Quantum Implementation for Comparing Sets of Data

Yehuda Roth
Oranim college
E-mail: yudroth@gmail.com

Abstract. By simultaneously analyzing sets of data, we propose a quantum implementation for handling a large amount of data. Our research may be useful in big data analysis.

1. Introduction
Based on entangled states, quantum computers have the advantage of simultaneously implementing a large number of processes. The coherence of entanglement enables a single operator (logical gate) to be activated simultaneously on all of the states in the superposition [1, 2]. Consequently, to implement a quantum computer, a quantum algorithm has to be implemented. Quantum algorithms were first developed in the early-1990s, such as the Deutsch-Jozsa oracle-algorithm [3], which was followed by Ethan Bernstein and Umesh Vazirani’s[4][5] algorithm. In Simon’s algorithm,[6] the advantage of a quantum computer is demonstrated with an algorithm that is exponentially faster than any classical algorithm. Meanwhile, Shor’s algorithm provides a polynomial complexity, whereas a classical algorithm usually take super-polynomial time (i.e. an algorithm not bounded above by any polynomial) [7]. Grover introduced a search engine algorithm with complexity $O\left(\sqrt{N}\right)$ that is more efficient than the classical algorithm, which is $O\left(N/2\right)$ [8]. The main problem in all of this promising research is the implementation; that is, the ability to built a quantum computer in reasonable terms. In recent years, the digital world has grown very quickly and has become ever more complex. This complexity is associated with the term big data[9], which is a set of data that is so large that it cannot be effectively managed by conventional data management tools [10]. Although it seems that quantum computers are an ideal tool to serve these big data systems, quantum computers are still impracticable to implement.

In this paper, we propose a different quantum approach that can simultaneously analyse a large amount of data. Although our process allows many processes to work simultaneously, it is not within the conventional frame of quantum computers.

2. Quantum Model for Data Comparison
Our system is composed of two $N$ cells in a row, where each cell is a 2-levels-quantum system; as shown in fig. 1. The two rows are then compared, where the first row accepts the input data and the second row serves for recognition or comparison purposes. A qubit is defined energetically though the ground and exited states, $|g\rangle$ and $|e\rangle$, of each cell, respectively. For example, a row can be composed of $1/2$-spin particles that are subject to a magnetic fields. The ground state is defined when a spin points in the magnetic field orientation whereas exited state is when it points in the opposite orientation. A cell in an exited state tends to emit a photon to decay to the ground state. In our system, photons are only emitted through channels that connect each cell in a row to a compatible cell in the second line. Thus, between two compatible cells, we can define the following rules:
Figure 1. Three segments of the two rows of $N$ components. Each cell is attached to a compatible cell with a photon’s channel.

Rule 1 No photon is emitted; both compatible cells are in the same state $|g\rangle$ or $|e\rangle$.

Rule 2 A photon is emitted: the compatible cells are in different states, regardless of the row identity.

By denoting the input row with the symbol $\rightarrow\blacksquare$, the recognition or comparing row with $\leftarrow\blacksquare$ and a pair of compatible cells with the subscript $k$, we can express the rule with the following states:

States for rule 1

No photon is emitted:

$$|0\pm\rangle_k = \frac{1}{\sqrt{2}} \left( |g\rangle_k,\rightarrow\blacksquare |g\rangle_k,\leftarrow\blacksquare \pm |e\rangle_k,\rightarrow\blacksquare |e\rangle_k,\leftarrow\blacksquare \right).$$

(1)

States for rule 2

A photon is emitted:

$$|1\pm\rangle_k = \frac{1}{\sqrt{2}} \left( |g\rangle_k,\rightarrow\blacksquare |e\rangle_k,\leftarrow\blacksquare \pm |e\rangle_k,\rightarrow\blacksquare |g\rangle_k,\leftarrow\blacksquare \right),$$

(2)

where we allowed some relative phase between the states (the $\pm$ sign). Note that rule 1 is stable in time, whereas rule 2 can produce ongoing-outgoing photons between a pair of cells.

3. Implementation

1. Recognition Suppose that the input row is loaded with data. For example, photons that arrive from an image produce a distribution of qubits along the $\rightarrow\blacksquare$-row. For a large number of photons, this happens almost immediately. As a start, we can prepare the $\leftarrow\blacksquare$-row with all states in the ground state. If two compatible cells are in the ground state, then no photon will emit while an exited state in the input row will deliver a photon to another compatible cell. Thus, if all of the channels are marked, then we can recognise the input by tracking the photons’ distribution.

2. Comparing databases Suppose that we have two databases that we need to compare. One row is loaded with the first database and the second row is loaded with the other database. We then allow the transmission of photons. If no photons are emitted, then the databases are identical. Observing a "lighted" channel allows us to track the differences.
3. **Conditional commands** If an *if*- command, such as *If a = b and c = d, ...*, runs in a computer program, then the *a b c* and *d* databases are compared. The difference between this part to the previous part is that the *if* command must be followed with some continuous action rather than just a comparison (*If a = b and c = d, then...*).

4. **Discussion**

In this paper we have offered a conceptual device that is mostly able to compare two databases. In computing, a system is generally more efficient over other methods thanks to the hardware rather than to a sophisticated algorithm. Consequently, given that our conceptual device has no technology implementation, it is difficult to determine if our idea contributes to the effort of improving big data analysis. Assuming that the time consumed to emit or absorb all photons (almost simultaneously) is very small, then it is clear that the number of cells determines the efficiency of our device. A deep learning engine chip (called the Wafer Scale Engine by Cerebras) was developed in 2019 that contains $1.2 \times 10^{12}$ MOSFETs [11, 12, 13, 14]. Thus, if our quantum approach is considered to be promising, then maybe the appropriate technology will be invented.

1. Mooney, G.J., Hill, C.D. and Hollenberg, L.C.L. Entanglement in a 20-Qubit superconducting quantum computer. Scientific Reports 9, 13465 (2019). doi:10.1038/s41598-019-49805-7
2. Horodecki, R., Horodecki, P., Horodecki, M. and Horodecki, K. Quantum entanglement. Reviews of Modern Physics 81, 865 (2009).
3. Deutsch, D. and Jozsa, R. Rapid solutions of problems by quantum computation. Proceedings of the Royal Society of London A. 439, 553–558 (1992). Bibcode:1992RSPSA.439..553D. CiteSeerX 10.1.1.655.5997. doi:10.1098/rspa.1992.0167
4. Bernstein, E. and Vazirani, U. Quantum complexity theory. SIAM Journal on Computing 26(5), 1411–1473 (1997). doi:10.1137/S0097539796300291.
5. Fallek, S. D., Herold, C. D., McMahon, B. J., Maller, K. M., Brown, K. R. and Amini, J. M. Transport implementation of the Bernstein–Vazirani algorithm with ion qubits. New Journal of Physics 18, (2016). doi:10.1088/1367-2630/aab341.
6. Simon, D.R. On the power of quantum computation. Foundations of Computer Science, 1994 Proceedings, 35th Annual Symposium on: 116–123 (1994). Retrieved 2011-06-06.
7. Shor, P.W. Algorithms for quantum computation: discrete logarithms and factoring. Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE Computer Society Press. doi:10.1109/sfcs.1994.365700. ISBN 0818665807.
8. Grover, Lov K. A fast quantum mechanical algorithm for database search (1996). arXiv:quant-ph/9605043.
9. Bakshi, R Prasad and Sonali A. International Journal of Database Theory and Application, 9,1,45-66 (2016).
10. Moniruzzaman A.B.M. and Hossain, S.A. NoSQL database: new era of databases for big data analytics: classification, characteristics and comparison International Journal of Database Theory and Application, 6(4) (2013).
11. Manners, D. Samsung makes 1TB flash eUFS module. Electronics Weekly (January 30, 2019). Retrieved 23 June 2019.
12. Hruska, J. Cerebras Systems Unveils 1.2 Trillion Transistor Wafer-Scale Processor for AI. extremetech.com (August 2019). Retrieved 6 September 2019.
13. Feldman, M. Machine learning chip breaks new ground with waferscale integration. nextplatform.com (August 2019). Retrieved 6 September 2019.
14. Cutress, I. Hot chips 31 live blogs: Cerebras’ 1.2 trillion transistor deep learning processor. anandtech.com (August 2019). Retrieved 6 September 2019.

The author declares no competing interests, he is the only author of this study and non-financial interests.