TECHNOLOGY SCIENCE JOURNAL FOR TEENS

JUNE 2019

How can we work with quantum computers today?

Authors:
Alwin Zulehner and Robert Wille
Associate Editors:
Elitsa Panayotova and Rachel Watson

Abstract

Quantum computers are a new, promising technology still in its infancy. Our conventional computers are already quite powerful. But this new technology could speed things up a lot! Because of this, many computer companies have already started building quantum computers. Unfortunately, they are still rather small compared to what we expect in the future. So how can we prepare programmers and users for this new technology? One way to do so: simulate the quantum computer on our conventional machines. Of course, this is very complex – if it were easy, we would not need a quantum computer in the first place. Here we propose a method that tackles this complexity using so-called *decision diagrams*. We tested our method by attempting several different quantum computations. Then we compared it to the other existing simulators. Our approach outperforms the other solutions in many cases. This allows everyone to simulate quantum computations today – even before the really powerful quantum computers are available.

Introduction

Have you ever wondered how your computer works? Maybe you have heard that it all boils down to ones and zeroes. Every input (and output) of a computer is actually a piece of information represented by "power on" or "power off" signals of electricity – or ones and zeroes. This on/off state is represented by a *bit* and it's the smallest unit of data in a computer. After we input some information (with a keyboard for example), the computer translates this information to bits. Then it processes the data by modifying and combining the bits. This happens with the help of *logic gates* which set rules about how the data will change. After the computer processes the information, it gives you the result.

Computers have become very fast following this method. In fact, a modern computer can easily process billions of operations per second. But for some problems, they still have limitations (and will for a long time). One of the most famous examples is *integer factorization*. That's when we take a number and figure out which numbers can be multiplied to produce it. For the number 15, the factors would be 3 and 5. Factorization is very important for internet security today – everything from online banking to your social media profiles are protected by encryption relying on factorization. While very easy for a number like 15, this is very difficult for large numbers. In fact, it can take years on today's computers! That's because of the *exponential growth* it involves. These limitations are why scientists have attempted to create *quantum* computers (Fig. 1).

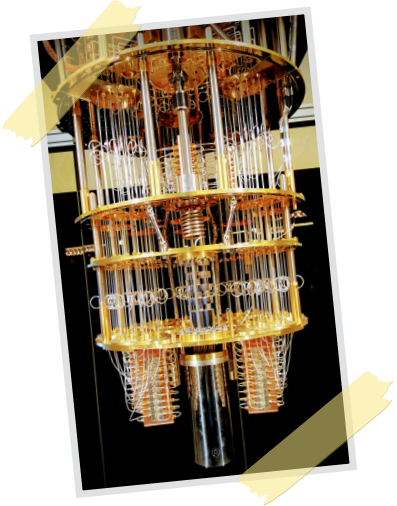


Figure 1:
IBM Q System One, a 20-qubit quantum computer.
Shared by Lars Plougmann under Creative Commons 2.0.

But what's the difference? Just as a conventional computer works with bits, quantum computers work with *qubits*. We can still represent their states as 1 and 0, but qubits can also be in a state which is 1 and 0 at the same time. For instance, a qubit can be in a state with 25% 1 and 75% 0. Even so, when we measure it, the qubit collapses to 1 or 0. This phenomenon is called *superposition* and it's one reason computations can be much faster with a quantum computer.

Companies such as IBM, Google, Intel, and more have already built quantum computers. But they are still rather small, and the technology will probably take some years to evolve. Until then, we have to prepare programmers and users for these computers by simulating them on conventional machines. So this is what we wanted to achieve: a more advanced and faster simulation of quantum computations.

For the curious: Another reason is entanglement. Sometimes two quantum particles correlate with each other. Knowing the state of one entangled particle makes it possible to predict the state of the second one regardless of how far apart they are.

Methods

How have researchers simulated quantum computations so far? They have just represented quantum states and quantum operations in conventional computers. *Vectors* are perfect for this, since they represent quantum input data. And just as in conventional computers, the input data goes through quantum logic gates, represented by *matrices*.

These gates/matrices transform the vectors (the states of the qubits/quantum system) to new ones. The problem is that with every qubit added to the system, these vectors and matrices double in size. That means a 1-qubit system requires 2x2 matrices and 1x2 vectors, a 2-qubit system requires 4x4 matrices and 1x4 vectors, and so on (Fig. 2). A quantum computer with only 30 qubits would require vectors and matrices of size 2^{30} - that's over a billion! With 300 qubits we would already reach a number of vector and matrix entries that is larger than the number of the atoms in the universe. Obviously, that makes things complicated for conventional computers!

Another way to represent quantum computations is through *decision diagrams*. Here, a vector and matrix are split into sub-vectors and sub-matrices, respectively (Fig. 3). If sub-vectors or sub-matrices are identical or only differ by a common factor, they can be combined. In this way, decision diagrams take advantage of repeated structures that often occur in vectors and matrices. They represent those repetitions only once instead of several times. That means their size doesn't necessarily double anymore when one more qubit needs to be considered. This allows for a much more efficient simulation.

For 1-qubit systems

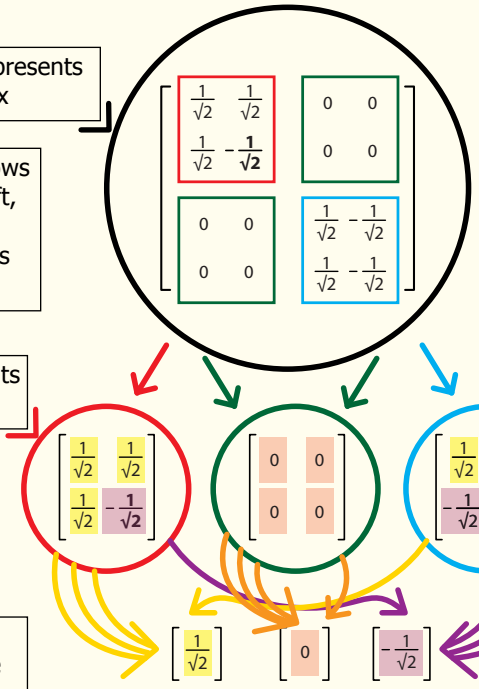
Operation		Input		Output
$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0		$\frac{1}{\sqrt{2}}$
$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	1		$-\frac{1}{\sqrt{2}}$
	×		=	
Size: 2x2		1x2		1x2

For 2-qubit systems

Operation				Input		Output
1	0	0	0	0		0
0	1	0	1	0		0
0	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	0		$\frac{1}{\sqrt{2}}$
0	0	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$	1		$-\frac{1}{\sqrt{2}}$
			×		=	
Size: 4x4				1x4		1x4

Figure 2:
Multiplying a matrix (representing a quantum operation) with a vector (representing a quantum input) leads to an output vector (representing the output of the operation). Unfortunately, the matrix/vector doubles in size with every qubit added to the system. That makes it very difficult to represent a quantum calculation with many qubits.

Figure 3



1. This node represents the entire matrix

2. The matrix is split and arrows point to the respective top-left, top-right, bottom-left, and bottom-right sub-matrices (in this order).

3. For example, this node represents the top-left sub-matrix.

4. The sub-matrices are split again and arrows point to the next four sub-matrices.

5. Eventually, this leads to the single matrix entries.

6. If the sub-matrices are identical (like the green ones), the nodes representing them are only needed once – saving space!

7. In summary, the decision diagram allows us to obtain all the matrix elements by following the arrows from the top to the bottom. For example, **the bold matrix entry** is obtained by following the first arrow (top-left sub-matrix) and then the fourth arrow (bottom-right sub-matrix).

8. In this way, the entire matrix, usually composed of 4x4=16 entries, can be represented with 4 nodes and the 3 entries at the bottom. So 4+3=7 nodes/entries, which is much more compact!

Results

We tested our new method by simulating several different quantum computations with different numbers of qubits. Then we compared them to the other simulators currently available. (See the details in the worksheet.)

Discussion

Our work introduces a new and powerful approach for the simulation of quantum computations based on decision diagrams. The results show that this new method outperforms the other simulators in many cases: it computes a lot faster and uses less memory in the process. Plus, the proposed simulator is able to simulate quantum computations for more qubits. This allows programmers and users to efficiently simulate computers that haven't been invented yet.

Another great aspect of our proposed method is that even a normal desktop computer can achieve all of these computations. Some of the other simulators rely on using supercomputers, which are in short supply. Our method gives a lot more researchers access to quantum computations.

Conclusion

Anyone can try IBM's quantum computer: you can even try it from your home. They have also integrated our simulator into their software tool. But what can you use quantum computer for? Here are a few examples people are already excited about:

- Artificial intelligence
- Building molecular structures
- Cyber security
- Discovering new drugs

Glossary of Key Terms

Bit – short for binary digit – the smallest unit of data in a computer. It has a single value of 0 or 1 (or true/false, yes/no).

Decision diagram – a compact graphical and mathematical representation of a decision situations, functions, or in our case, vectors and matrices.

Exponential growth – increase in number at a constantly growing rate. For example, when bacteria divide, one cell gives rise to 2 cells, and these two cells result in 4, then 8, then 16, etc.

Integer factorization – the decomposition of a composite number to smaller integers - down to prime numbers. For example, the prime factorization of 57 is 3 and 19.

Logic gate – a part of a circuit that performs logical functions. Usually two bits enter and the output is either 1 or 0. For example, the AND gate outputs 1 only if both input bits are 1.

Matrix – a collection of numbers arranged into a fixed number of rows and columns. In quantum computations, matrices define the operations of the computers' logic gates. They transform one vector (state of the qubit) to another (see Fig. 2).

Quantum – the smallest possible unit of matter or energy, zooming down to molecules, atoms and the parts atoms are made of.

Qubit – quantum bit, the basic unit of data in a quantum computer. It can be an electron or a photon for example.

Redundant – not needed, extra.

Superposition – the ability of a quantum system to be in a state that is both 1 and 0 at the same time. It may be partially 1 and partially 0 at any given moment, but when we measure it, it collapses to be either 1 or 0.

Vector – vectors can represent more values, not just 1 or 0 but both at the same time (see Fig. 2).

Check your understanding

1. What is the difference between a bit and a qubit?
2. Why can quantum computers work faster than conventional ones?
3. What is the difference between adding a bit and adding a qubit to a computer?
4. How do decision diagrams help with the simulation of quantum computations?

REFERENCES

Alwin Zulehner and Robert Wille. *Advanced Simulation of Quantum Computations*. 2018 IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.
https://doi.org/10.1109/TCAD.2018.2834427

Medium Free Code Camp: What is a Quantum Computer?
https://medium.freecodecamp.org/what-is-a-quantum-computer-explained-with-a-simple-example-b8f602035365

IBM: What is Quantum Computing?
https://www.research.ibm.com/ibm-q/learn/what-is-quantum-computing/