# Beyond the qubit: quantum computing, practical alternatives, and Memory-Driven Computing

Ray Beausoleil, PhD., HPE Senior Fellow, Large-Scale Integrated Photonics, Hewlett Packard Labs  
Rebecca Lewington, Senior Marketing Manager, Analytics and Advanced Architectures Marketing, HPE

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>2</td>
</tr>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>The state of the art: quantum computers today</td>
<td>2</td>
</tr>
<tr>
<td>How quantum computers work</td>
<td>3</td>
</tr>
<tr>
<td>Data analytics: a real-world failure</td>
<td>4</td>
</tr>
<tr>
<td>A real-world solution</td>
<td>5</td>
</tr>
<tr>
<td>Optimization problems: making NP-hard easy</td>
<td>6</td>
</tr>
<tr>
<td>Accelerators: task-specific processing engines</td>
<td>8</td>
</tr>
<tr>
<td>An architecture for accelerators: Memory-Driven Computing</td>
<td>8</td>
</tr>
<tr>
<td>Conclusion</td>
<td>9</td>
</tr>
</tbody>
</table>
Abstract

We are experiencing an exponential increase in data, coming from an explosion of sources and we have a vanishingly small time to turn that data into meaningful action. We need new technology to bridge the widening gap between what we want to achieve and our actual capabilities. Quantum computing could be a powerful technique to solve quantum-like challenges in drug discovery and material science. But quantum computing is inherently unsuitable to solve the data-intensive challenges faced by our enterprise customers.

There is a better way.

HPE is developing practical accelerators capable of massive performance and energy-efficiency gains and an architecture to plug those accelerators into: Memory-Driven Computing.

Business customers should not wait for quantum computing. Our innovations promise to provide users with practical solutions to real-world enterprise problems much sooner.

Introduction

We are experiencing an exponential increase in data, coming from an explosion of sources and we have a vanishingly small time to turn that data into meaningful action. With transistor scaling slowing or stopped (depending on who you ask) the gap between what we desire to accomplish and what we can accomplish is widening. Clearly, we need a new kind of computer.

Quantum computing\(^1\) is often held up as a solution to all our data-driven prayers. But is that true? Or are there quicker, more practical ways to solve those problems? To answer the question, we have to understand classical computing, quantum computing, and the difference between the two.

Classical computing

The defining elements of classical computing are that each bit is always either 1 or 0; we can watch it work, without affecting the result; there is well-understood error correction; and its actions are reliable and repeatable.

Quantum computing

Quantum computing on the other hand, is defined by each qubit being 1 and 0 at the same time. Although each qubit collapses to 1 or 0 when you look at it, it can hold exponentially more than two bits of information. Fifty perfect qubits can hold a petabit (more than 100 terabytes) of information, and every ten additional qubits multiplies that capacity by a factor of a thousand. Sixty perfect qubits can hold an exabit, 70 can hold a zettabyte. This exponentiality accounts for a great deal of the enthusiasm surrounding quantum computing today.

But there is a problem: Today's qubits are noisy, or “dirty.”\(^2\) We can compensate for noise, but to so we need 1,000 noisy qubits to make one “clean” qubit.

Additionally, for reasons we will explain shortly, you cannot watch a quantum computer work. It is neither reliable nor easily repeatable. You must run the computation many times to get a reliable result.

The idea of using quantum effects for computation was first mentioned by Richard Feynman in 1959,\(^3\) nearly seventy years ago. Why do not have them, yet? Because it is very hard to do.

The state of the art: quantum computers today

When people hear the term “quantum computer” what most of them are thinking about is actually a universal quantum computer. A useful universal quantum computer would need about 50 clean qubits, which would require 50,000 dirty qubits. However, the biggest quantum computer today has only dirty 72 qubits! This difference does not represent an insuperable barrier. Remember, silicon chips went from one to a billion transistors in just 50 years. But it is going to take time.

Quantum annealing, which is a popular current bridge between the present and a future in which we could produce enough clean qubits to make quantum computing practical, is designed for one class of applications, optimization problems, and it is not clear whether these machines actually do any useful computation.\(^4\)

Both universal quantum computers and quantum annealing machines need to be operated very close to absolute zero (−273 Celsius). This reality means you are not going to be able to put one in your company servers, much less your laptop or phone.
How quantum computers work

With the proper technology, human observers can see how the Newtonian universe works. But the quantum universe, due to its size, is inherently non-observable. Figure 1 is a diagram that shows everything humans can directly observe about the quantum universe—nothing!

Figure 1. Map of the observable quantum universe

Table 1 illustrates how an experiment is conducted using a universal quantum computer.

Table 1. Running an experiment on a quantum computer

<table>
<thead>
<tr>
<th>Step 1: Open a portal to the quantum universe.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2: Next you insert your experiment and instruments. The computer required to control the quantum computer has to stay outside.</td>
</tr>
<tr>
<td>Step 3: Close the doors, hit the start button and wait. You cannot look inside until it is done or the soufflé will collapse.</td>
</tr>
<tr>
<td>Step 4: When the bell rings, the soufflé is done and you can look inside.</td>
</tr>
</tbody>
</table>
Note that in step 3, there is a little bit of leakage from the classical universe. The seal is never going to be perfect. The macro universe tends to leak into the quantum universe, so the hole tends toward collapse, which would destroy the experiment. This is called “decoherence.” Dirty bits collapse faster. The cleaner the bit, the more likely the experiment would be capable of completing.

If even partial decoherence occurs, the experiment could result in the wrong result. The solution is to repeat the experiment several times until you have a statistically significant number of matching results.

It may sound like we are making a case that quantum computing has no utility. Not so. There is a host of applications for which quantum computing would be extremely useful. What these application areas have in common is that they are all problems that are simple to state, complex to calculate, and have a simple output.

- Drug discovery
- Materials design
- Decryption
- Quantum dynamics

The problem is what is missing from this list: Big Data analytics and optimization problems.

**Data analytics: a real-world failure**

Materials science and drug discovery are terrific challenges with real benefits to all of us, but immediate problems that we are currently being overwhelmed by, such as mainstream enterprise analytics—big data and AI—are not going to be solved by quantum computing. Quantum computing, by its very nature, is not going to be useful for real customer problems involving big data. In the end, quantum computers are not very good at the “Three Rs”: readin’, writin’, and ‘rithmetic.

To contrast their various strengths, we can examine how to train a neural net using both classical and quantum computing.

Let us say we have a petabit of training data held in a classical database (Figure 2). In classical computing, we load it into memory and we do some computation. The output is a trained model. A typical deep learning model with 200 layers and 50 nodes per layer might be a manageable 25 megabits (about three megabytes).

![Figure 2. Teaching a neural network using classical computation](image)

With quantum computing, well, it is not quite so elegant or effective (Figure 3). For one thing, you cannot just load classical data directly into a quantum memory. Nature doesn’t work that way. So, we would need a quantum encoder. The problem with that solution is that we do not know how to build one. But let us assume this problem will be overcome in time.
Using the quantum encoder, we can now pack that petabit into just 50 clean qubits. To get them, we will need 50,000 dirty ones. That means we will need a thousandfold improvement in the size of the machine we can build. But again, we will figure that out, eventually.

Now we run the experiment. We open the doors and take a measurement and here is where we see yet another problem, a big one: The output is only 50 bits, enough to contain only one weight in a neural network. Given our whole model requires 25 megabits, we would need to run the whole experiment at least 500,000 times.

Yet another complication of using quantum computing to train a neural net is the process of reading the output clears the quantum memory. There is no such thing, as a quantum hard drive. You cannot simply copy quantum states from one place to another. So you have to start again from the classical database. Every time. Quantum computing, even if we allow it a theoretical complexity and completeness which it does not currently have, is not a practical instrument in training a neural net. Nor is it useful for any computing task involving large datasets, which is virtually everything enterprises need to do.

A real-world solution

Quantum computing has become a popular go-to when examining possible paths around the end of Moore's Law. However, we have shown that quantum computing is not, and likely never will be, the solution for real analytics problems. But Hewlett Packard Labs, by moving beyond the transistor, has developed something that is.

We have developed a novel technology, the Dot Product Engine, which employs analog circuit elements to perform the vector matrix multiplication that underlies many applications such as deep learning, video processing, graph analytics, and streaming analytics. It works using a memristor crossbar array, which multiplies and adds through Ohm’s law and Kirchhoff’s law. Instead of fetching network weights from memory for every pass, we load our weights into the crossbar array where they remain until reprogrammed, even if the power goes away. This analog computation in situ produces an answer that comes in one step, instead of by executing thousands of lines of code.

This solution is 10-100X faster and requires 10-1000X less energy than typical GPUs on a variety of neural network applications, especially for larger networks. The data flows in, is transformed instantly, and comes right out. No encoding necessary. This solution is here, real, and does not require a dilution refrigerator.

We have built prototype microchips to prove this technology works\(^5, 6, 7\) (Figure 4). Our next step is to build complete systems-on-chip that can plug into a computer, much as a GPU does today. We are also working on the software stack, so that using the Dot Product Engine will be as easy as a function call from an API.
This is different from accelerators like Google™’s Tensor Engine in two important ways. First, while other accelerators perform ordinary step-by-step calculation, the Dot Product Engine effects the calculation in a single step. Second, those accelerators are all single-purpose. The Dot Product Engine, on the other hand, can be programmed and reprogrammed for any task requiring matrix operations, which are myriad.

**Optimization problems: making NP-hard easy**

One of the most intractable problem types for computing is the NP-hard category. These are allegedly unsolvable business optimization problems, with the “travelling salesman problem” being the best known. In this problem, a traveling salesman wants to visit every city in his territory in the most efficient way possible and taking the least amount of time.

This sounds like a reasonably simple problem, until you really dig into it. Then you discover that a mere 15 destination cities produce 4.3 billion routes. NP-hard problems are a class for which the execution time grows exponentially with the size of the problem.

Such a problem quickly overwhelms computers due to their requirement of analyzing every possible combination of routes prior to rendering a decision. When you consider that NP-hard problems include challenges like airline optimization, trade logistics, fleet management, and traffic control systems, and you start to see why it is considered one of the most important outstanding problems in computing.

Today, we can only produce approximate solutions, and even that requires enormous compute resources. Quantum computing, particularly quantum annealing, has been proposed as a true solution for NP-hard problems. The reality of this claim is a subject of lively debate in the scientific community.4 Regardless, we will still have to wait until 50,000 qubit machines actually exist to test that assertion. But we have a better way: More practical, more energy efficient, and available on a much shorter timeline.

Our solution is to express all NP-hard problems using the same mathematical model, enabling us to find a general purpose solution.

Using the Ising Model,8 all NP-hard problems can be expressed and solved the same way. Taking the traveling salesman problem as an example, each node is a city and each line is a road. The model shows all cities connected to all other cities and expresses the network as an “energy.” The action then is to find the minimum energy state.

In the case of the travelling salesman problem, the minimum energy state is easy to visualize: It is the route that uses the least amount of fuel.

So, the math is clear. The next step is to employ technology to arrive at the solution. To do so, we harness the strange properties of light at the microscale to compute without transistors.

When you throw two pebbles into water, the two sets of waves can add or subtract (or interfere) to produce high and low spots. At the microscale, silicon is transparent to laser light, allowing us to do an analogous action. We use a pair of silicon channels (waveguides) each with two inputs and two outputs. As we change the relative phase of the two laser inputs (similar to dropping the pebbles at different times), we can steer the light to one output or the other. This is called “coherent interference.”

Our solution, the Optical “Coherent Ising Machine” prototype, is not an idea, not a theory, not a science experiment. We have built it and it works. We have built prototypes with a world-record thousand optical components 9 (Figure 5). We know we can scale this to 100,000 components, using ordinary chip manufacturing processes to make a simple chip requiring no refrigerators that can plug into a classical computer. As with the Dot Product Engine, we also know how the APIs will work and 95% of the compute stack is already in place.
Our CIM consists of a set of nonlinear nodes, or “neurons,” connected all-to-all in such a way that the output of the nodes can be added together like the waves in the above example. We program the Ising solver by setting the phases of the waves at the start using a special compiler. We run the solver until it settles into a stable state, a process that takes just a few milliseconds and then read out the result by looking at the nodes.

We know from theory that the solution time will grow as a polynomial with problem size. That means the growth is non-exponential and the problem is computationally tractable. By focusing on innovative approaches to real technology, we have made a class of intractable compute tasks practical. If that turns out to be true, we will have made history.

We are not alone in thinking Coherent Ising Machines are a better solution than quantum annealing. Figure 6 compares a benchtop CIM implementation with 1,000 nodes built by a team in Japan with commercially available quantum annealers from D-Wave Systems. The results show that the Japanese CIM works, and continues to work as the problem size grows. This indicates that our microscale version will also scale well. In contrast, the limited connectivity between qubits in D-Wave quantum annealers means that solving real-world size problems is not going to work.

Figure 5. Optical “Coherent Ising Machine” prototype

**Figure 6.** Comparison of CIM and quantum annealing for solving NP-hard problems.

- **Optical CIM**
  - High probability of success
  - Milliseconds, scales well

- **D-Wave**
  - It’s not going to work
Accelerators: task-specific processing engines

We know that general purpose computers have their limitations. They are not fast enough, or energy-efficient enough, to meet the challenges of the future. Instead of theoretical solutions that at best will be practical only in the future, we need different computational techniques for different problems right now. We call these task specific solutions “accelerators,” special purpose components designed to perform specific tasks very fast, using very little energy. Both the Dot Product Engine and the Optical Coherent Ising Machine are examples of accelerators. In fact, a quantum computer can also be thought of as an accelerator, one that is the ideal way to solve quantum-like tasks.

Table 2. Accelerator applicability by application

<table>
<thead>
<tr>
<th>Quantum systems</th>
<th>Quantum computing</th>
<th>Analog neuromorphic</th>
<th>Optical computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Drug discovery</td>
<td>Excellent</td>
<td>Nope</td>
<td>Questionable</td>
</tr>
<tr>
<td>- Materials design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Decryption</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Quantum dynamics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analytics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Deep learning</td>
<td>Nope</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Signal processing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Graph analytics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Streaming analytics</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP-hard</td>
<td>Questionable</td>
<td>Questionable</td>
<td>Promising</td>
</tr>
<tr>
<td>- Traveling salesman</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Airline optimization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Logistics optimization</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An architecture for accelerators: Memory-Driven Computing

Today, the CPU vendor defines the architecture. New technologies can only be introduced with a CPU update. If you want to connect more than the CPU has lanes for, you have to get a second CPU and deal with complex, slow communication between them.

Memory-Driven Computing breaks those constraints. We are not just making incremental improvements within the same old architecture. We are re-architecting. This new architecture features ultimate composability, allowing any combination of components to be connected together at will, over an ultra-fast web of interconnects (called a “fabric”), enables communication using simple, efficient commands, all at the speed of memory communications. We have built the perfect architecture to take advantage of all kinds of accelerators, as illustrated in Figure 7.
Memory-Driven Computing is built on a fabric that uses a new open interconnect standard called Gen-Z created by most major players in the IT industry. We call this “precision compute.” Assembling and reassembling precisely the right elements to complete a workload as quickly and efficiently as possible.

This composability means you can take the following actions.

• Mix and match different processor types: x86, Arm®, open-source RISC-V, etc.
• Assemble different memory types, both volatile and persistent; as new memory technologies come on the market, they can slot right in, without needing a new generation of CPUs
• Add task-specific accelerators of any kind

Any resource, any data, is available at the speed of memory. There is less need to move data around, less need to convert between “working” and “persistent” data structures. Today’s computers spend most of their time moving data between tiers of memory and storage. When you stop chopping up data sets to mask the limitations of today’s CPU-centric architecture, massive speed gains can be achieved.

Memory-Driven Computing is a universal, open architecture that can mix and match; compose and recompose, resources to address any computing task in any location from every edge to any cloud.

Imagine an edge device like a smart video camera. You could combine a neural net accelerator with non-volatile memory to create an ultra-low power vision system capable of understanding its environment.

Memory-Driven Computing can be used to replace today’s large hyperscale deployments with much greater responsiveness and power-efficiency. Microservices, containers, and virtual machines can be spun up and down in microseconds at massive scale.

You can build big-memory machines capable of holding any data set at once. When you stop chopping up data sets to mask the limitations of today’s CPU-centric architecture, massive speed gains can be achieved. More on that in a couple of slides’ time.

At the highest end, we are using Memory-Driven Computing principles to build exascale computers. At a billion, billion calculations per second, exascale computing will transform our ability to solve some of mankind’s most complex problems, arriving at solutions never before thought possible. Memory-Driven Computing principles allow us to compose diverse computing elements to build high-end systems to solve previously impossible enterprise problems at any scale, advantages that inform smaller enterprise HPC systems.

**Conclusion**

There are a lot of people and companies out there counting on quantum computing to solve their problems. For some problems, it will not do so for a long time. For Big Data analytics problems that constitute the bulk of enterprise computing, it is impossible to see how it will ever will. In the meantime, the problems enterprise consumers face simply cannot wait.

While we wish quantum computing researchers every success, we believe HPE can best use its innovation engine to follow a different path to the same goal and do so on a much shorter time frame.

In the face of the enormous obstacles besetting the enterprise, the only practical path is an unconventional path. That less traveled path is the path we are on. Join us. The views of the future are breathtaking.

**Learn more at**
labs.hpe.com