

Towards Distributed Quantum Algorithms

Pablo Andres-Martinez



Master of Science by Research
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
2018

Abstract

Acknowledgements

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Pablo Andres-Martinez)

Table of Contents

1	Quantum Computing: A brief overview	1
1.1	The principles of Quantum Computing	2
1.2	Building Quantum Computers	4
1.2.1	Scalability challenges	5
1.2.2	Models of computation	6
1.3	Programming on Quantum Computers	8
1.4	Summary	8
2	Distributed Quantum Computing	9
2.1	Architecture	9
3	Towards Distributed Quantum Algorithms	10
3.0.1	Non-local operations	10

Chapter 1

Quantum Computing: A brief overview

Quantum computing aims to take advantage of quantum mechanics to speed-up computations. There is strong theoretical evidence that quantum computers are capable of solving some problems substantially faster than standard (classical) computers. Well known examples are:

- Shor's algorithm for *polynomial-time* large number factorisation ?. There is no known algorithm on classical computers that can perform this task in polynomial time, and it is suspected not to be possible¹. Quantum computers theoretically provide an exponential speed-up on this task.
- Grover's algorithm for efficient unstructured search ?. The algorithm performs a brute-force search (i.e. requiring no knowledge about the search space) over a N data-points in time proportional to \sqrt{N} . A brute-force search in a classical computer should always take time proportional to N .

Besides, in May of the present year, Raz and Tal ? gave formal proof of the existence of problems that a classical computer may never solve in polynomial time, but are solvable in polynomial time on a quantum one. Nevertheless, all of these results are theoretical in nature and there are some caveats on their practical implementation, discussed in § 1.2.1. Giving experimental evidence of this time-efficiency separation between quantum and classical computers is a highly active area of research, known as *quantum supremacy* ?. The general opinion of the scientific community is that we will eventually overcome these caveats, and take advantage of quantum computing in fields of science such as:

¹RSA, a widely used encryption system, relies its security on the assumption that factorisation of large numbers can not be computed efficiently.

- Chemistry: **TODO**
- Machine learning: **TODO**
- Engineering: **TODO**

For these applications, we will require large scale quantum computers. Due to the obstacles in the way of building a large mainframe quantum computer (see § 1.2.1), some authors have advocated the alternative of building a grid of smaller quantum computer units that cooperate in performing an overall computation ?. In this work, particularly in Chapter 3, we contribute to this perspective, providing a method for efficiently distributing any quantum program originally designed for a monolithic quantum computer.

1.1 The principles of Quantum Computing

The advantage of using quantum mechanics to perform computations is usually traced down to the following three principles:

- *Superposition*: In classical computing, the unit of information is the *bit*, which may take one of two values: 0 or 1. In quantum computing, the *bit*'s counterpart is the *qubit*, whose value may be *any linear combination* of 0 and 1, known as a *superposition*, and usually written as:

$$|qubit\rangle = \alpha|0\rangle + \beta|1\rangle$$

where α and β are complex numbers that must satisfy: $\alpha^2 + \beta^2 = 1$.

A popular analogy of a qubit's superposition is a coin spinning²: the classical states (0 and 1) are *heads* and *tails*, but when the coin is spinning, its state is neither of them. If we knew exactly how the coin was set spinning, we would be able to describe the probability distribution of seeing heads or tails when it stops; these would be our α^2 and β^2 values. We may *measure* a qubit, and doing so corresponds in our analogy to stopping the coin: we will get either 0 or 1 as outcome.

²Note this is just an analogy, and while a coin spinning can be perfectly modelled using classical physics, a qubit can not. In fact, superposition is key in the (even weirder) two other principles that differentiate quantum and classical computing.

The essential aspect of this analogy is that, before measurement, the *qubit*'s state is neither 0 nor 1. Through certain operations (that would correspond to altering the axis of spin of the coin), we may change the coefficients α and β of the superposition. Interestingly, in quantum computing, we encode input and read output (after measurement) as standard classical binary strings, and thus, *for input/output we use as many qubits as bits would be required*. What superposition provides is the ability to – during mid-computation – maintain a superposition of all potential solutions to the problem, and update all of them simultaneously with a single operation to the qubits. In some sense, superposition allows us to explore multiple choices/paths of the computation, using only the resources required to explore a single one of those paths.

- *Interference*: Superposition gives us the ability to simultaneously explore different paths to solve a problem. However, in the end we will need to measure the qubits – stop the coins, in order to read heads or tails – and the result will be intrinsically random. For quantum computing to be any better than a probabilistic classical computer, we require the ability to prune the paths that have led to a dead-end. This is precisely what *interference* provides: some operations on the qubits may make different classical states in the superposition cancel each other out. Interference is at the core of any speed-up achieved by a quantum algorithm, and taking advantage of it is the main challenge when designing quantum algorithms.
- *Entanglement*: Quantum mechanics allows us to have a pair of qubits a and b in a superposition such as:

$$|a,b\rangle = \frac{1}{\sqrt{2}}|0,0\rangle + \frac{1}{\sqrt{2}}|1,1\rangle$$

This implies that, when we measure the qubits, we may only read $a = 0, b = 0$ or $a = 1, b = 1$ as outcome (the coefficients for $|0,1\rangle$ and $|1,0\rangle$ are both 0). Then, what happens if we only measure a ? In this case, we would also know b 's outcome, without measuring it. More surprisingly, if we measured both a and b at the same instant, we would obtain $a = 0, b = 0$ half of the times and $a = 1, b = 1$ the other half. In short, it seems like acting on one qubit has an instantaneous effect on the other. Whenever a group of qubits exhibits this property, we say they are *entangled*. Entanglement holds regardless how far apart a is from b ; for instance, they could be on two different quantum processing units of a distributed

grid. Indeed, entanglement will be key in our discussion of distributed quantum algorithms, and we explain how to use it to perform non-local operations in § 3.0.1.

1.2 Building Quantum Computers

Physicists have come up with different ways of realising qubits in labs. The key idea is to find a physical system that displays non-classical behaviour, and put it under the appropriate circumstances so we can manage its quantum properties, but noise in the environment may not interfere with these. Examples of such successful technologies are:

- *Photonics*: The state of a qubit is represented in the properties of photons, for instance, their polarization ?. A great advantage of this technology is that photons can be easily sent over long distances, while preserving the quantum state. Thus, protocols in quantum information that heavily rely on communication, such as Quantum Key Distribution ?, are usually discussed and experimented with using photonics. The downside of photonics is that it is inadequate for building general-purpose quantum computers, as the technology is not scalable.
- *Ion-traps*: Each qubit is embodied as an ion, confined inside a chamber by means of an electric or magnetic fields. The qubit is acted upon by hitting the ion with electromagnetic pulses (e.g. laser light or microwave radiation). This is one of the most promising technologies for the future of quantum computing, with groups of experimentalists having proposed how to scale up the technology ?.
- *Superconductors*: Small circuits, similar to classical electrical circuits, are cooled down to near absolute zero so the quantum interactions of electrons are not obscured by other perturbations. Then, different parts of the circuit encode different qubits, which can be acted upon by applying different electric potentials. One of the main advantages of this technology is that the challenges for scaling up such circuits, apart from the cooling system, are similar to the challenges we have encountered over the years for classical computers. Therefore, this technology seems to be the most feasible in the near future and evidence of that is the fact that, using it, both IBM ? and Intel? have already built small generic-purpose quantum computers of 17-20 qubits.

However, for quantum computers to be useful in real world applications, their qubit count should raise up, at the very least, one order of magnitude. And, unfortunately, increasing the amount of qubits in a quantum computer is particularly difficult, due to some caveats we will now discuss.

1.2.1 Scalability challenges

There are two main challenges to overcome in order to build large scale quantum computers:

- *Decoherence*: In § 1.1 we discussed the importance of having superposition in quantum computing, and we compared a qubit in superposition with a coin spinning. For the same reason why a coin spinning will eventually stop, a qubit in superposition will eventually degenerate into a classical state (i.e. either $|0\rangle$ or $|1\rangle$): physical systems have a tendency towards the state at which they are most stable, for the coin it is laying flat, for the qubit it is losing its superposition. This phenomenon is known as *decoherence* and it will always occur in any given technology³, in some faster than others. Experimentalists attempt to increase the time it takes for the state of the qubit to degenerate, which in both ion-trap and superconductor technologies it is in the order of microseconds. There has been an steady improvement of this decoherence time across the years ?, and it is expected to keep improving. Nevertheless, it remains the main constraint to scalability of quantum computers, as it dictates the lifespan of qubits, limiting the number of operations that can be applied in a single program.

Certainly, the state of bits also degenerates in classical computers. However, in their case this is easier to account for: intuitively, we can keep monitoring the bits, and make sure to correct any unwanted change. This is not so simple in quantum computers, as monitoring a qubit would require *measuring* it, and that destroys any quantum superposition. Nevertheless, it is still to some extent possible to protect our quantum state from errors – either due to decoherence or imperfect hardware – through quantum error correction routines ?. This is a very active area of research, and it will be key for the implementation of reliable large scale quantum computers.

³Some experts have discussed that encoding qubits in *Majorana fermions* would avoid decoherence ?. Although promising, currently this proposal has little experimental underpinning, and it is not regarded as attainable in the near future.

- *Connectivity*: In order to run complex computations on qubits, we will need to be able to apply multi-qubit operations on any subset of the available qubits. However, it is not realistic to expect that quantum computers will have fast connectivity between all qubits, due to spatial separation of these in the hardware. In classical systems, this problematic is solved by a memory hierarchy, with a ceaseless flow of data going up and down of it, from main memory to registers and back. However, the memory hierarchy model works because most of the data can stay idly in main memory while computation on the registers data is carried out. In quantum computers, we must avoid qubits being idle, as decoherence prevents the existence long-lasting memory. An alternative found in classical computers is to distribute the computation across different processing units, each having its own local memory which they use intensively, and communicating – through message passing – as little as possible. In § 2.1, we discuss an abstract distributed quantum architecture in detail.

1.2.2 Models of computation

In this section, we give a brief introduction to some models of quantum computation relevant to the this thesis.

- *Circuit model*: Also known as the network model. Any operation on n qubits – as long as measurement (i.e. destruction of information) is not involved – can be represented as square matrix on complex numbers, of dimension 2^n . These matrices are always unitary, which means that a matrix U satisfies $UU^\dagger = I = U^\dagger U$, where I is the identity matrix and A^\dagger is the conjugate transpose of A . Essentially, unitarity ensures that any operation on qubits can be reversed (i.e. undone), reason why this model is sometimes called the reversible model. Multiplying matrices AB corresponds to applying the operation described by B first, then A , on the same qubits. Application of two operations on disjoint set of qubits corresponds to the Kronecker product of the matrices $A \otimes B$. The fundamental concept is that any matrix can be represented as a product of other matrices, so we may decompose any operation into smaller building blocks: quantum gates.

Qubits are pictured as wires to which quantum gates are applied, similarly to a classical digital circuit. The set of quantum gates used is dependent on the architecture. There exist an (uncountable) infinite amount of different quantum operations, but a small finite set of them is enough to approximate any of them,

up to a desired error factor. The most common choice of such a universal gate-set is Clifford+T, which contains six one-qubit gates, and a single two-qubit gate. The depiction of such gates and some of their most important properties are shown in Figures **TODO**.

TODO: A figure depicting, and giving the matrix of, each of CNOT, X, Y, Z, H, S and T. Figures showing $HXH = Z$; $XX = I$, $ZZ = I$, $YY = I$, $HH = I$; $SS = Z$; $TT = S$; $H^2 \text{ CNOT } H^2 = \text{NOTC}$, and also their algebraic notation.

TODO: A figure showing an abstract quantum circuit.

- *MBQC model:* Initials stand for Measurement Based Quantum Computing. Unlike the circuit model, where measurements are done at the very end of the circuit, MBQC carries out computations by means of repeatedly measuring an initial entangled resource. The process, sketched in Figure ?? can be thought of as sculpting a statue from a block of granite. The initial entangled resource, which is a collection of entangled qubits forming a lattice structure, corresponds to the granite block. By measuring some qubits in the lattice – hitting the rock with a chisel – we remove some of the excess qubits, changing the overall state in the process. The outcome of measurements is probabilistic so, in order to provide deterministic computation, we must apply corrections on the neighbouring qubits whenever the measurement outcome deviated from the desired result. After multiple iterations of measurements and corrections, we end up with a set of qubits encoding the result. The input was incorporated into the lattice at the beginning of the process.

TODO: Figure from slides

In this way, any computation may be performed by applying 1-qubit measurement and 1-qubit correcting gates (controlled by classical signals). The initial resource state contains all the entanglement that is required, which may be prepared experimentally through multi-qubit interactions, such as Ising interactions ?, which are within our experimental capabilities. Hence, in some sense this model solves the problem of connectivity by applying a single large operation at the beginning of the process, and then only requiring cheap single qubit operations. The main drawback of MBQC is the large amount of qubits that are required for even the simplest of operations, but given this might be surmountable considering the rest of the architecture is greatly simplified. The MBQC model was presented for the first time by Raussendorf and Briegel ? under the

name of *one-way quantum computer*, highlighting its main difference with the circuit (reversible) approach.

- *Distributed model*: We may find a balance between the circuit model and MBQC, where multiple small quantum processing units run fragments of the overall circuit, and communication is achieved through a shared entangled resource. This model has been discussed in detail in the literature ? and it is at the core of the Networked Quantum Information Technologies (NQIT) project ?. In § 2.1, we discuss an abstract distributed quantum architecture in detail.

1.3 Programming on Quantum Computers

Very low level and fixed on circuit model!

1.4 Summary

TODO

Here are the key concepts to keep while reading the rest of this thesis:

Quantum computers are great Small quantum computers are available Challenges to scalability: decoherence (potentially solved by engineering and error correcting) and connectivity (potentially solved by distributed architectures) No programming support for distributed architectures

Chapter 2

Distributed Quantum Computing

2.1 Architecture

Chapter 3

Towards Distributed Quantum Algorithms

3.0.1 Non-local operations