Progress Report II

Dylan Miracle

ICS 698-02

Spring 2021

Feb 17, 2021

Dr. Jigang Liu

Survey of benchmarking quantum computers: annotated bibliography

Dylan Miracle
Department of Computer Science
Metropolitan State University
St. Paul, Minnesota, USA
dylan.miracle@my.metrostate.edu

***Abstract*—Quantum computing has a rich theoretical background but practical quantum computers have only recently become available to researchers and business users. These computers are still rare and operated by a small number of hardware vendors. Implementing useful algorithms on existing quantum hardware will require understanding how well quantum computers replicate their theoretical ideal. Measures of fidelity to theory have been created for specific hardware and frameworks have been developed to classify different types of benchmarks. We seek to understand these benchmarking methods and how they may be implemented using different programming languages available for programming quantum computers.**

***Index Terms*—quantum computing, quantum engineering, quantum benchmark**

I. Introduction

Quantum computing has fundamental implications for computation. Researchers are continuing to expand the understanding of what quantum computers can do while hardware vendors build more powerful quantum processors. Users of quantum computers need a theoretical understanding of the algorithms they will use on quantum computers, but also need to understand that quantum computers are not perfect facsimiles of their theoretical counterparts. In classical computing when we look to evaluate how well hardware implements a specific algorithm, we look to different types of benchmarks. Some benchmarks directly measure FLOPs while others try to create a program that will execute a series of operations that the benchmark developer believes are a good representation of a typical business application. In quantum computing there are many areas that can be benchmarked. In this study we look to several benchmarking studies on real quantum hardware as well as general methods that we can use to provide users a characterization of various hardware. This information will allow users to choose the best quantum hardware for their specific application. In addition it gives hardware builders a standard rubric they can use to describe the capabilities of their quantum processors.

II. Review of quantum computing theory and quantum circuits

Quantum computation takes several forms including quantum annealers and gate model quantum processors. In this paper we will work with the gate model of quantum computing as it has the most available hardware and is the most broadly studied. The theoretical underpinnings of gate model quantum computing is developed by first considering operations on single qubits, then operations on multiple qubits. At the most fundamental level the tools we will develop for working with qubits involve gates that allow us to put qubits into superpositions and gates that allow us to entangle multiple qubits. This section will develop the operations that are used for superposition and entanglement.

Quantum computing is currently at the gate level – when classical computing was at the level of ands, ors, nots etc. This thesis is a comprehensive overview of all the quantum operations that could be available to a quantum computer.

The Nielsen and Chuang textbook [2] is the canonical reference on quantum information. This will be our resource for understanding the basic gate mode of quantum computing as well as the linear algebra used to define each specific quantum gate.

References

[1] Michael S. Saraivanov, Quantum Circuit Synthesis using Group Decomposition and Hilbert Spaces, MS, Portland State University, 2013.
[2] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, 10th Anniversary ed., Cambridge, 2016, pp. 171–215.

III. Benchmarking digital computers

In order to understand benchmarks for quantum computers, it is good to review how we use benchmarks in digital computing. There are several important considerations when we create benchmarks for digital computers. Certain benchmarks are only concerned with a single operation, for example floating point multiplication, and a benchmark for this would do the same operation or a limited set of operations (multiplication, addition, division, and subtraction) over and over again, then time how long it took to perform these operations. Another strategy employed for benchmarking is to create a program that has a cross section of operations that are similar to the operations used in some business logic, then time how long it would take a system to perform this operation. There are several such benchmarks tailored to different conditions.

As we study benchmarks it is important to consider how we will evaluate the statistics we gather. Flemming and Wallace [3] argue that taking an arithmetic mean of data obtained in benchmarking studies leads to systematic misunderstanding of

data. They argue that a geometric mean is a more accurate way to summarize benchmarking data.

New computing paradigms require different benchmarking methods. Dai and Berleant [4] cover a range of benchmarking methods for new hardware specifically designed for deep learning. This study is important for my work because it shows how to adapt a mature field, benchmarking digital computers, for a more cutting edge application.

REFERENCES

[3] Philip J. Fleming and John J. Wallace. 1986. How not to lie with statistics: the correct way to summarize benchmark results. Commun. ACM 29, 3 (March 1986), 218–221. DOI:https://doi.org/10.1145/5666.5673.

[4] W. Dai and D. Berleant, "Benchmarking Contemporary Deep Learning Hardware and Frameworks: A Survey of Qualitative Metrics," 2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI), Los Angeles, CA, USA, 2019, pp. 148-155, doi: 10.1109/CogMI48466.2019.00029.

IV. QASM AND PYTHON FRAMEWORKS FOR PROGRAMMING QUANTUM PROCESSORS

This is the most referenced paper on quantum assembly – a language that gives us access to the basic quantum gates. It looks like C and will be how we program some benchmarks, or at least the basis of a framework we use to program our benchmarks.

Authors of this paper developed a quantum assembly language that takes the mathematics of quantum circuits and creates a formalized interface language. The language can be compiled and simulated as well as run on certain hardware. The language gives a uniform representation of quantum circuits using gates. This paper goes into depth implementing several famous algorithms using QASM. Unfortunately QASM lacks some very basic logic that we like as programs such as loops and conditionals.

Qiskit [10] is a framework for quantum computing developed at IBM and implemented as a python library. Authors have created a book documenting how to program a quantum processor using the qiskit language. The textbook offers a way to learn the fundamentals of quantum computation while learning the qiskit framework. The authors are dedicated to implementation on a particular type of quantum computer, namely a gate model quantum processor that can evaluate quantum circuits.

REFERENCES

[5] Andrew W. Cross and Lev S. Bishop and John A. Smolin and Jay M. Gambetta, "Open Quantum Assembly Language," arXiv:quant-ph/1707.03429, 2017.

[6] Abraham Asfaw and Luciano Bello and Yael Ben-Haim and Mehdi Bozzo-Rey and Sergey Bravyi and Nicholas Bronn and Lauren Capelluto and Almudena Carrera Vazquez and Jack Ceroni and Richard Chen and Albert Frisch and Jay Gambetta and Shelly Garion and Leron Gil and Salvador De La Puente Gonzalez and Francis Harkins and Takashi Imamichi and Hwajung Kang and Amir h. Karamlou and Robert Loredo and David McKay and Antonio Mezzacapo and Zlatko Minev and Ramis Movassagh and Giacomo Nannicni and Paul Nation and Anna Phan and Marco Pistoia and Arthur Rattew and Joachim Schaefer and Javad Shabani and John Smolin and John Stenger and Kristan Temme and Madeleine Tod and Stephen Wood and James Wootton., Learn Quantum Computation Using Qiskit, 2020, http://community.qiskit.org/textbook, University of Waterloo, 2008.

V. QUANTUM BENCHMARKING THEORY

Development of quantum benchmarks relies on understanding how a real quantum computer may deviate from a theoretical quantum computer. As we study all the ways our real computer can deviate, we are able to theorize measurements we can perform to expose how well or poorly specific operations mimic what we expect from theory. In this section we can tie our new exploration of quantum computing to the methods of benchmarking we explored in the previous section on benchmarking digital computers. For example, we can choose specific quantum operations and perform them repeatedly and time how long it takes for the computer to complete the operation. We could get a more general benchmark by devising a quantum circuit with a predictable outcome that uses many different operations and repeated fun that circuit to get a different benchmark. Both of these types of benchmarks have classical analogs, however another class of benchmarks exist for a quantum computer that have no classical analog. These include measuring if a previous calculation impacts the outcome of the next calculation, or studying how long entangled qubits remain entangled.

Authors [8] create a map of benchmarking methods that classify protocols on two axis: from weak assumptions to strong assumptions and from less information gain to greater information gain. This study gives us a framework for analyzing benchmarking methods for quantum computers. They provide an outline certification methods that can be used to classify quantum computers. One certification described in this review concerns Measures of Quality. These methods will give concrete numerical measures of performance based on how well a quantum computer creates correct output. This corresponds to the classical notional of correctness (as in correctness and liveness). The benchmark in the following study [10] is of this type.

REFERENCES

[7] TBD

[8] J. Eisert, D. Hangleiter, N. Walk, I. Roth, D. Markham, R. Parekh, U. Chabaud, and E. Kashefi, "Quantum certification and benchmarking," Nature Reviews Physics, 6/17/2020.

VI. QUANTUM BENCHMARKING EXPERIMENTS

This study [10] implements a benchmark on IBM quantum hardware available in 2017. They implement an algorithm on a 5 qubit quantum processor 8192 times (a maximum set by the hardware). The goal of the repeated measurement is to determine if previous results have any impact on future results, which an ideal quantum processor will not. This lack of memory of past events is known as a Markovian process and it is a feature that should be present in any quantum processor. Their method involves implementing an algorithm that entangles qubits and applies a series of gates then makes a measurement with a known theoretical outcome. At the time of this paper the hardware was highly dependent on calibration but gave qualitative agreement with theory. The code is shown in an appendix and written in QASM.

References

[9] Martin Laforest, Error characterization and quantum control benchmarking in liquid state NMR using quantum information processing techniques, Ph.D. Dissertation, University of Waterloo, 2008.

[10] Kristel Michielsen, Madita Nocon, Dennis Willsch, Fengping Jin, Thomas Lippert, Hans De Raedt, Benchmarking gate-based quantum computers, Computer Physics Communications, Volume 220, 2017, Pages 44-55.

VII. Conclusion

Benchmarking is necessary for deciding what computer to use to solve your problem. In digital computing we can theoretically have a solution to a problem, but from a benchmark can learn that we will not have time to complete out theoretical program (ie factoring large prime numbers). Benchmarks help businesses and users decide what computers to buy and what resources will be needed to solve particular computing problems.

Quantum computing is able to solve whole new classes of computing problems, however users will need benchmarks to know what hardware is actually capable of, beyond what theory has shown to be possible. We can use new programming frameworks to program quantum computers and devise methods to determine how effective different hardware is at solving specific problems.

From experimentation we can see what researchers have been able to do to benchmark quantum computers in the past. We are now at a stage in the development of quantum computers where we can start to develop quantum benchmarking programs that are independent of hardware and programmed in human readable languages. With this capability we can develop programs to compare the available hardware and evaluate its efficacy at solving real world problems.

VIII. Revised Problem Statement

Quantum hardware is starting to become available to business users. There are many quantum algorithms available. The goal of this project is to create a survey of quantum benchmarking. The survey will cover the basic theory of a specific kind of quantum computer – the quantum processing unit, understand benchmarking historically and currently from a digital perspective, and finally explore how benchmarks will be used with a quantum computer. We will cover the basic frameworks available to program quantum computers as well as the theory and practice of benchmarking quantum computers.

IX. Research Plan

A. Timeline

- Write intro to quantum computing
- Complete section on historical benchmarking
- Complete section on quantum programming frameworks
- Describe a few different kinds of quantum benchmarks
- Create a sample program for benchmarking a QC
- Complete first draft of paper
- Revise and update

B. Preparation and concerns

Currently I am working through some text on the basics of quantum computing. My main concern for this project is that the material is very dense and I may not have time to fully understand the theory behind the quantum benchmarking studies I am using as references. I believe that if I narrow my scope sufficiently I will be able to complete the project.

C. Plan-B

In the case that I cannot complete the project as outlined I would narrow the scope further to only compare digital benchmarking to quantum benchmarking.