy of the particle, then the conjugate variable position[4] should be expressed as a superposition of all the possible states of position, and then some of these possible states will locate the particle at the other side of a potential barrier of energy greater than the total energy of the particle. This is another way of viewing quantum tunnelling, an effect that will explain part of the QC described in Section III-B.

E. Measurement

To measure a physical system means to assign values to the variables that describe the system. In a quantum system, a measurement modifies the state of the system. The reason is because the quantum system might be in a linear combination of all possible states but, when we measure, the measurement will reveal one, and only one, of those states. Mathematically, if the quantum system is a qubit $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, in superposition of the states $|0\rangle$ and $|1\rangle$, after a measurement/observation of the qubit, it will end in one its base states, either $|0\rangle$ or $|1\rangle$.

The lost of superposition of a quantum system, for example a qubit or a set of them, is usually referred to as the *wave function collapse*, because the function that represents the quantum system "collapses" to one of its base states. The collapse of the wave function is a natural end in QC, because after running a QA the final step will be measuring the state of the system. Nevertheless, a set of qubits might also collapse and lose superposition for reasons beyond measurement, such as interaction with the environment, resulting in quantum errors/defaults. The interaction of the quantum system with the environment inducing loss of superposition is called *decoherence*. Preventing quantum computers from quantum decoherence and quantum errors, and increasing their fault tolerance, is a very active field in QC, and one of the big challenges that will be treated in Section V.

III. From Classical to Quantum Computers

Due to the quantum properties already discussed in Section II, QC often needs a different approach when designing algorithms to solve problems. There is not always a quantum counterpart to a classical algorithm or, if it exits, it might not be the best/most efficient solution. QC is a completely new form of programming that we can divide in two basic representative paradigms: one is based on universal quantum gates, used to build quantum circuits and QAs; the other is based on quantum annealing, where we pursue the global minimum of a target function as an optimisation problem.

[3] A transformation U is unitary if the product $UU^\dagger = I$, where $U^\dagger$ is the conjugate transpose of U and I is the identity matrix.

[4] In general, the operator total energy does not conmute with the rest of the usual operators of the system: position, momentum, potential energy and kinetic energy. The total energy is a conjugate variable of all the others [3].

These two paradigms are depicted in Figure III. From a problem definition, one could adopt two different strategies to search for a solution with QC. A first option is to design a QA, convert it into a quantum circuit, and send this circuit configuration to a quantum computer or quantum computer simulator. A second option is to express the problem in the Quadratic Unconstrained Binary Optimisation (QUBO) formalism, called *Ising form*, that consists in defining a cost function in the form of the Hamiltonian of the system, and adapt the QUBO to the hardware of quantum annealing computer. In the following subsections we further explain these concepts in more detail.

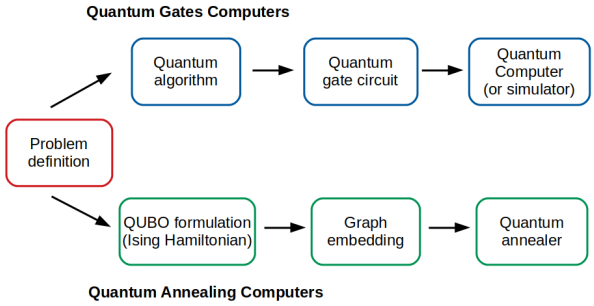

Fig. 1. The two paradigms in QC: quantum gates computers or quantum annealing computers. Although, in principle, we can formulate any problem in the two paradigms, some problems fall more naturally in one of the two.

A. Quantum Gates Computers

The transformations of the quantum states of the qubits are performed through quantum gates. This is similar to the way in which classical bits are transformed using digital gates. Since quantum states can be seen as vectors in a Hilbert space, quantum gate transformations are easily visualised as rotations of these vectors. All quantum gates are unitary transformations and they keep entanglement and superposition. A graphical view makes use of the Bloch sphere from Figure 2: a quantum state is represented by $|\psi\rangle$, written as a superposition of the two basic states in the 2-dimensional Hilbert space: $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ or, alternatively, $|\psi\rangle = cos(\theta)\,|0\rangle + e^{i\phi}\sin(\theta)\,|1\rangle$, where $-\pi/2 \leq \theta < \pi/2$ and $0 \leq \phi < 2\pi$. The quantum gates can also be represented as matrices of the rotations. As an example, three basic quantum gates are shown in Table I. The *Pauli* and *Hadamard* gates act on single qubits, whereas the *controlled-NOT* (C-NOT) acts on two qubits.

In this QC paradigm, QAs must be translated into quantum circuits: a collection of quantum gates that performs the transformations on the necessary qubits. A very simple example of this is shown in Figure 3. There, two qubits are swapped using three consecutive C-NOT gates. This simple circuit is important in the implementation of quantum error correction, and important subfield in QC that we will detail in Section V.

B. Quantum Annealing Computers

The term *annealing* comes from the forges, where metalworkers heat and hammer the steel to improve its quality. This technique allows to slowly mesh the atoms of iron and other metals, resulting in layers of an stronger product. In Physics, it extends to processes that use a heat treatment to change the physical properties of a system, and later slowly cools down and evolves to an state of low inner/total energy.

A classical algorithm inspired in this concept is called *simulated annealing* (SA) [7] and has been applied to many optimisation problems. SA is a metaheuristic built in such a way that the variables of a target function are randomly modified until the function arrives at its global minimum. In general, the algorithm starts with a high energy and high temperature state of the system, and randomly moves the system to another state. If the resulting total energy is lower, then this new state is accepted and the algorithm takes the next step cooling down the system, that is, with an slightly lower temperature. The algorithm tries to find the minimum total energy of the system with the lowest temperature. The risk is the local minima, which implies larger computing time to better explore the parameter space.

Quantum annealing (QA) [8] is based on the classical simulated annealing. It is a quantum metaheuristic algorithm

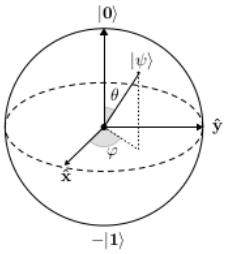

Fig. 2. The Bloch sphere. Here, the qubit $|\psi\rangle$ is not in any of the pure states $|0\rangle$ or $|1\rangle$, but in a superposition of the two.

TABLE I
THREE BASIC QUANTUM GATES, THEIR DIAGRAMS AND THEIR TRANSFORMATION MATRICES.

Name	*Circuit symbol*	*Matrix*
X - Pauli	X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Hadamard	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Controlled-NOT		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

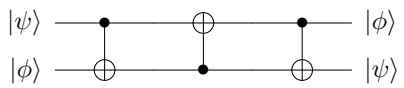

Fig. 3. An example on the use of quantum gates to swap two qubits.

for optimisation that particularly exploits the quantum property of tunnelling. In particular, the difficulties of local minima in classical simulated annealing are overcome by the quantum system tunnelling through the potential barriers without the need of returning to higher energy states. QA should find, theoretically, global minima in combinatorial optimisation problems faster than the classical simulated annealing for similar problems.

There are two main steps in QA: mapping and embedding. Mapping consists in writing the physical problem in its QUBO form, selecting the proper binary variables and constraints for the cost function. Embedding is a hardware-dependent step that links binary variables to pairs of connected qubits. Let us see these two steps further.

The QUBO formulation is a cost function, represented by an Ising Hamiltonian, with the form (e.g., [9] and references therein):

$$H_0 = -\sum_{i<j} J_{ij}\sigma_i\sigma_j - \sum_j h_j\sigma_j \qquad (2)$$

where σ_i is the binary value of the qubit i, J_{ij} is the coupling between the qubits, and h_j is the bias of each qubit j. Since not all qubits are coupled, the embedding step guarantees the connection between qubits (and so, the J_{ij} values). QA evolves adiabatically this problem Hamiltonian H_0 until it reaches its ground state, that will be the global minimum/solution.

The challenge of QA is to properly express the problem in its best QUBO form and to develop tools to help programmers to make this transformation an easy task [12].

IV. Scalability and Quantum Computing

Most of the current efforts in QC pursuit to enlarge the size of quantum computers or, from a simplistic but illustrative sight, the number of qubits available. The larger the number of qubits within a quantum computer, the higher the number of variables that a QA will handle.

The number of qubits of a quantum computer depends on the technology and architecture applied. Two flagships in the race of this scalability are the IBM [10] and D-Wave companies [11]: the first promises more than 1000-qubits at the end of 2023; the second more than 5000-qubits this year. Figure IV shows the timeline of QC in terms of the number of qubits for these two companies, which well represents the state-of-the-art of quantum computers.

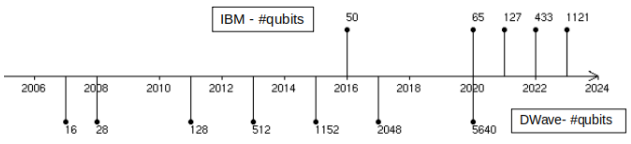

Fig. 4. Timeline of the number of qubits available in the two QC paradigms.

The increase of the number of qubits in quantum computers implies less stable machines and higher probability of errors in QC. How to control quantum errors is explained in the next Section.

V. Robustness and Quantum Computing

We refer to *robust* QC to quantum computations isolated from the environment and tolerant to hardware defects. Robust QC must guarantee that an arbitrary large quantum computation will end with success independently of the number of qubits involved and the computing time needed. There are many limitations to assure robust QC: difficulty in isolating quantum systems, impossibility of coping unknown quantum states (the no-cloning theorem), and managing the interactions between qubits at hardware level are the most prominent examples. We can divide the challenge of reaching a robust QC in two main aspects of it: quantum error correction and quantum fault tolerance. We discuss both in the rest of this Section.

A. Quantum Error Correction

Quantum error correction (QEC) is probably one of the most active fields in QC. The first QEC code was independently published by Shor [13] and Steane [14]. In general, the idea behind QCE is the use of extra qubits to "control" the ones performing the computations. This means that from the total amount of built qubits in a quantum computer, some of them must be dedicated to QEC, reducing the number of available qubits for computation. The simplest case is the transmission of a qubit that is helped with two additional ones to assure error-free transmission, that we now review mirroring [15]. We adhere to the standard convention where *Alice* transmits and *Bob* receives the transmission.

Alice wants to send a qubit to *Bob* via a noisy channel. We assume an artificial noise on the channel that randomly produces one of the two following effects on the qubit: either the state of the qubit remains unchanged, with probability $(1-p)$, or the state of the qubit changes according to the X-Pauli operator (see Table I), with probability $p < 1/2$. The state of the qubit that Alice will send can be written as $|\psi\rangle = \alpha|0\rangle+\beta|1\rangle$, and she prepares two additional qubits in the state $|0\rangle$. The state of the three initial qubits will be $|\phi\rangle = \alpha|000\rangle+\beta|110\rangle$. *Alice* applies then a C-NOT gate (see Table I) from the first qubit to the third one, producing a new state $|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$. *Alice* sends the three qubits. *Bob* receives the three qubits, but he knows the channel is noisy. There are eight possibilities of the states of the qubits, from $(\alpha|000\rangle + \beta|111\rangle)$, with probability $(1-p)^3$, to $(\alpha|111\rangle + \beta|000\rangle)$, with probability p^3. *Bob* introduces two further qubits in the state $|00\rangle$, called *ancilla* qubits, to check on the effect of the noise. Then, he applies two C-NOT gates: one from the first and second qubits to the first ancilla qubit; another from the first and third qubits to the second ancilla qubit. Now *Bob* has five qubits with again eight possibilities for the states, from $(\alpha|000\rangle + \beta|111\rangle|00\rangle)$ with probability $(1-p)^3$, to $(\alpha|111\rangle + \beta|000\rangle|00\rangle)$, with probability p^3. We summarise all those states, both from Alice and Bob, in Table II to better visualise them.

Bob measures now the ancilla qubits. Possible output values are $00, 01, 10, 11$. The information obtained by this measurement is called *error syndrome*, because it is a diagnosis of

TABLE II
QUANTUM STATES AND PROBABILITIES ON TRANSMISSION.

Alice	Bob + ancilla	probability
$(\alpha\|000\rangle + \beta\|111\rangle)$	$(\alpha\|000\rangle + \beta\|111\rangle)\|00\rangle)$	$(1-p)^3$
$(\alpha\|100\rangle + \beta\|011\rangle)$	$(\alpha\|100\rangle + \beta\|011\rangle)\|11\rangle)$	$p(1-p)^2$
$(\alpha\|010\rangle + \beta\|101\rangle)$	$(\alpha\|010\rangle + \beta\|101\rangle)\|10\rangle)$	$p(1-p)^2$
$(\alpha\|001\rangle + \beta\|110\rangle)$	$(\alpha\|001\rangle + \beta\|110\rangle)\|01\rangle)$	$p(1-p)^2$
$(\alpha\|110\rangle + \beta\|001\rangle)$	$(\alpha\|110\rangle + \beta\|001\rangle)\|01\rangle)$	$p^2(1-p)$
$(\alpha\|101\rangle + \beta\|010\rangle)$	$(\alpha\|101\rangle + \beta\|010\rangle)\|10\rangle)$	$p^2(1-p)$
$(\alpha\|011\rangle + \beta\|100\rangle)$	$(\alpha\|011\rangle + \beta\|100\rangle)\|11\rangle)$	$p^2(1-p)$
$(\alpha\|111\rangle + \beta\|000\rangle)$	$(\alpha\|111\rangle + \beta\|000\rangle)\|00\rangle)$	$(p)^3$

the error. Depending on output, *Bob* performs the following actions: if 00, he does nothing; if 01, 10 or 11, he applies a X-Pauli gate to the third, second or first qubit, respectively. Finally, *Bob* keeps the most probable state and applies a C-NOT gate to it. He obtains the original qubit sent by *Alice* with probability greater than $(1-p)$ (remember that $p < 1/2$).

This code corrects a single-qubit bit-flip error. The importance of this very simple example of qubit transmission and quantum error correction is that the probability that *Bob* fails in getting the qubit sent by Alice is $O(p^2)$, whereas without error correction it would have been $O(p)$.

The development of QEC methods and frameworks is a difficult task and is one of the most important goals to the success of a robust QC for the incoming years. An obvious way of generalising the above QEC circuit is concatenating several of them to decrease even more the probability of failing in qubit manipulation. One of the challenges in concatenating quantum correction circuits is that we increase the number of qubits and quantum gates, so errors might appear more often. We treat this in the next Subsection.

B. Quantum Fault Tolerance

There is an important question in QC that we need to think of at this point: can we perform a quantum computation during an arbitrary amount of time without being overcome by noise and quantum decoherence?

The answer to this question comes from the work by [16]. The *quantum threshold theorem*, also called *quantum fault tolerance theorem* establishes that the application of QEC codes can eliminate any logical error rate to arbitrary low levels. This theorem implies that we can indeed build fault tolerant quantum computers. The general scheme for the quantum threshold theorem is given by [17]: the idea is to apply periodically an error-correction code on encoded states, so we prevent error accumulation, and creating hierarchical fault tolerant procedures.

Consequently, noise in QC is no longer an intractable problem. The challenge is to create proper QEC codes and design strategies to decrease the number of additional quantum circuits to increase fault tolerance, so that only a small fraction of all the qubits are intended to fault tolerance.

VI. CURRENT ACCESS TO QUANTUM COMPUTERS

We briefly describe in this section some of the companies in the market and universities that offer access to quantum computers.

- *Amazon Braket - Amazon*. A development environment to explore and design QAs, test them on a simulated quantum computer, and run them on your choice of different quantum hardware technologies.
- *D-Wave SDK - D-Wave Systems* It is a suite of open-source *Python* tools. Unlike the others, D-Wave machines are quantum annealers. The company offers a quantum cloud service.
- *Forest SDK - Rigetti Computing*. Based on the *Python* library *pyQuil*, with a Quantum Virtual Machine (QVM) and a compiler (*quilc*). Offers cloud access to their 32-qubits *Aspen-9* quantum computer.
- *Forge - QC Ware*. Access to D-Wave hardware as well as Google and IBM simulators. The platform offers a 30-day free trial, with 1-minute of quantum computing time.
- *IBM Q Experience - IBM*. Uses the *Qiskit* language and gives access to several quantum processors, ranging from 27 to 65 qubits.
- *LIQUi|⟩ - Microsoft*. It is a quantum computing simulation platform. The succesor to this platform is *Q♯*.
- *Quantum in the Cloud - The University of Bristol (UK)*. A free, open web interface to a 4-qubit optical quantum photonic chip with *simulation* and *experiment*.
- *Quantum in the Cloud - Tsinghua University (China)*. It is a 4-qubit quantum chip based on nuclear magnetic resonance.
- *Quantum Inspire - Qutech*. It is a cloud-based quantum computing with access to two hardware chips: a 5-qubit transmon (a type of superconducting charge qubit) quantum processor and a 2-qubit electron spin quantum processor.
- *Quantum Playground - Google*. A simulator with a simple interface, and a scripting language and 3D quantum state visualization.
- *Xanadu Quantum Cloud - Xanadu*. It is a cloud-based access to three fully programmable photonic quantum computers, with 8-qubits, 12-qubits and 24-qubits. Uses the programming suite called *Strawberry Fields*.

VII. FUTURE RESEARCH IN QC PARADIGMS

Most of the steps in the two QC paradigms moves towards larger quantum machines in terms of the number of qubits. Main companies point towards this goal: IBM promises 1000-qubit computers for 2023 and is already preparing a 1 million qubit machine for the near future; D-Wave will offer at the end of this year a 5000-qubit quantum annealer. The byproduct of this scalability in the number of qubits in both QC paradigms is clear: there is an active research in the physical construction of the qubits, more compact architectures, lower noise levels and more efficient QEC circuits.

However, quantum software development does not seem to scale as fast as quantum hardware. Nowadays there is a gap

between quantum programmers and quantum hardware, and clearly quantum software needs a boost in many aspects. High level quantum languages as transparent to the programmer as they can, testing and debugging tools and standard ways of physical problems transcription into quantum formulations are needed.

An obvious open question arises now: could we use quantum computers to design and build better quantum computers? According to the nature of the problem, the answer should be yes, but to our best knowledge we do not know plans in this direction.

VIII. Other Open Issues: Industrial Challenges

QC might play a fundamental role in Industry, bringing innovative solutions to practical industrial applications. We briefly point some of the ones with more societal impact:

- *Automotive*. The industry of the automobile is moving fast towards electrical and hybrid technology. Batteries suffer from low autonomy and slow recharge. New batteries with higher energy density and small weight. QC offers higher computational power to investigate new chemical reactions and materials in this field [18].
- *Health – Drug discovery*. The formulation of candidates to medical compounds is very expensive in the Pharma industry. Simulating the behaviour of the proper molecules in compounds can save large amounts of money and make medicaments cheaper [19].
- *Automated planning & scheduling*. QA computers are specifically design for this type of optimisation problems, where QC seems to have a huge potential [18]
- *Quantum Artificial Intelligence*. Major challenges are: replace training by better QA, large datasets and QC, and standardised interfaces [20].

A very general SWOT analysis of the present status of QC is presented in Figure 5.

STRENGTHS	WEAKNESSES
• Theoretical framework • Quantum supremacy • Robustness of QC	• Complexity of QAs • Low number of qubits • Quantum software
OPPORTUNITIES	**THREATS**
• Large public investment • Main computer companies involved • Many industrial sectors interested	• Prize of services • Classical supercomputing easier to access • Difficult access to non experts

Fig. 5. A SWOT analysis on quantum computing.

IX. Conclusions

QC remains a challenge in itself. Much of the efforts in QC are targeting the hardware of quantum computers, rather than designing new QAs and quantum software to assist quantum programmers. The two QC paradigms described here point towards two different hardware architectures: QA designed for specific families of optimisation problems and quantum gate computers conceived as general-purpose quantum computers for a wider community of users. We can summarize the close future challenges of the two QC paradigms in the following items:

- Scalability with regards quantum decoherence, quantum errors and quantum fault tolerance to keep and improve QC robustness.
- Software testing and debugging tools for quantum programmers, with quantum platforms/interfaces for transparent use of quantum machines.
- Exhaustive research in new QAs for quantum gate computers and QUBO formulations for quantum annealers.

Acknowledgments

This research is partially funded by the Universidad de Málaga (DataPol UMA-CEIATECH-07), Consejería de Economía y Conocimiento de la Junta de Andaluía and FEDER under grant number UMA18-FEDERJA-003 (PRECOG); TAILOR ICT-48 Network (No 952215) funded by EU Horizon 2020 research and innovation programme.

References

[1] P.W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring”, 35th Annual Symposium on Foundations of Computer Science Proceedings, Santa Fe, USA, 1994
[2] L. Grover, “A fast quantum mechanical algorithm for database search, ArXiv quant-ph/9605043, 1996
[3] D.T. Gillespie, “A Quantum Mechanics Primer”, International Textbook Company, 1970
[4] A. Einstein, B. Podolsky, N. Rosen, “Can Quantum-Mechanical Description of Physical Reality Be Considered Complete”, Phys. Rev. 47, pp. 777, 1935
[5] D.P. DiVincenzo, “Quantum Computation”, Science 270, 255, 1995
[6] E. Knill, “Quantum Computing”, Nature, Vol. 463, 2010
[7] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, “Optimization by simulated annealing”, Science, Vol.220, pp.671, 1983
[8] B. Apolloni, N. Cesa-Bianchi, D. De Falco, “A numerical implementation of quantum annealing”, in Stochastic Processes, Physics and Geometry, Proceedings of the Ascona-Locarno Conference, July, 1988
[9] E. Cohen, B. Tamir, “Quantum annealing – foundations and frontiers”, Eur. Phys. J. Spec. Top. 224, pp.89110, 2015
[10] IBM website, https://quantum-computing.ibm.com/. Accessed on: June 2021
[11] D-Wave Systems website: https://quantum-computing.ibm.com/. Accessed on: June 2021
[12] E. Boros and A. Gruber, “On Quadratization of Pseudo-Boolean Functions”, International Symposium on AI and Mathematics, 2012
[13] P.W. Shor, “Scheme for reducing decoherence in quantum computer memory’ ’, Phys. Rev. A, 52, 1995
[14] A.M. Steane, “Error Correcting Codes in Quantum Theory”, Phys. Rev. Lett. 77, 793, 1996
[15] A.M Steane, “Tutorial on Quantum Error Correction”, in “Quantum Computers,Algorithms and Chaos”, Proceedings of the International School of Physics Enrico Fermi, pp.132, IOS Press, 2006
[16] D. Aharonov, M. Ben-Or, “Fault-tolerant quantum computation with constant error”, Proceedings STOC'97, pp. 176-188, 1997
[17] M.A. Nielsen, I.L. Chuang, “Quantum Computation and Quantum Information”, Cambridge University Press, 2010
[18] A. Luckow, J. Klepsch, J. Pichlmeier, “Quantum Computing: Towards Industry Reference Problems”, Digitale Welt, vol.5, num.2, 2021
[19] Y. Cao, J. Romero, A. Aspuru-Guzik, “Potential of quantum computing for drug discovery”, IBM Journal of Research and Development, vol. 62, no. 6, 2018
[20] T. Gabor, L. Sünkel, F. Ritz et al., arXiv:quant-ph/2004.14035, 2004