Quantum Computation and Information

A Gentle Introduction

Pedro Rivero

Argonne National Laboratory
Illinois Institute of Technology
`priveroramirez@anl.gov`

October 21, 2019

Contents Overview

What is Quantum Computing?

Quantum Computing is the application of **Quantum Information Science (QIS)** to the development of *machines* capable of performing operations and calculations based on quantum logic instead of the usual classical logic.

The usefulness of this kind of computation lays not only on the ability to engineer exponentially faster machines, but also on things such as being able to **efficiently simulate nature at the quantum level**, or creating encryption systems which are fundamentally secure.

CLASSICAL LOGIC

Set Theory (Boolean algebra)

- **AND** $\Rightarrow$ $A \cup B$
- **OR** $\Rightarrow$ $A \cap B$
- **NOT** $\Rightarrow$ $\overline{A}$
- **XOR** $\Rightarrow$ $A \cap B - A \cup B$

QUANTUM LOGIC

Quantum Theory (Non-Commutative)

- **Probabilistic measurement**
- **Measurement causes disturbance**
- **Superposition**
- **Entanglement**
- **Uncertainty principle**

The Abacus Effect

To try to understand the power of Quantum Computing in abstract, we can resort to what might be called **"The Abacus Effect"**.

Before the abacus was invented, the only way of counting was through a "thermometer-like" scale, which was highly inefficient as it meant adding one physical bead per unit. The abacus did something great: it introduced the concept of **digits**, which was later put into an even more concise notation by the Arabs through the introduction of **numerals**.

This means that, instead of counting up to N using N raw elements, we can count using only $\log_b(N)$ abacus elements —where b is the base of our number system (i.e. the amount of beads per row in the abacus). Therefore, for $b \geq 2$, the abacus introduced an **exponential decrease in resource requirements** to represent the exact same thing.

Quantum computers do something analogous. To represent the quantum state of a system made out of N subsystems —each of which with b degrees of freedom— we would usually need b^N classical elements (i.e. numbers). By using quantum systems for this representation, we would only need N quantum elements (i.e. each subsystem). Again, an exponential decrease in resource requirements.

Quantum Software: IBM's Qiskit

Working with quantum systems is extremely difficult, and in many cases very expensive, this makes quantum computing research unavailable to many groups and institutions. In order to solve this problem, IBM has developed **Qiskit**, a python framework for quantum computation and quantum information.

This framework allows the construction of quantum circuits —today's most common way of representing quantum computation— for running using different backends. These backends range from simulators to actual quantum devices accessible on the cloud through the IBMQ API included in Qiskit. It also includes tools for handling noise and error correction, developing quantum based applications and tools, or making use of well known, pre-programmed algorithms. It also allows work through different levels of abstraction, from microwave pulses on hardware to high level algorithm implementation.

Case Study

As a starting point we decided to reproduce the paper **Cloud Quantum Computing of an Atomic Nucleus** (arXiv:1801.03897) by Dumitrescu *et al.*

In this paper, IBM's QX5 quantum computer was used to simulate the **deuteron binding energy**, through a Hamiltonian from pionless effective field theory:

$$H_N = \sum_{n,n'=0}^{N-1} \langle n' | (T+V) | n \rangle \, a^\dagger_{n'} a_n$$

Where N is the basis dimension for a discrete variable representation in the harmonic oscillator basis.

Qubit Mapping

In order to simulate a quantum system on a quantum device, we need to map its degrees of freedom onto the qubits of the device. To do so, in this case, we can use the **Jordan-Wigner transformations** to map the creation/annihilation operators onto Pauli matrices, which can be easily studied in a quantum circuit.

$$a_n^\dagger \rightarrow \frac{1}{2}\left[\prod_{j=0}^{n-1} -Z_j\right](X_n - iY_n) \qquad a_n \rightarrow \frac{1}{2}\left[\prod_{j=0}^{n-1} -Z_j\right](X_n + iY_n)$$

Where a spin up $|\uparrow\rangle$ on qubit n represents zero deuteron in state $|n\rangle$, and a spin down $|\downarrow\rangle$ one deuteron. For different values of N, this returns:

$$\begin{aligned} H_1 &= 0.218291(Z_0 - I) \\ H_2 &= 5.906709I + 0.218291Z_0 - 6.125Z_1 - 2.143304(X_0X_1 + Y_0Y_1) \\ H_3 &= H_2 + 9.625(I - Z_2) - 3.913119(X_1X_2 + Y_1Y_2) \\ &\dots \end{aligned}$$

References

"The only thing demonstrated by an impossibility proof is a lack of imagination."
– John Stewart Bell –

Argonne

NATIONAL LABORATORY