6-2016

What Will Make Computers Faster: An Approach Based on Computational Complexity

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Comments:

Technical Report: UTEP-CS-16-39

What Will Make Computers Faster: An Approach Based on Computational Complexity

Vladik Kreinovich

Why do we want computer to become even faster? Computers are very fast already. Yes, they are getting faster year after year, but, honestly, other than making videos more realistic and faster to download, from the purely scientific viewpoint, there is no big reason to replace a 5-year old computer with the new one: already on a 5-year old computer, we can do all the editing and all simple computations very fast.

However, there are practical problems in which it is desirable to make computers much faster.

For example, we now have reasonable algorithms for predicting weather, algorithms that are much better in predicting tomorrow's weather – and even weather for a weak ahead – than a random guess. There are even algorithms that can reasonably well predict the path of a tornado.

However, while algorithms for predicting tomorrow's weather take less than a day on modern high performance computers, algorithms for predicting where the tornado will be in 15 minutes requires several days of intensive computations.

This large computation time defeats the purpose: we will learn where the tornado moved before the computations are over.

How can we make computers faster: traditional engineering approach. Traditional approach to making computers faster is based on introducing newer and newer engineering ideas:

- sometimes, a new material can help,
- sometimes, a new technology for manufacturing these computers, etc.

Limitations of the traditional approach. The traditional engineering approach to speeding up computers have worked reasonably well since 1960s: the resulting com-

Vladik Kreinovich
University of Texas at El Paso, 500 W. University, El Paso, TX 79968, USA
e-mail: vladik@utep.edu

putational speed grew exponentially, doubling every 1.5-2 years; this is known as *Moore's law* [10].

However, lately, this engineering-motivated speed-up have drastically slowed down: all existing ideas seems to be exhausted, and no new ideas have led to successful speed-up; see, e.g., [1]. The only possible exception is *quantum computing*: it is not yet practical, but most probably it will lead to a significant breakthrough; see, e.g., [11].

So how else can we speed up computers: analysis of the problem. Since the traditional engineering approach to speeding up computations does not seem to work as well as it did in the past, how can we speed up computers?

To answer this question, let us consider this problem from a more abstract viewpoint, without going into engineering details. Every time a new idea leads to a computer speed-up, this means that we start utilizing a new physical process in computation.

- In case of quantum computing, this is indeed a drastically new physical process that has never been previously used in computer design.
- In other cases, it is a new way to form a transistor out of several semiconductor layers, a process which is definitely new but not as drastically new as involving quantum processes.

In principle, we can simulate each of these processes on the previously designed computers. So, the only way that using these processes instead of simulating them speeds up computations is when computations needed to predict the results of these physical processes takes longer than waiting for their results. Thus, we arrive at the following idea.

How to speed up computers: new idea. To speed up computers, we need to utilize physical processes that are currently very difficult to predict, in the sense that:

- it is faster to run these processes and wait for their results
- than to compute the result of these processes on the existing computers.

What are these processes? So what are such processes? We have already mentioned tornados, but using tornados is probably an overkill.

Good news is that in addition to tornados, there are many other physical processes whose prediction on modern computers is difficult. There are many Grand Challenges initiatives that list such processes; see, e.g., [6]. Let us mention a few of them:

- prediction is very difficult for many *biological* processes, such as DNA folding and brain activity, and in predicting the effect of different chemical substances on a given protein, bacteria, or virus; neural networks (simulating brain activity) and DNA computing (simulating DNA-involving biochemical processes) are already a promising direction in the future of computing [12];
- computations are very difficult in *string theory*; so far, there have been no schemes for using string theory in computing – except that there are schemes

for using acausal anomalies that may be associated with cosmological strings; see, e.g., [8, 9, 14]; this seems to be an interesting direction to pursue;
- computations are also extremely difficult in *quantum chromodynamics* and *quantum gravity*; this is also a possible direction of research;
- on a more realistic level, many phenomena of *solid state physics* are difficult to compute – so maybe using these phenomena can also speed up computations;
- similarly, many phenomena of *plasma physics* are difficult to compute; maybe we can use plasma fields in computing?

How is the proposed approach different from the current one. At present, we had two different research directions, directions which are practically unrelated:

- on the one hand, computer engineers come up with new ideas on how to speed up computing;
- on the other hand, computational scientists analyze practical computational problems and try to separate them into feasible problems – which can be efficiently computed on modern computers – and computational problems whose computations require too long a time on modern computers.

Our analysis shows that these two direction are (or at least should be) related: namely, once it turns out that some physical process is difficult to compute, this means that we can potentially use devices based on this process to speed up computations.

When will all this happen? Of course, what we propose is a general research direction, not specific engineering solutions. The resulting vague ideas ("let's use quantum gravity") sounds more like science fiction than the real engineering ideas. So, a natural question is: when will all this be possible?

It is difficult to predict when exactly all these ideas will come to fruition, but we can have a crude estimate based on the history of quantum computing.

- The main ideas of quantum physics appeared in 1900, in the papers by Max Planck. The first vague idea of quantum computing were proposed by Richard Feynman in [3], approximately half a century later.
- A fully formal description of quantum physics, a description that fully relies on mathematical formulas as opposed to physicsts' intuition, appeared in 1932 [15], in a book written by John von Neumann (yes, the same John von Neumann who is also one of the computing pioneers); the first quantum computing algorithms appeared in 1990s, also about half-a-century later; see, e.g., [2, 4, 5, 11, 13].

Our prediction is thus that the above general ideas will be transformed into specific engineering suggestions approximately half a century from now.

It is difficult to predict beyond fifty years. The same argument shows that it is difficult to predict beyond half a century. Indeed:

- if we want to predict what will happen in 100 years,
- then we will need to know the physics that will only appear 50 years from now,

- and the history of physics shows that predicting future physical theories is not possible.

So let us be optimistic. Fifty years sounds like a lot, but it is not a very distant future that only one grandchildren will see: most young readers of this article will still be alive and active 50 years from now. So let us re-phrase the above predictions in realistic terms.

This rephrasing is similar to what Khrushchev, the 1950s-1960s Soviet leader, made in his 1961 speech: "The current generation of Soviet people will live in communism" – meaning in total happiness. Well, that prediction did not materialize, but we hope that the following one will be: "The current generation of people will see drastic computer speed-up – caused by using new physical phenomena in physics, biochemistry, etc."

Based on the current engineering methodologies, Moore's law may be dead, but we strongly believe that it will be still true 50 years from now – just new physics ideas will have to be used in computer design. Moore's law is dead – long live Moore's law!

References

1. 50 Years of Moore's Law: Special Report, in: IEEE Spectrum, April issue (2015)
2. Deutsch, D., Jozsa, R.: Rapid solutions of problems by quantum computation, In: Proceedings of the Royal Society of London, Ser. A, Vol. 439, pp. 553–558 (1992)
3. Feynman, R.P.: Simulating physics with computers, International Journal of Theoretical Physics, Vol. 21, No. 6/7, pp. 467–488 (1982)
4. Grover, L.K.: A fast quantum mechanical algorithm for database search, In: Proceedings of the 28th ACM Symposium on Theory of Computing, pp. 212–219 (1996)
5. Grover, L.K.: Quantum mechanics helps in searching for a needle in a haystack, In: Physical Reviews Letters, Vol. 79, No. 2, pp. 325–328 (1997)
6. Hoare, T., Milner, R.: Grand challenges for computing research, In: The Computer Journal, Vol. 48, No. 1, pp. 49–52 (2005)
7. Kavanagh, J., Hall, W. (eds.), Grand Challenges in Computing Research GCCR'08, UK Computing Research Committee, London, UK (2008)
8. Koshelev, M., Kreinovich, V: Towards computers of generation omega – non-equilibrium thermodynamics, granularity, and acausal processes: a brief survey, In: Proceedings of the International Conference on Intelligent Systems and Semiotics (ISAS'97), pp. 383–388, National Institute of Standards and Technology Publ., Gaithersburg, Maryland (1997)
9. Kreinovich, V., Finkelstein, A.M.: Towards applying computational complexity to foundations of physics, In: Notes of Mathematical Seminars of St. Petersburg Department of Steklov Institute of Mathematics, Vol. 316, pp. 63–110 (2004); reprinted in Journal of Mathematical Sciences, 2006, Vol. 134, No. 5, pp. 2358–2382 (2006)
10. Moore, G.E.: Cramming more components onto integrated circuits, In: Electronics, 19 April 1965, pp. 114–117 (1965)
11. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, U.K. (2000)
12. Paun, G., Rozenberg, G., Salomaa, A.: DNA Computing: New Computing Paradigms, Springer, Berlin, Heidelberg (2006)

13. Shor, P.: Algorithms for quantum computation: Discrete logarithms and factoring, In: Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science, pp. 124–134 (1994)
14. Thorne, K.S.: Black Holes and Time Warps: Einstein's Outrageous Legacy. W.W. Norton & Co., New York (1995)
15. von Neumann, J.: Mathematische Grundlagen der Quantenmechanik, Springer, Berlin, Heidelberg (1932); English translation: Mathematical Foundations of Quantum Mechanics, Princeton University Press, Princeton, New Jersey (1964)