Profiling of Quantum Computers and Simulators

Filippo Posta
Mathematics Division
l*Estrella Mountain Community College*
Avondale, USA
filippo.posta@estrellmountain.edu

Professor Andreas Spanias
SenSIP Center
Arizona State University
Tempe, AZ, USA
spanias@asu.edu

Glen Uehara
SenSIP Center
Arizona State University
Tempe, AZ, USA
guehara@asu.edu

***Abstract*— Quantum Computers (QC) are a reality and evolving to fulfill their promise of revolutionizing the field of computer science. Algorithms and protocols are being developed to prepare for the day when Quantum Computers will be reliable enough to perform the calculations they are designed for. Shor's algorithm is one of these calculations and it requires the computation of Fast Fourier Transforms that may take many years on a classical computer. This work uses Quantum Fourier Transform programs to profile available Quantum Computers and simulators in an effort provide a snapshot of the current capabilities of QC.**

Keywords—quantum computers, system profiling, Braket, Qiskit, FFT

I. Project Description

Quantum Computers (QC) are a developing computational platform based on the principles of quantum physics. Their fundamental units are called Qubits. A Qubit can be on sate 0, or 1, or a linear combination of the two. This property is called *superposition* and provides a superficial reason of why QCs represent the next step in the evolution of computers [1]. The QC architecture has been constantly developed since the first (2 Qubit) quantum computer was built in 1998 and many companies have created QCs and made them available to the public via cloud systems [2,3]. In addition to QCs, many tech companies have developed QC environment simulators (QS) to provide programmers with the tools to explore the computational possibilities of QC's on classical computers [2]. The variety of cloud-based and simulator-based QC environments has created the need to profile them so that users can make educated decision of what system to use for their needs.

We aim to profile a few of the publicly available QC and QS and to do so we decided to use the Quantum Fourier Transform (QFT) and its inverse (IQFT) [4]. This choice is rooted on two factors. QCs are supposed to revolutionize cryptography by rendering the RSA Algorithm obsolete through the quick implementation of the Shor's Algorithm [5]. This quick implementation is currently unfeasible in classical computers and the bottleneck consists of the period finding step of Shor's algorithm which involves calculating Fourier transforms. The other fact is the authors' interest in machine learning and signal processing two fields that have widespread use of Fourier transforms [6].

The way that Quantum Programming (QP) works is to use a classical computer to prepare data, then feed the data to a quantum circuit and finally process the result on the classical computer. Most computational environments use Python for the classical computing steps, but quantum circuits are created differently. Figure 1 shows a QFT circuit for the Qiskit platform by IBM [7].

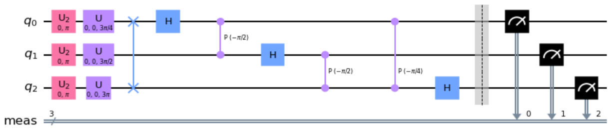

Figure1: 3 Qubits QFT circuit.

The circuits are then assembled and run in either a real cloud-based hardware or a simulator. Figure 2 shows the results of calculating QFT, its inverse and a combination of the two for the IBM Qiskit platform.

Figure 2: Performance of various circuits as a function of number of qubits and environments.

From our analysis we observe that the real-hardware QCs are still very unreliable when compared to their simulators. Both systems show a well-known issue of decreased precision as the number of Qubits increases.

References

[1] Resch, Salonik, and Ulya R. Karpuzcu. "Quantum computing: An overview across the system stack." *arXiv preprint arXiv:1905.07240* (2019).

[2] J. Hidary, Quantum Computing: An Applied Approach. Champ: Springer, 2019.

[3] Devitt, Simon J. "Performing quantum computing experiments in the cloud." *Physical Review A* 94.3 (2016): 032329.

[4] Weinstein, Yaakov S., et al. "Implementation of the quantum Fourier transform." *Physical review letters* 86.9 (2001): 1889.

[5] Terhal, Barbara M. "Quantum supremacy, here we come." *Nature Physics* 14.6 (2018):530-531.

[6] P. Wittek. Quantum Machine Learning: What Quantum Computing Means to Data Mining. Academic Press, 2014.

[7] Qiskit Development Team "Learn quantum Computing Using Qiskit". URL: https://qiskit.org/textbook/preface.html (last accessed June, 2021).

RET Site sponsored by NSF Award 1953745