Practical Annealing-Based Quantum Computing

WHITE PAPER

Catherine C. McGeoch, Richard Harris, Steven P. Reinhardt, and Paul Bunyk

2019-07-22

Overview

We give an overview of quantum computers that are based on the annealing paradigm and manufactured by D-Wave. We present an introductory survey of this approach to quantum computing, together with a snapshot of what is known about performance. We make some evidence-based predictions about future developments in this region of the quantum computing space.

CONTACT

Corporate Headquarters
3033 Beta Ave
Burnaby, BC V5G 4M9
Canada
Tel. 604-630-1428

US Office
2650 E Bayshore Rd
Palo Alto, CA 94303

Email: info@dwavesys.com

www.dwavesys.com

14-1036A-B

Notice and Disclaimer

This is the accepted version of "Practical Annealing-Based Quantum Computing." *IEEE Computer Magazine.* June 2019, pp. 38-46, vol. 52 DOI Bookmark: 10.1109/MC.2019.2908836

Contents

1 Introduction

Quantum computers have the potential to facilitate extraordinary breakthroughs that will completely transform modern society: lives could be saved by more efficient and effective drug designs; discovery in materials science could be revolutionized by simulations of quantum processes; and internet encryption as we know it could be broken and then replaced by much more secure methods [1,2,3]. Or maybe quantum computers will never do anything useful [4].

Much of the public's confusion concerning the current state of quantum computing comes from receiving mixed messages about what is possible in theory versus what can be achieved in reality. The amazing theoretical capabilities of quantum computers are based on having registers of qubits that can exhibit exotic properties, such as superposition and entanglement, that are not available to registers of classical bits. However, those seemingly omnipotent quantum computation resources are fragile in the face of noise and are also difficult to control. The challenge before every organization currently attempting to realize practical quantum computing is to build quantum hardware that is big enough, is sufficiently robust against errors, and possesses sufficient control fidelity so as to outperform the world's most powerful classical computers at tackling some of the world's most challenging computational problems (nobody claims that *every* computation will be faster, better, or cheaper on a quantum computer). The technical hurdles are daunting—and some argue that they are insurmountable.

There are two dominant approaches in play for achieving practical quantum computing (QC) at an industrial scale: *gate model* (GM), also known as the circuit model, and *quantum annealing* (QA) (the term *adiabatic quantum computing* refers to the same general concept). Superficially, these approaches are often portrayed as being completely unrelated, but careful examination reveals that the models can be equivalent [5]. Nonetheless, GM and QA are markedly different approaches in practice, which has considerable bearing on their prospects for achieving practical advantage over classical computers. D-Wave™ is the only company that builds, sells, and sells time on, annealing-based quantum computers [6].

For over a decade, D-Wave has focused on delivering practical quantum computers aimed at solving NP-hard problems. A somewhat restricted version of QA was selected for this purpose because it is more robust against noise than known approaches to GM. Although the restricted model is general enough to express any Turing-computable function, this choice meant temporarily foregoing some of the grander goals of quantum computing, such as efficiently solving the Schrödinger equation for a large number of electrons; the company intends to remove this restriction at some future date.

The strategic decision to prioritize increasing qubit counts and developing a strong user base over implementing the fully-general computational model provided the impetus for subsequent developments, such as successful deployments of commercially available QA systems starting in 2011, a sixteen-fold increase in qubit counts in later systems [the D-Wave 2X™ (1000 qubits) system, and the D-Wave 2000Q™ (2000 qubits) system shown in Figure 1, are in current use], and the proliferation of more than 100 early-phase applications implemented on D-Wave products.

This article provides a snapshot of where D-Wave currently stands and where we expect to go in our quest to build useful quantum computers. We begin with a brief introduction to the physics of quantum annealing and a description of D-Wave processors from the user's perspective. An overview of how D-Wave systems have been used to date and a survey of the current state of understanding regarding performance of these quantum computing systems is then provided. Looking to the future, we offer six evidence-based predictions pertaining to annealing-based quantum computers that we anticipate will come to pass within 2 to 5 years.

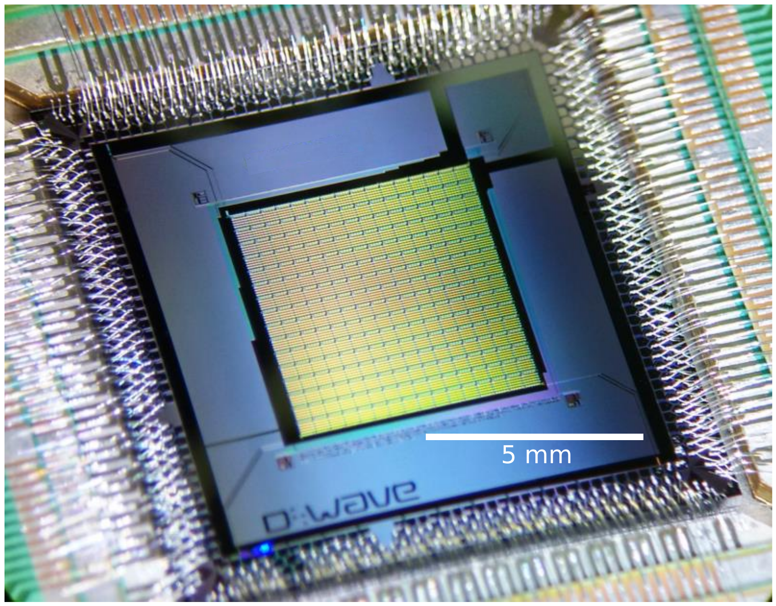

Figure 1. A D-Wave 2000Q quantum processing unit (QPU). It is a superconducting circuit containing 2048 qubits, and it is operated at a temperature below 20 mK.

2 Overview

2.1 The Physics

Several sources are available that describe the physics of QA in detail [7,8]. For brevity, all that will be stated here is that QA is a means of harnessing the physics of quantum phase transitions for performing computations. Here, a phase transition is defined as a discrete change in some macroscopic property of a physical system that has been induced by tuning an external control parameter. As implemented by D-Wave, the physical system is a network of qubits that are pairwise coupled and biased in a way that encodes an NP-hard problem of interest. The goal is to find a configuration of spin values (+1 and -1) that can be assigned to the qubits such that the cost function of the NP-hard problem is minimized. Such a configuration is referred to in physics terms as a *ground state*.

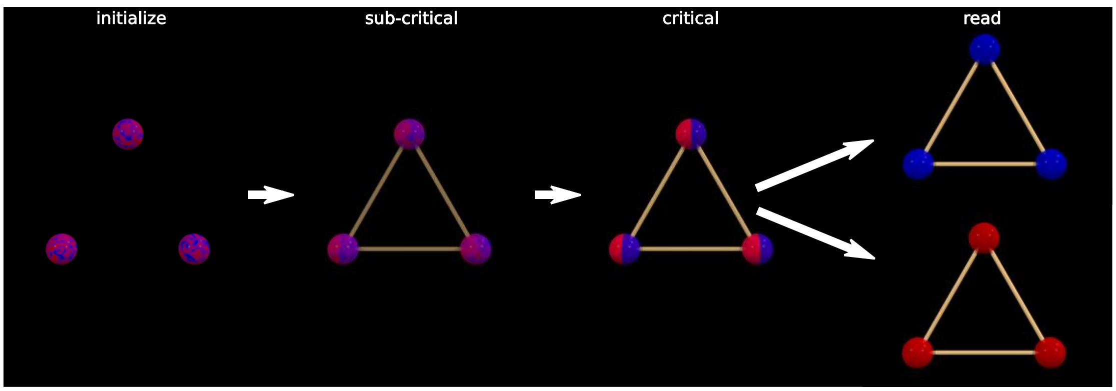

Figure 2. Qubits are represented by colored spheres and interactions by solid bars. QA begins with all qubits uncoupled and their states completely uncorrelated, as indicated by mixed colors (+1 = red, -1 = blue) on each sphere. As QA progresses, the interactions are slowly added and correlations between qubits become more apparent. At some critical point in the anneal, the states of the qubits become strongly correlated. In this simple example, there are two ground states corresponding to all-red (+1, +1, +1) or all-blue (-1, -1, -1). Upon approaching the critical point from the disordered phase, the system is in an equal superposition of the two ground states, indicated by spheres that are half red/half blue. In passing through the critical point, the system is forced to randomly choose to be in one of those ground states, which is subsequently read.

As illustrated in Figure 2, the method involves initializing the qubits in a quantum state that broadly explores all possible solutions, and then slowly ramping up the relative strength of the NP-hard problem couplings and biases. One iteration of this process is referred to as an *anneal*. At some *critical point* during the anneal (there may be more than one), the system can undergo a phase transition from a disordered quantum paramagnetic phase to an ordered classical magnetic phase. The final state that is measured after passing through the phase transition is a potential solution to the NP-hard problem. The key physics that one hopes to harness in QA is the formation of entangled quantum states with long-range correlations between individual qubits that persist during the phase transition, which enhances the probability of finding the system in a ground state at the end of the anneal.

2.2 The Quantum Machine Instruction

The quantum processing unit (QPU) of a D-Wave quantum computer carries out the annealing process described above. Like all quantum computing devices being built today, the QPU runs within a classical framework resembling the familiar von Neumann architecture containing an arithmetic logic unit (ALU), memory, I/O, and a control unit. The QPU can be thought of as an ALU-like quantum circuit containing an arrangement of qubits and pairwise couplers, combined with a specialized control system. The network of qubits and couplers is programmed using a powerful and flexible quantum machine instruction (QMI) that is defined by a set of parameters that specify the desired output and how to obtain it. The output consists of a vector S of spin values $S_i \in \{\pm 1\}$.

The most important component of the QMI is the specification of the desired output according to input vectors (h, J). These are inputs to the Ising model (IM) optimization problem: Given a graph $G = (V, E)$ with fields h_i on vertices V and interactions J_{ij} on edges E, find a spin vector S that minimizes the objective function

$$E(S) = \sum_{(i,j)\in E} J_{ij} S_i S_j + \sum_{i \in V} h_i S_i.$$

This problem is NP-hard when G is nonplanar. To invoke a QMI, the user provides the values for $h = \{h_i\}$ and $J = \{J_{ij}\}$, and an anneal time interval T. Furthermore, as with all real-world quantum computing today, there is a chance that physical limitations of the device or interference from external noise will have an adverse effect on the calculation. This means that the QPU does not guarantee to return a ground state solution every time; thus, it is prudent and cost-effective to repeat the anneal many times for each input. For this reason, the user must also specify R, the number of solutions to be returned.

A variety of annealing protocols are supported through additional parameters to the QMI, called anneal path features. For example, the anneal can be specified as a piecewise linear waveform that can be used to alter the evolution near a phase transition. It is also possible to adjust the individual qubit annealing schedules to a limited degree by specifying anneal offsets. The reverse anneal protocol allows the user to specify initial qubit values and explore nearby solutions.

Note that in contrast to GM, programming an annealing-based QPU does not involve writing out step-by-step instructions for accomplishing the task at hand. Instead, the user specifies the desired result—an optimal spin assignment for the objective function defined by inputs (h, J)—together with parameters specifying how to accomplish the result, and the quantum algorithm implemented in hardware does the rest. This indicates that the natural programming paradigm for QA is declarative rather than imperative in nature. Examples of declarative programming languages in classical computation include Prolog for logic programming and SQL for computations on databases.

The declarative paradigm makes possible a simple and scalable interface for using the QPU, which is an important component of D-Wave's efforts to make quantum computers available to a broad spectrum of potential users. The Leap™ Quantum Application Environment, which provides real-time cloud access (with limited free option available) to D-Wave systems, is a new piece of this long-term strategy. Visit www.dwavesys.com to learn more, and to find descriptions of the latest QMI variations and support tools (such as post-processing utilities and hybrid quantum-classical computing frameworks).

2.3 Workflows for Quantum-Problem Solving

The core operation of the QPU is to return a sample of R solutions to a given IM input (h, J). All NP-hard problems can be translated to IM using well-understood methods from NP-completeness theory, which means that in principle this core operation applies to those problems as well. Solving a given application problem P using the quantum processor involves a few discrete tasks, as described below and illustrated in Figure 3.

Translate inputs for P to inputs for IM or QUBO. The IM problem is defined above for spin values (-1, +1); the quadratic unconstrained binary optimization (QUBO) problem is equivalent but defined on binary values (0, 1). The QPU interface accepts either format. Cookbook methods are known for translating many NP-hard problems to IM or QUBO [9]. Note that some problems are more suitable for this approach than others, depending on how much input expansion is created by the translation.

Decompose or distill the problem. If a given input turns out to be too big to fit on the QPU, the input can be decomposed by breaking into pieces that are solved separately (note the piecewise solutions may not be part of an optimal solution to the full problem). This approach does have its limits, in the sense that a given problem instance may be less suitable if only a small part of the overall structure can be represented on the quantum hardware.

Minor-embed the input. An IM or QUBO input is defined by weights assigned to the vertices and edges of general graph $G = (V, E)$. These weights must be mapped to the qubits (vertices) and couplers (edges) inside a D-Wave QPU, which do not have general connectivity; instead, they have a so-called Chimera graph structure C. Mapping G onto an equivalent representation in C involves a process called minor-embedding. Tools for minor-embedding are available in the D-Wave software library, although problem-specific custom embedders can sometimes give better results.

Query the QPU. An input of suitable format and size can be sent to the QPU together with appropriate parameter settings; note that well-chosen parameter settings can sometimes dramatically improve solution quality. Besides this one-shot approach to problem solving, the QPU can be incorporated in a hybrid approach that performs a sequence of QPU queries interleaved with a classical computation that modifies (h, J) between queries. This approach is used, for example, during the training cycle in machine learning applications.

Return the results. The classical infrastructure for the QPU (optionally) post-processes the results and then maps embedded problems back to their unembedded form. Solutions in IM/QUBO representation must also be translated back to their original form.

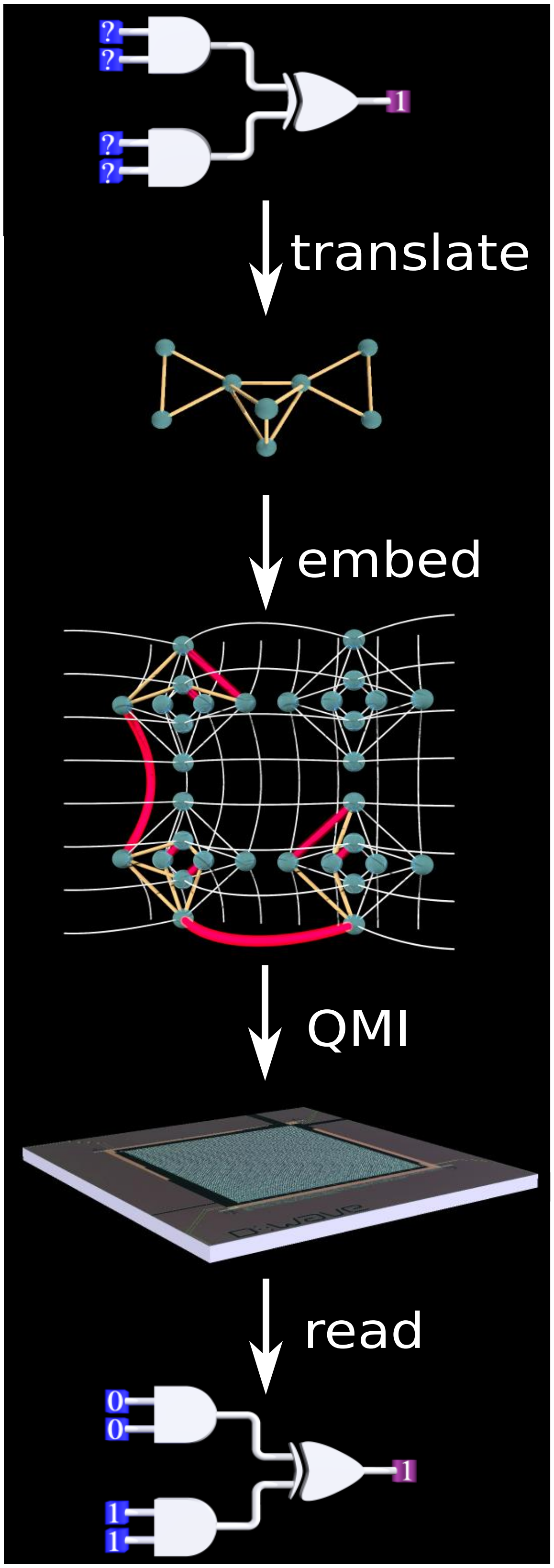

Figure 3. Steps in the problem-solving workflow. An input for the NP-hard *circuit-satisfaction problem* is a description of a logic circuit: the goal is to find an assignment of inputs that makes the circuit evaluate to 1. This input is translated to a graph representation for IM or QUBO (weights are not shown here); then it is minor-embedded onto a Chimera graph (the extra red arcs are created during embedding).

This representation of the input is sent to the QPU via the QMI; and the results are read and translated back to the original problem. Assuming noise and control errors are adequately suppressed, the sample of R results from the QPU contains a valid output for this input.

3 Applications and Performance

This section gives a brief summary of what is known about suitable applications and about performance of D-Wave processors to date. Note that this work is far from finished, and many questions remain unanswered.

3.1 Applications

Two recent meetings of the D-Wave user group (Qubits Europe 2018 and Qubits North America 2018) showcased the variety of early-phase applications that have been implemented on D-Wave QPUs [10]. Table 1 lists some applications described at those meetings that were run on either a D-Wave 2000Q (2000 qubits) or a D-Wave 2X (1000 qubits) (the last few are applications in machine learning). Overall, in recent years about 100 early applications developed by research institutions and commercial enterprises, ranging from proof-of-concept to nearly-industry-ready, have been demonstrated to run on D-Wave QPUs.

Table 1. Sample Applications for Quantum Annealing Processors

3-satisfiability	Fault tree analysis	Simulating atomic magnometers
3D image tomography	Job-shop scheduling	Simulating quantum lattice transitions
Bayesian inference in imaging	Linear least squares	Telecommunications network design
Binary matrix factorization	List order optimization	Topological data analysis
Budget pacing in auctions	Modeling molecular dynamics	Traffic flow optimization
Capacitated vehicle routing	Modeling terrorist networks	Tsunami evacuation routing
Chemical structure analysis	Optimizing factory vehicles	ML: accelerating deep learning
Computational hydrology	Phylogenetics	ML: classification
Constrained shortest paths	Portfolio selection	ML: quantum boosting
Election modeling	Satellite scheduling	ML: training neural networks
Factoring	Simulating KT phase transitions	ML: reinforcement learning
Fault diagnosis in networks	Simulating material structures	ML: unsupervised learning

Table 1 also illustrates the wide variety of use cases that are compatible with this approach. Some cases (e.g., factoring) require ground-state solutions; many (e.g., traffic flow optimization) require good solutions in short time frames; and some (e.g, portfolio selection) require diverse samples of solutions. As well, some applications in machine learning require samples drawn from a Boltzmann distribution on the solution set; and some applications in quantum simulation require distributions sampled from entangled quantum states mid-anneal. This latter case was made possible using anneal path features that have been recently developed.

3.2 Performance

While the expanding list of early-phase applications for QA is encouraging, its existence does not answer the most important question: Can annealing-based quantum processors outperform classical approaches for solving hard computational problems? Not surprisingly, this question is a subject of vigorous current research. Presentations at user group meetings, and papers published by researchers in academia and industry, describe work to evaluate D-Wave quantum systems under many definitions of performance. This body of work is very briefly summarized here.

First, complexity-theory versions of the question—evaluating abstract QA algorithms in terms of asymptotic worst-case performance—remain open. Note that empirical results cannot be used to settle open questions in complexity theory, since the latter contain universal quantifiers over infinite sets. As well, theoretical results tend to have little relevance to practice, since they assume noise-free computations and focus on worst-case performance bounds.

Second, a physics-based version of the question looks for a phenomenon known as quantum speedup, comparing QPU performance terms of runtime scaling on synthetic inputs, to classical heuristics, under best-case (optimally tuned) conditions [11]. An observation of "limited quantum speedup" has been reported, but the general question remains open [12]. Note that results about quantum speedup also tend to have little relevance to practice: the synthetic inputs do not resemble real-world inputs and the scaling analysis ignores constant-factor speedups, sometimes favoring the QPU by many orders of magnitude [13], that are of keen interest to the practitioner.

Third, from the practitioner's point of view, answering the question requires identifying applications and use cases for which the QPU—together with the classical framework supporting the workflow tasks described earlier—can outperform standard classical alternatives. Here, performance has many meanings, often involving combinations of speed, solution quality, cost, and energy consumption. In this arena, a handful of cases have been found for which current-sized quantum systems can outperform classical alternatives to a modest extent. Here are two examples from Qubits North America 2018:

- Ushijima-Mwesigwa et al. [14] looked at a graph partitioning problem arising in molecular dynamics simulation; they reported that a D-Wave 2000Q QPU, combined with the `qbsolv` decomposition tool [15], found solutions of equal and sometimes better quality than state-of-the-art classical approaches.
- Tanahashi et al. [16] reported that a similar decomposition method found better solutions faster than their industry-standard approach, when applied to a problem of finding optimal listing orders for online search results.

While these results are encouraging, they must be tempered by the knowledge that superior classical methods for solving those particular problems may someday be found. For that matter, better methods for using the QPU may also be found: it is a property of the NP complexity class that empirical work on these types of questions should never be considered finished. An industrial user of QC technology may prefer having the means in hand to quickly solve the problem, as opposed to spending time and money searching for better algorithms that may or may not be found.

A few general observations about performance of D-Wave quantum processors gleaned from the research literature and reports of users' experiences are listed below.

QPUs exhibit fast convergence to good solutions. A D-Wave QPU routinely returns near-optimal solutions within a few anneals, but then may require a significantly larger number of anneals to find at least one ground state. The reason for this phenomenon may be a combination of noise, physical limitations of the system, and finite precision representation of the IM parameters on-chip.

Size matters. If an input is too small or too easy, a classical solver using nanosecond-scale instruction sets can find solutions in times well below 10 ms, the time needed to set up the problem on a current-generation QPU. On the other hand, if the input is too large to fit on the QPU, it must be decomposed, as described earlier: doing so adds classical overhead time and may impact solution quality, and can therefore only be effective if raw performance on QPU-sized problems is differentiated enough to overcome that overhead.

This suggests the existence of a sweet spot with respect to input size: an ideal problem must be small enough such that substantial portions of it fit on current hardware but also big enough (and hard enough) that it cannot be solved quickly by purely classical means. Tests of smaller previous-generation QPUs on real-world application problems revealed very few cases that meet both criteria; however, as mentioned above, there is some evidence that the current 2000-qubit systems are becoming large enough to break even with or slightly outperform classical counterparts on problems involving finding near-optimal solutions.

Favorable phase transitions lead to better performance. As a condition for an annealing-based quantum computer to perform well, the input should have features that elicit the special capabilities of the QPU that are not available to classical solvers. This observation has implications for identifying input sets that are most suitable for demonstrations of quantum speedup, as well as for demonstrating substantial quantum performance advantage on wide-ranging applications. Much work is needed to identify large application inputs that exhibit such features, and to understand how to tune system parameters for best performance on those inputs. Of course, one prerequisite for this work is having QPUs large enough to study suitably-sized inputs.

Infrastructure is key. As discussed above, the quantum algorithm runs within a classical support framework that provides tools for transforming, decomposing, and embedding inputs. Basic tools for performing these tasks, as well as for other types of support, are available in D-Wave software libraries and GitHub repository [15,17]. However, there is much room for improvement, in terms of both performance and application scope.

Note that this discussion of performance refers to raw QPU performance, without the use of error mitigation or error correction schemes that would presumably be necessary for current versions of GM QPUs to achieve comparable results on similarly-sized inputs. This illustrates the observation in the introduction that QA is fundamentally more robust against noise than GM, an important consideration in D-Wave's early selection of this paradigm.

4 The Future

The current generation quantum systems manufactured by D-Wave are but a step on the path to realization of industrially-relevant quantum computing. Looking beyond that goal, an internal research program begun within the past couple years targets the longer-term objective of building fully-general quantum computers (called *universal quantum computers*, which have different properties than universal Turing machines).

While the amazing innovations mentioned in the beginning of this article may be considered strong motivation for making the attempt in the first place, predicting the feasibility and timing of major technology-driven societal changes is beyond our ken. However, based on ongoing empirical work and our current understanding of past and current D-Wave systems, we describe below some improvements to next-generation QA technologies we are confident will come to pass within the next two-to-five years.

Next-generation architectures will solve much larger problems. D-Wave is currently developing a new architecture (called Pegasus™) that will contain more qubits, more inter-qubit couplers, and higher coupler count per qubit (i.e., greater connectivity). The anticipated benefits for the end-user include the ability to represent larger inputs in hardware, to find better embeddings leading to better solution quality and higher ground-state probabilities, and to experience lower overhead cost for performing embeddings.

For these reasons, we believe that the next architecture will be able to show differentiated performance at solving a wider range of industry-relevant problems than is currently possible, and that this trend will continue in subsequent generations.

Error and noise suppression will keep pace with increasing qubit counts. We are not aware of any fundamental impediments to building ever-larger QA-based QPUs. However, increasing processor size and complexity is only worth doing if control precision and noise suppression can also be scaled up. D-Wave has a continuously running materials science program to reduce noise, which is closely linked to our mainstream QPU fabrication program. Every successive generation of D-Wave QPUs to date has made use of new technologies that improve control precision, and work is on track to do so again with our next-generation QPU.

Suppression of noise and control errors means that the near-optimal solutions found quickly by the QPU are generally better: that is, closer to optimal or more likely to be optimal. We believe that these innovations will broaden the set of use cases where demonstration of a quantum performance advantage may be possible.

New QMI parameters will lead to better performance and wider applicability. Much of the recent excitement surrounding D-Wave QPUs has been related to their use in the context of quantum simulation [18,19]. When used in this mode, the task for the QPU is not to solve a classical problem, rather it is to expose the physics of a given quantum system via measurement of qubit states at intermediate points during the anneal. Expanding this capability required development of new features and new QMI parameters; these features will be released to all users.

In the longer term, D-Wave scientists will continue to research means to build additional controls into the QPU that will enhance and expand upon current capabilities. (This includes modifications aimed at realizing a universal quantum computer, although at present we cannot predict precisely how that research timeline will unfold.)

We anticipate developing a better understanding of how to use these new capabilities by making them available to a diverse and rapidly growing user community. We believe that these features will lead to better ways to use the QPU to solve industrially relevant problems as well as performing quantum simulation experiments of interest to science.

Efficient bigger-than-chip solvers will be developed. We have found that many industrially relevant problems are bigger than near-term QPUs, and thus see an urgent need for development of hybrid classical-quantum solvers that work by decomposing inputs, as discussed in a previous section.

D-Wave Hybrid, a framework to support this development work, is available in the D-Wave GitHub repository [13]. Several prototype solvers exist that demonstrate the value of connecting industrial users to the QPU in this way [20] and more are under development. This is the subject of much current research both internally at D-Wave and by the growing user community, and we anticipate rapid improvements in the near term.

More-effective software infrastructure will broaden the range of applications. Discovery of better translation and embedding methods, introduction of new and better post-processing utilities, and automatic choice of QMI parameters, will be effective at changing that tipping point from "competes with" to "outperforms" classical alternatives for an ever-expanding range of application problems. Work on developing better classical support tools surrounding the QMI is ongoing, and better tools will continue to be made available.

While many real-world use cases will be able to find quantum advantage with loose integration of classical and quantum processing components (i.e. with a small number of QPU queries per problem), many applications will need tighter integration. In particular, effective hybrid solvers will require low-latency couplings between the classical and quantum portions, leading to geographic co-location of classical and quantum processing components, with appropriate security and scheduling mechanisms in place. D-Wave is working on developing new techniques to support tighter integration of classical/quantum processors, which should yield improvements in overall workflow efficiency.

Hybrid solution methods will outperform purely classical or quantum methods in some cases. Together with improvements in the size and performance of the QPU, increased flexibility of the QMI, and more efficient infrastructure support, we anticipate an upsurge in the development of efficient and effective hybrid algorithms that interleave quantum and classical computations. These hybrid solutions will combine the best attributes of quantum and classical processors: finding good solutions fast and transforming good solutions into optimal ones, respectively.

Very preliminary work on these approaches suggests that they show promise to deliver differentiated performance as quantum systems continue to grow and improve.

5 Concluding Remarks

Of all the quantum computing platforms currently under development, in our view annealing-based quantum computing offers the most viable way forward to connect quantum hardware to real-world applications. Providing annealing-based QPUs as part of a complete computing platform within a declarative paradigm has made this nascent technology accessible for a large number of users who, in turn, are helping to drive this approach forward. The development of a QA user ecosystem can be directly attributed to D-Wave's strategic decision to develop a technological model that puts applications performance first, and that prioritizes demonstrations of usefulness for tackling real-world problems.

The D-Wave technology development program works in a tight feedback loop with practitioners and users, which provides a foundation upon which to build next-generation systems. Because of this approach, we see a clear path forward to improving performance, which we believe will allow us to demonstrate broader industrially-relevant performance advantages over classical computing within the next few years.

References

1. A. Ambainis. What can we do with a quantum computer? Institute for Advanced Studies Ideas, 2014, www.ias.edu/ideas/2014/ambainis-quantum-computing Accessed November 30, 2018.
2. A. Beall and M. Reynolds. What are quantum computers and how do they work? WIRED explains, *Wired Magazine* 16 February 2018.
3. W. Knight. Serious quantum computers are finally here. What are we going to do with them*? MIT Technology Review*, 21 February 2018.
4. M. Dyakonov. The case against quantum computing, *IEEE Spectrum*, 15 November 2018.
5. A. Mizel, D. A. Lidar, and M. Mitchell, Simple proof of equivalence between adiabatic quantum computation and the circuit model, *Phys. Rev. Lett*. 99, 070502, 16 August 2007.
6. There is some debate as to whether a quantum annealer is properly called a ``quantum computer.'' Our usage is based on the fact that it exploits quantum effects in its computations, and that in principle it can compute any function that a Turing machine can. The variety of applications that fit this solution approach is discussed in the paper.
7. S. Tanaka, R. Tamura, B. K. Chakrabarti. *Quantum Spin Glasses, Annealing, and Computation*. Cambridge University Press, 2017.
8. T. Albash and D. Lidar, Adiabatic quantum computation, *Rev. Mod. Phys*. 90, 015002, 29 January 2018.
9. A. Lucas, Ising formulations of many NP problems, *Frontiers in Physics*, 2, 5, 2014.
10. Qubits Europe 2018: www.dwavesys.com/qubits-europe-2018. Qubits North America 2018: www.dwavesys.com/qubits-north-america-2018.
11. J. Job and D. A. Lidar. Test-driving 1000 qubits, *Quantum Science and Technology*, 3 020501 2018.
12. T. Albash and D. A. Lidar. Demonstration of a scaling advantage for a quantum annealer over simulated annealing, *Physical Review X*, 8, 031016, 2018.

13. V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, H. Neven. What is the computational value of finite range tunneling? *Physical Review X*, 6, 031015, 2016.
14. H. Ushijima-Mwesigwa, C. F. A Negre, S. M. Mniszewski. Graph partitioning using quantum annealing on the D-Wave system. Presented at Qubits North America 2018 (ref. 10).
15. D-Wave Systems GitHub repository. github.com/dwavesystems. Links to qbsolv, dwave-hybrid, and other hybrid solvers may be found on the main page. Accessed 3 December 2018.
16. K. Tanahashi et al., Global mutual information based feature selection by quantum annealing. Presented at Qubits Europe 2018 (ref 10).
17. Documentation on D-Wave programming tools may be found at www.docs.ocean.dwavesys.com.
18. R. Harris, Y. Sato, A. J. Berkley, M. Reis, F. Altomare, M. H. Amin, K. Boothby, P. Bunyk, C. Deng, C. Enderud, S. Huang, E. oskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, T. Lanting, R. Li, T. Medina, R. Molavi, R. Neufeld, T. Oh, I. Pavlov, I. Perminov, G. Poulin-Lamarre, C. Rich, A. Smirnov, L. Swenson, N. Tsai, M. Volkmann, J. Whittaker, J. Yao. Phase transitions in a programmable quantum spin glass simulator, *Science*, V361 I 6398, pp. 162-165, 2018.
19. A. D. King, J. Carrasquilla, J. Raymond, I. Ozfidan, E. Andriyash, A. Berkley, M. Reis, T. Lanting, R. Harris, F. Altomare, K. Boothby, P. I. Bunyk, C. Enderud, A. Fréchette, E. Hoskinson, T. Oh, G. Poulin-Lamarre, C. Rich, Y. Sato, A. Y. Smirnov, L. J. Swenson, M. H. Volkmann, J. Whittaker, J. Yao, E. Ladizinsky, M. W. Johnson, J. Hilton, M. H. Amin. Observation of topological phenomena in a programmable lattice of 1,800 qubits, *Nature* 560, pp.456-460, 2018.
20. G. Rosenberg, M. Vazifeh, B. Woods, E. Haber. Building an iterative heuristic solver for a quantum annealer, *Computational Optimization and Applications*, V 65, I 3, December 2019, pp. 845-869.