Researcher sketches a path toward quantum computing

November 16 2017, by John Sullivan

Professor Margaret Martonosi answers questions about her recent article in Nature in which she and colleagues sketch the future of quantum computing. Credit: David Kelly Crow

As new devices move quantum computing closer to practical use, the

journal *Nature* recently asked Princeton computer scientist Margaret Martonosi and two colleagues to assess the state of software needed to exploit this powerful computational approach.

Relying on subtle quantum mechanical effects for data storage and computation, quantum computers show promise to vastly speed up certain types of calculations. Martonosi, the Hugh Trumbull Adams '35 Professor of Computer Science, explained in an interview that although quantum computers are fundamentally different than classical ones, both require an efficient chain of software to operate. Her co-authors in Nature are computer science professors Frederic Chong and Diana Franklin of the University of Chicago.

What is quantum computing, and how is it different from standard—or classical—computing?

In classical computing, we've built computers for many years that rely on binary values for what we call the state, or the storage data, in the machine. So the value can either be 0 or 1. And we built up the ability to do arithmetic or to do logic operations based on the 0 or 1 values. In quantum computers, instead of these classical 0 or 1 bits, we have what are called quantum bits or qubits. You can think of a qubit as a probabilistic distribution of many possible values. So it's not 0 or 1, but some "superposition" of different states. Being able to manipulate these complex states, one can do unique calculations that go beyond the simple addition or logic operations of a classical computer.

Quantum computing allows one to do considerably more powerful calculations, conceptually at least, with relatively fewer qubits than the bits of state required by a classical alternative. There are some quantum algorithms that show the opportunity for considerable speedup, sometimes even exponential speedup, over the classical approach. For

example, there are some large-scale problems that would take tens or hundreds of years to compute on a classical machine —rendering them essentially intractable—but if suitable quantum hardware existed, the corresponding quantum algorithm could allow those tasks to be solved in hours instead of decades. It's the fact that we can do things potentially exponentially faster in a quantum computer that has led the world to be very intrigued by the possibilities.

So a quantum computer is not just a faster version of a standard computer?

It's using profoundly different physical characteristics to do the calculations. And that allows it to be faster, potentially, at some calculations, although it still relies on classical sequencing of the operations and classical control of the operations. So one of the big emphasis areas over the past 10 years has been getting from quantum algorithms that show theoretically exponential speedup to seeing how these algorithms will really map to real quantum hardware, and what sort of speedups will be possible as we start to build actual quantum hardware.

Your article in *Nature* says that quantum computing has reached a critical stage, which you call an 'inflection point.' Why now?

It's a range of things. For many years we had quantum algorithms that theoretically sketched out how they could use quantum superposition and entanglement (the ability of quantum states to interact with each other), but didn't have any hardware to map onto. Meanwhile, there were physicists who were building individual qubit technologies, but building so few qubits—one at a time, or two at a time—that you couldn't really get a sense for how to actually compute with them.

What's happening now is that the number of qubits that can be built will foreseeably soon be large enough that one actually needs to think practically about how to build systems to compute with them. So where it was previously OK to simply build individual qubits and test their characteristics in a one-off way, now people are starting to think about how to build real computer systems out of them, including understanding how the storage will work, how the communication will work.

So when we talk about building quantum compilers (software that executes programs' instructions in the hardware) or quantum tool flows (software that optimizes applications), we do it for a few reasons. One reason is that when quantum computers of increasingly interesting sizes are built, we want to be able to compile for them. Another reason is that, even before the machines are built, we want to be able to assess different design tradeoffs better. So the tool flows that the paper discusses, the type my collaborators and I have worked on, are a way of doing some of the assessments that will help see which algorithms benefit from which technology choices, or which organizational choices, as researchers build the hardware.

The other aspect to the inflection point is in terms of interest and funding. We now are at a point where you can use a 16-qubit quantum computer on the web. IBM, through its Quantum Experience effort, has put out a quantum computer for anyone to use. Google, Microsoft, Intel and others are all pushing to build substantially larger quantum computers than have ever been built. And there's a bit of a race underway to see who will get how far and when. So with industry putting considerable attention to building quantum computers, I think it's raised the credibility that there's something here, there's something to focus on. And as a result, it's increased the pace at which other parts of the quantum research space have moved as well.

Could quantum computing be as sweeping as classical

computing or is it likely to be more specialized?

If you look at the quantum algorithms that have been developed so far, they are relatively focused. There are a few areas where quantum shows the potential for speedup, but there are a lot of areas where we don't yet have quantum algorithms that show speedup. So nobody sees quantum computing wholly supplanting classical. It will not be used in that way in the foreseeable future. Rather, people see quantum computing being useful for some very focused computations. You can think about it like a specialized accelerator for those computations.

For many years, a key catalyst for interest in quantum computing was the fact that many of our current encryption methods rely on the assumption that factoring large numbers will be computationally difficult. And quantum computing, particularly something called Shor's algorithm, has shown a way to speed that factoring up dramatically. So for many years, one of the key attention-getters about quantum was the concern of whether quantum computing would — quote, unquote—"break encryption."

What we're seeing right now is, first of all, the encryption community is developing new algorithms that are designed to be quantum resistant. That's progressing at some level. Simultaneously, we're seeing that the factoring algorithm that could "break encryption" actually requires so many qubits that it will be a while before we can use it to factor the large numbers that are used in our encryption algorithms. So, for that reason, factoring is not the biggest algorithmic attention-getter right now within the quantum computing community itself.

But rather, there are other algorithms that are getting attention in terms of things like simulating molecules. So-called quantum chemistry is of interest these days, and seems to be an application area that we could get to sooner with the kinds of machines we envision being able to build

earlier in the timeline.

You mention the concept of hybrid systems combining classical and quantum computing in the paper.

That's inevitable. You're not going to build quantum computer systems that are solely quantum. And people in the field know this, but it hasn't been well portrayed to the outside world. To make a quantum computer work, and to execute a set of quantum operations, you will still have a classical control sequencer that steps in through a set of physical manipulations. And so you will always have this classical control of quantum operations.

So that duality will be there no matter what. And there's interesting work to be done in terms of deciding how to organize that, how much classical control goes where. The quantum operations are often done under very low temperatures, close to absolute zero. The question is, how much of that classical control can be done at those temperatures versus how much should be done out at room temperature the way we're used to doing classical computing? And so those kinds of design tradeoffs remain mostly unanswered.

Quantum computing is very exciting, but there's no guarantee that quantum computing will have the same trajectory or the same breath that classical computing has had. In many ways, everything right now looks as if quantum computing may be more narrow than classical in its applications. But it's still useful and instructive to try to look across different innovation cycles and try to see where you see parallels or not.

Quantum computing might be just another useful way to do computing?

The hope is that it will accelerate certain things quite a bit. So, for example, if quantum chemistry becomes the viable application that it seems to be, then one can imagine that being deeply influential for things like agriculture, understanding how to build better fertilizers, and so forth, and also for drug development. So even if it is somewhat focused in where it has applicability, it could still be very impactful in those areas.

More information: Frederic T. Chong et al. Programming languages and compiler design for realistic quantum hardware, *Nature* (2017). DOI: 10.1038/nature23459

Provided by Princeton University