REPORT

Demystifying quantum computing for commercial companies

dunnhumby

> If you're a tech leader, it can be pretty hard to know what to make of quantum computing. How do you distinguish hype from reality? Is quantum computing a likely flash in the pan that you can safely ignore, or an extinction event for companies who don't start preparing now? What sort of team do you need to put together, and how do they get started? How much will it cost, and what will the benefits be?
>
> In this article we'll walk you through the lessons we've learned in our role as a commercial data science company approaching quantum computing. We'll answer the above questions and more, and help clarify the busy, ever-changing world of quantum computing and what it could do for you.

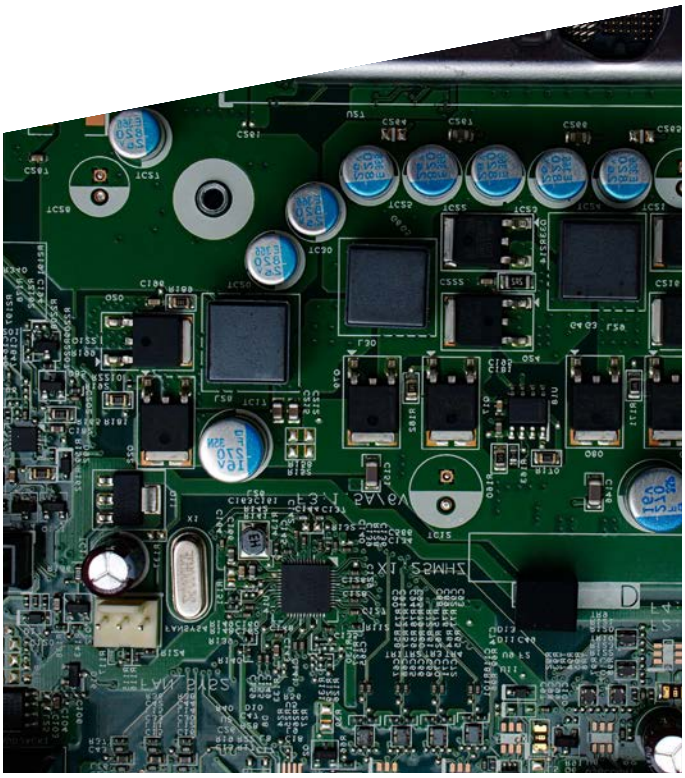
Why now?

Given quantum computing (QC) has been an academic field for nearly forty years, one of the best questions to ask is "why am I hearing about this now?". While it's true that the burgeoning quantum hardware ecosystem has seen unprecedented investment and expansion over several years, it is the recent advent of Quantum-as-a-Service (QaaS) that has been the trigger for many 'regular' companies to sit up and take notice.

QaaS has been gathering pace in the past 12 months. IBM was the first company to offer cloud access to quantum computers (their own devices in that case). More recently, in 2020, **Amazon BraKet** was announced, with Microsoft following soon after with **Azure Quantum**. Google – owner of possibly the world's most powerful quantum computer – also has a **cloud offering**, though it is currently open to selected partners only.

Cloud access to third-party quantum devices marks the start of a democratisation of quantum computers, with the best applications likely to come after these devices become available to an audience that extends beyond just physicists. The cloud providers are aware that the average company is not overflowing with QC experts, nor well-enough connected to QC start-ups to negotiate access to nascent quantum computers.

As a result, many QaaS vendors are striving to provide a full development environment for learning, exploring, simulating, and ultimately executing quantum computation. This is great news for your company, but also for your competitors. To make sure you're not left behind as quantum devices continue to evolve and become more powerful, you need to begin considering how to harness the power of quantum computation now.



How seriously should I take it?

One of the hardest things for a tech leader to do when diving into quantum computing is looking beyond the hype and working out how important it could be to your business. Can you sidestep the need to develop a detailed understanding of quantum computers and the current state of the possible under the assumption that if Amazon, Microsoft, and Google are getting involved then everything must be fairly legitimate? It's certainly tempting.

One cautionary comparison, though, might be the separate field of driverless cars: a lot has been promised within certain timeframes, and while impressive technical feats have been demonstrated, we still seem to be some way from fulfilling the early promises. This is also a field in which some of the tech giants play – so it bears asking whether quantum might be on the same trajectory.

The answer to that question is best broken down into two parts:

i. understanding what the different types of quantum computer are, and what their limitations might be, and;

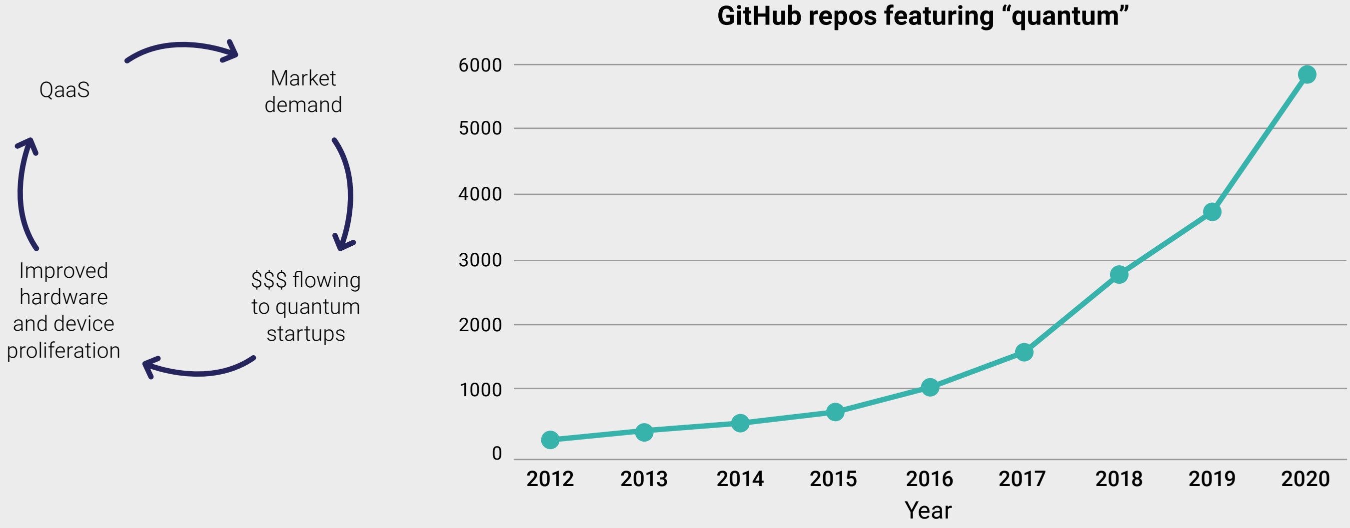
ii. making some informed judgements on how quickly things will evolve.

We (attempt to) answer these key questions later in this paper.

We will see that many things – the hardware itself, programming languages, algorithms, and even the metrics for judging progress – are already changing very quickly. That pace of change is unlikely to slow any time soon. The era of QaaS may herald the start of a virtuous circle leading to rapid market evolution, as QaaS stimulates demand, leading to money flowing to quantum startups which can in turn be invested into hardware improvements and capacity expansions. In addition, the entry of Amazon, Microsoft, and others into the QaaS market is likely to mean that competition for cloud business will stimulate improvements for the end-user.

So, if you are going to get into QC, what do you need to know? Let's start at the beginning.

Figure 1: Does the advent of QaaS herald a virtuous circle of investment and device improvement?



What is quantum computing?

The easiest way to explain QC is by comparing it to classical computation. You're probably aware that in a normal computer we represent information using bits. A bit can be 0 or 1, but – importantly – not both at the same time. A sequence of bits gives us a binary number, e.g., the binary number 1101, which represents the decimal number 13.

Binary numbers are stored in a computer memory location called a register. Registers can vary in size. The binary number 1101, for example, would be stored in a 4-bit register. Applying different mathematical operations to these registers allows us to build up more complicated calculations and ultimately construct the software which we run on our laptop and desktop machines.

Underneath all of this is an actual physical system – such as the electric charge level in a RAM chip, or the direction of magnetisation on a disk – that allows us to physically store a bit. Bits can be stored either temporarily (as is the case with RAM) or permanently, usually on a disk. So far, so good – but where does quantum come into this?

At small scales and extremes of temperature, the real physical world follows the laws of quantum mechanics. Quantum mechanics is a fascinating and sprawling subject, and one that we could easily devote the rest of this paper to. The most important thing to know about quantum mechanics in relation to computing, however, is that quantum particles can exhibit some fantastically strange and counterintuitive properties that can be exploited for information processing.

The quantum equivalent of a bit is known as a qubit. Like the 0 and 1 states of the bit, a qubit has two states, commonly denoted by the symbols $|0\rangle$ and $|1\rangle$. Unlike bits, however, a qubit – written as $|q\rangle$ – can exist as any combination (superposition) of these two states. We write this as, $|q\rangle = \alpha|0\rangle + \beta|1\rangle$. This is the qubit equivalent of **Schrödinger's famous cat** that is a superposition of 'alive' and 'dead'.

As with classical computation, we can build up registers of qubits, and apply mathematical operations to these quantum registers.

Let's take a 4-qubit register as an example: $|q_4\, q_3\, q_2\, q_1\rangle$. Since this 4-qubit quantum register can represent a superposition of all the $2^4 = 16$ possible 4-bit binary numbers, when we apply a mathematical operation to this quantum register, what we get back is a single (quantum) register that encapsulates the results of applying the calculation to all 16 possible 4-bit binary numbers. Similarly, a mathematical operation applied to an 8-qubit register encapsulates the results of the 2^8, or 256 classical calculations. A 64-qubit register encapsulates the results of $2^{64} \cong 1.84 \times 10^{19}$ classical calculations.

While these quantum registers are relatively small compared to the size of the classical registers available in a standard laptop, note we can manipulate mathematical calculations at tremendous scale. Quantum computing takes us way beyond anything that any classical computer can currently do, or is likely to be able to do even in the future – no matter how state-of-the-art it may be. This is what is referred to as **quantum supremacy**.

It is important to emphasise at this point that quantum computation is not doing parallel computation. We state above that a mathematical operation applied to an N-qubit quantum register returns an object that 'encapsulates' the results of applying the mathematical operation to all the possible N-bit binary numbers. However, when we look at or 'measure' the contents of the object returned by the quantum calculation, we see only the result of the mathematical calculation applied to a single N-bit binary number. Which of those 2^N calculations gets returned by the 'measurement' is random.

This is the equivalent of opening the box in Schrödinger's cat experiment – once we open the box, we don't see a cat that is a superposition of 'alive' and 'dead'. Instead, obviously, we see either a cat that is alive, or one that is dead. The goal of a quantum computation program is to perform operations on the quantum registers in such a way that when we measure the result, we get – with a very high probability – values that are more useful to us. This is the primary skillset required of a quantum computer programmer – to design such a sequence of operations.

The computational scale offered by QC is clearly vast, but the question remains: how could QC help your company? In the following sections we'll look at the current quantum landscape – how it differs from the hype you may have heard, and what exactly the current generation of QaaS devices are good for.

$$|q\rangle = \alpha|0\rangle + \beta|1\rangle$$

What can I use quantum computing for?

Firstly, let's state what quantum computers are not. They are **not a next generation of faster, smaller, more powerful 'regular' computers. They are significantly different beasts.** It is unlikely that quantum-based machines will ever displace classical computers. Instead, they will excel at a subset of very specific problems.

What is that subset of problems?

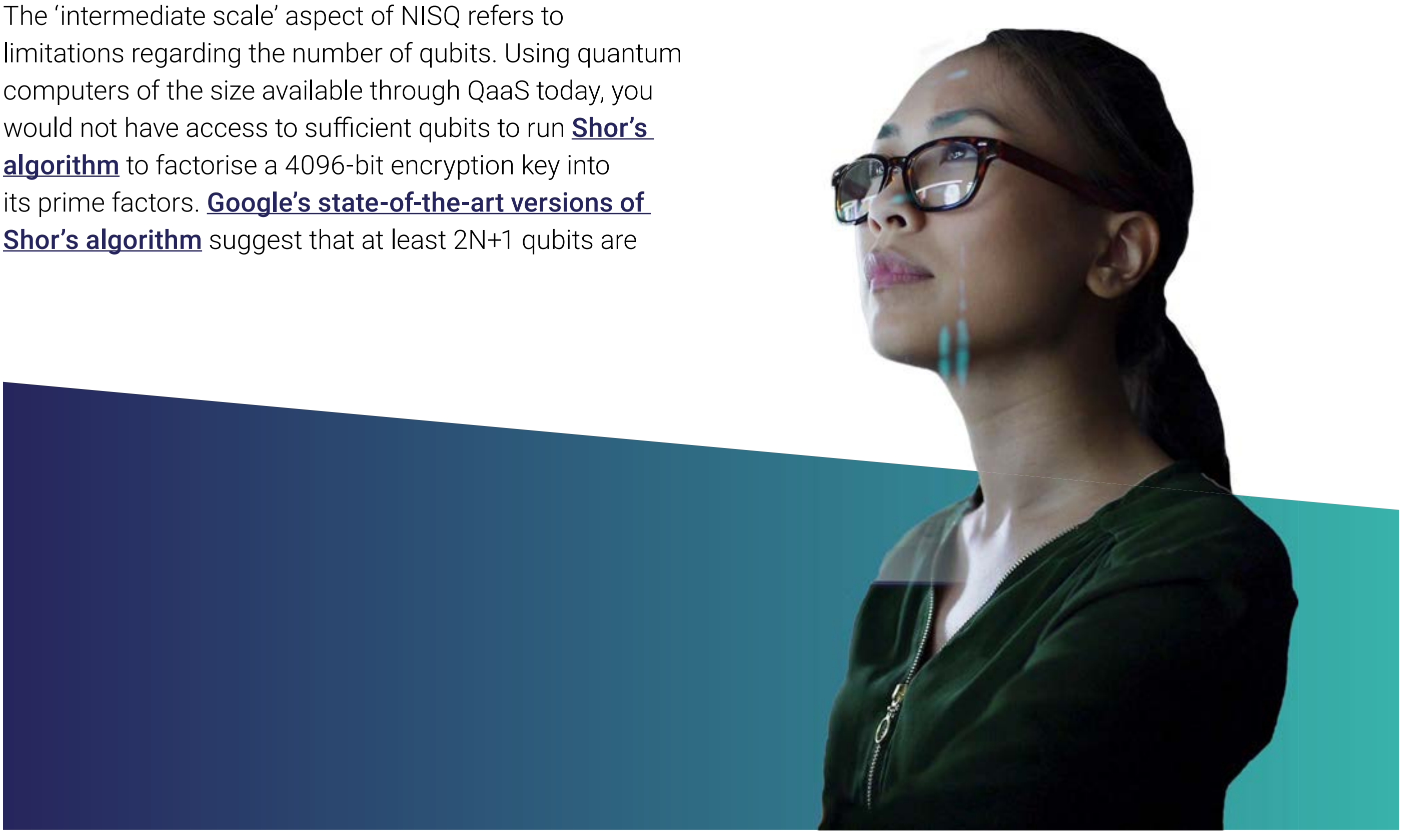
You may have heard or read that, in the future, quantum computers will be used to break the encryption systems that currently underpin much of the world's digital communications and commerce. That may be true, but the key words to bear in mind there are "in the future". We're not there yet.

The limitations of the current generation of quantum hardware are best summarised by the now-famous acronym **NISQ: Noisy Intermediate Scale Quantum.** As discussed above, today's quantum devices can perform a series of manipulations on a qubit register to carry out a calculation. The 'noisy' part of NISQ means these gate operations (amongst other things) are not perfect. Because of this, there are a limited number of computational steps one can deploy before the fragile coherence of the quantum register – and thus any hope your calculation returns a good answer – is irreparably degraded. Think of NISQ's 'N' as limiting the length of a computation you can perform.

The 'intermediate scale' aspect of NISQ refers to limitations regarding the number of qubits. Using quantum computers of the size available through QaaS today, you would not have access to sufficient qubits to run **Shor's algorithm** to factorise a 4096-bit encryption key into its prime factors. **Google's state-of-the-art versions of Shor's algorithm** suggest that at least 2N+1 qubits are required to factorise an N-bit number, meaning that at least 8193 qubits would be required to factor a 4096-bit key. At time of writing, the largest quantum computer (by number of qubits) available via either the Microsoft or Amazon QaaS offerings has 32 qubits.

So if we can't use current QaaS hardware for breaking encryption, and we are limited for the time being to intermediate scale problems, what is QC good for? Our explanation above of 'What is QC?' provides a hint. QC will be good for challenges such as high-dimensional discrete optimisation problems, where solving them classically would require us to perform numerous function evaluations. With QC, we can instead take advantage of superposition and entanglement to enhance the probability of finding states that are interesting solutions to our problem.

Given that all commercial organisations seek to refine their operations and maximise profits, it is no surprise that optimisation is one of the main potential applications of QC in retail. Indeed, it is the possibility of exploiting near-term quantum advantage in the many optimisation problems that arise in dunnhumby's customer data science that piqued our interest in QC.



> “At around fifty qubits one famously hits ‘quantum supremacy’ - the point at which a classical computer cannot match the performance of a quantum counterpart.”

We have used a quantum algorithm to determine the optimal set of products to include in a promotional campaign. This requires selecting products on the basis of minimizing the sales that one product will cannibalize from another. One also has to choose a promotional strategy for each product - advertising and display mechanisms - to maximize the profit. For a best-in-class grocery retailer these optimization tasks are central to the profitable management of a product category, whilst for the Data Science teams within those retailers, they represent a significant high-dimensional, highly interacting, constrained optimization challenge.

Another hard combinatorial retail problem we have explored quantum solutions to is that of optimizing store arrangement and the allocation of shelf space to products. Like the promotional strategy example, this is a constrained problem with many interacting components that can affect revenue, including: product position on shelf, proximity of complementary products, number of product facings, and shelf location in store. Current solutions are limited in scope, as only a limited number of products can be tackled at a time. The potential paradigm shift as promised by QC may enable data scientists to tackle the richer, higher value, problem of optimizing over an entire store at once.

It is not just dunnhumby that is exploring the potential for QC to solve high-dimensional commercial optimization problems. Quantum computing has been applied to problems as diverse as financial portfolio optimisation, used by the likes of **JP Morgan and Goldman-Sachs**, to **traffic light arrangement** studied by Toyota. Again, **as QC expert Scott Aaronson explains on his blog**, we want to emphasise that QC is good at solving these high-dimensional optimisation problems not because it is doing any sort of naïve parallel computation and searching for a needle in a high-dimensional haystack, but because the QC algorithm is able to exploit some global characteristic of the problem to enhance the chances of returning the best solution.

What is the current state-of-the-possible?

One key dichotomy for tech leaders to consider is the difference between current efforts to achieve quantum advantage for specific problems with small, noisy (but rapidly improving) NISQ hardware, and the ultimate goal of massive (millions of qubits), fault-tolerant (immune to noise) quantum computers. The latter of those realities is likely at least a decade away, and we do not address it here. What we are concerned with in this paper is the evolution of the NISQ QaaS landscape as it stands today.

We've talked about "rapidly improving" NISQ hardware, but how does one measure the power of a quantum computer and justify the claim that things are improving? Some metrics are easier to grasp than others, with the number of qubits being the easiest. For every qubit you add to a quantum computer, for example, the size of the regular classical computer needed to match its performance it **doubles**. At around fifty qubits one famously hits 'quantum supremacy' – the point at which a classical computer cannot match its performance (in certain tasks, at least).

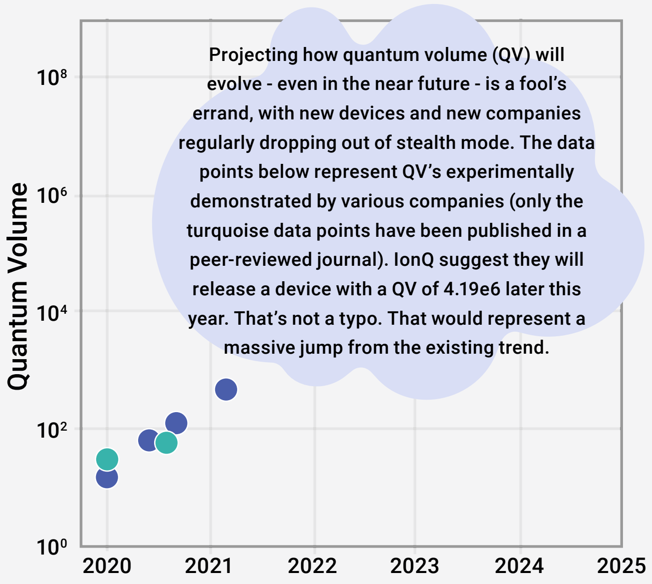
The raw number of qubits is not the final word, however. You could have thousands of qubits, but your computer might still be useless if the coherence of the qubit register cannot be maintained across the duration of time required to run your program. This is where the concept of noise comes in, often quantified through a statement of the *error rate*. For instance: the higher the error rate of two-qubit gates in a quantum device, the fewer computational steps you can apply to your qubit register before your calculation is terminally degraded. Qubit connectivity also has a crucial effect on performance.

It is even possible to roll metrics like the number of qubits, error rate, and qubit connectivity into a higher-level metric known as the quantum volume (QV), an attempt at creating a one-stop shop for quantifying device power. Things are changing so fast, however, that even the metrics for judging the computational power of a quantum device are evolving. This makes it even harder for the quantum newcomer to understand the lay of the land.

Recently, IonQ have taken an additional step in defining "algorithmic qubits" – logical qubits protected by error correction. Each logical qubit is realised by a larger number of physical qubits, trading the number of qubits for reduced error. This concept of an algorithmic qubit somewhat replaces quantum volume as the layperson's go-to metric for parsing the power of a device.

As **Peter Chapman, CEO of IonQ**, notes, "with better quantum computers, the QV metric will become unusable because the numbers grow so quickly. We foresee a time in the near future when QV numbers will grow so large they won't fit on your screen." The number of algorithmic qubits is logarithmically related to the quantum volume (doubling QV is equivalent to adding one algorithmic qubit). Having to change a metric to a logarithmic version is a fairly dramatic indication of the speed of progress!

Figure 2: Recent trends in the performance of quantum computers



Projecting how quantum volume (QV) will evolve - even in the near future - is a fool's errand, with new devices and new companies regularly dropping out of stealth mode. The data points below represent QV's experimentally demonstrated by various companies (only the turquoise data points have been published in a peer-reviewed journal). IonQ suggest they will release a device with a QV of 4.19e6 later this year. That's not a typo. That would represent a massive jump from the existing trend.

Using these metrics, we can at least begin to parse which of the competing NISQ devices available via QaaS are the 'best'. But what do they tell us about what one can actually achieve with a nascent, commercially available quantum computer? Are you going to be able to find better routing solutions for your fleet of 2,000 delivery vans using QaaS today?

Well, the answer to that question, to varying levels of detail is:

i. **No.**

ii. **No, probably not right now unless you are cleverer than everyone else.**

iii. **No, probably not right now unless you are cleverer than everyone else – but things are changing very quickly, so who's to say that practical quantum advantage isn't just round the corner in a few years?**

Facetious though it might be, this sequence is meant to highlight three things. At time of writing, QaaS devices are not good enough to give obvious quantum advantage for real business optimisation tasks. Rather, they are mostly used to demonstrate clever ideas that people have for attacking small, related problems. As a result, current QaaS work is concerned more with tailoring of algorithms to specific problems rather than attacking broad quantum advantage.

That said, it is by no means impossible that either some super-smart person will think of a killer algorithm that will cause an immediate paradigm shift for certain problems, or that with rapidly improving hardware and startups coming out of stealth mode, someone will drop a major improvement that makes quantum advantage suddenly achievable.

It is also worth calling out what we have not discussed so far – data. The problems and examples we've used to date in order to illustrate the potential and advantages of QC are largely mathematical calculations. The input data for these problems is small – an integer here, to represent the size of the problem, perhaps a small-scale matrix there, stored as a 2D-array, to represent some interaction costs.

As data scientists we are used to processing big data, building statistical and machine learning models. It is only natural to ask if quantum computers can be applied to machine learning. **The short answer is yes**, but caution should be exercised. Whilst **quantum machine learning** is a growing field, with machine learning frameworks such as **TensorFlow having a quantum offering**, we have yet to see what we consider to be any killer applications. Even many of its **leading proponents supposedly suggest being wary of the hype**.

In part, this may be because the focus of research in QC to date has been on computation, not on end-to-end data-processing workflows. To use QC for processing of data requires the **loading of classical data into quantum registers**, and hence a whole new set of quantum circuits and associated qubits. Likewise, extracting the results of any quantum data processing back into classical registers would also introduce additional overhead. As yet, this is a less explored, or at least less discussed aspect of QC.

Finally, while this article is chiefly concerned with gate-based quantum computers, we should highlight that two distinct types of hardware are available via QaaS: gate-based devices and quantum annealers. The fully programmable gate-based computers fulfil the picture we painted in "What is QC?", with a qubit register that one can apply a series of gate operations to in order to manipulate and progress the processing of information. Quantum annealers physically realise a system with an energy landscape that can represent a mathematical problem. Rather than applying gates to specific qubits, the whole system continuously evolves in an effort to reach the lowest energy state, representing, for example, the best solution to an optimisation problem.

The key message is that although you won't experience a paradigm shift in the speed and quality of technical solutions today – and no one can say exactly when that day will arrive – it is approaching. And given the complexity of the subject matter, you may need to start planning for that day now.

Crucially, that means answering questions like...

How does my company get started?

Given the current state of quantum hardware and the anticipated roadmaps into the future, what can we say about a quantum workplan at commercial companies today? The best advice we can give is that now is the time to be in learning and exploration mode, gaining technical expertise and developing informed opinions in preparation for the time when quantum advantage for business applications is robustly demonstrated.

For those who wish to start down that path, there is likely a resourcing issue to take into consideration too. Constructing a quantum program requires a different way of thinking about computations. As a result, in the short-term at least, the practical utilisation of quantum computation within industry may be limited by the availability of developers familiar with quantum programming concepts and paradigms.

The good news is that there has never been a better time to start learning about quantum computation. The cloud providers are striving to open up quantum for everyone and, clearly aware that many will be newcomers, are providing a realm of introductory material and example code for implementing quantum algorithms. A good example is Microsoft's **Quantum Katas**, a series of notebooks through which one can learn key concepts and literal code.

When it comes to applying quantum computing to business problems, what do you need to think about? Figure 3 demonstrates one way to break down the elements involved, based on our experience of approaching combinatorial optimisation problems that occur in a retail grocery context.

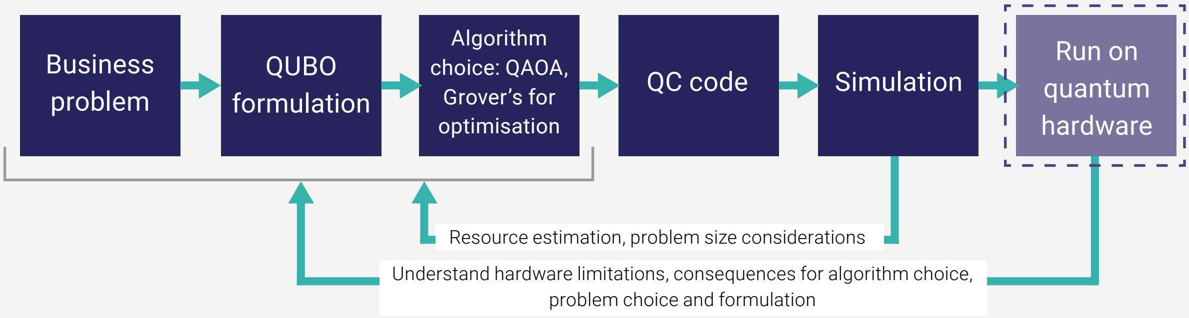


Figure 3: How to think about quantum approaches to business problems. The workflow can become very rich when these elements interact with each other. For example, understanding how choices in problem and formulation, broad-level algorithm-selection, and algorithm tailoring interact with each other to impact resource scaling and performance in the face of imperfect hardware.

The elements to consider:

- How to map a business problem into a form suitable for tackling on a quantum device. This can involve encoding constrained combinatorial optimisation problems in a constraint-free form, including Quadratic Unconstrained Binary Optimisation (QUBO) formulations.
- Understanding and choosing quantum algorithms suitable for NISQ hardware.
- Gaining experience programming for quantum hardware. Both explicit experience in specific languages, and, conceptually, how to think when writing quantum programs.
- Simulation / execution, and what you can hope to learn from this, including consideration of classical vs. quantum performance and mid-term scope for non-trivial achievements / quantum advantage.

Even selecting the right problem is a non-trivial exercise, as discussed in **this paper** by Chancellor et al. Many business optimisation problems can be considered in the form of a network of nodes and edges. How closely the underlying structure of this network matches the qubit connectivity – a hardware specific characteristic – can have important consequences for how many qubits you will need to execute a calculation. More importantly, you need to pick problems that are currently intractable at a certain scale, and that will really benefit from the potential paradigm shift offered by QC.

Once you have your problem, you also need to ensure that it is expressed in a suitable mathematical form. We previously discussed how quantum computers are not general-purpose supercomputers, but instead excel at certain tasks. One problem formulation which can be attacked by a quantum computer is that of QUBO, and so if a constrained optimisation problem can be mapped into the QUBO form then it can also be addressed by QC.

Once you have your optimisation problem in the right mathematical form, there are some big decisions to make – and this is where it gets hard for the quantum novice. Gate-based computation or quantum annealing? If gate-based, which algorithm to use? A huge amount of understanding and expertise is needed to answer questions like this, as well as how the choices you make will interact with hardware limitations (system size, error rates, qubit connectivity) and problem you're seeking to resolve.

As well as parsing the different approaches that exist, one also has to consider the extent to which quantum algorithms for NISQ hardware benefit from problem-specific tailoring. The best QC industry teams in the world today are publishing papers that demonstrate how they pick a problem, map it to QUBO form, simulate solving a small problem instance with an original tailoring of a quantum algorithm, and finally demonstrate it running on a quantum device via QaaS.

Whatever your choice of algorithm, you will need to code it up and represent the quantum circuit employed to solve the problem as a program. The state of programming languages for quantum hardware serves as a nice microcosm of the embryonic, still-settling state of the wider quantum computing world: there are many choices, and each of the big cloud providers is pushing their particular flavour. In time, a champion will likely emerge. Even for a specific language, rapid changes in syntax and libraries can mean that frequent updates to code are required. That said, we have not found learning specific languages to be a major overhead – it is a second-order concern compared with understanding quantum algorithms.

Finally, one has to consider execution on real quantum hardware. And how do you get a quantum computer? You don't: it's the era of QaaS. But how much is it going to cost? Amazon BraKet have a **nicely explained price breakdown**. And while figures like $0.01 per shot (one execution of a specific quantum program) may seem low, bear in mind in that for some of the algorithms and approaches for NISQ hardware you may need to run a program thousands of times.

The arrival of quantum advantage for real business problems won't just mark an inflection point in hardware development. It will also represent the turning point in the mode of working for companies. When that happens, we will move from the exploration and education that we see today into the real integration of quantum advantage into technical solutions. We might not have a date to aim for, but there's no doubt that waiting for the demonstration of quantum advantage is risky: you need to start preparing today.

Quo Vadis quantum computation?

If you do dip your toe into the QC waters, you can do so with the expectation that things are going to evolve rapidly. But where to, precisely?

Trying to predict what will happen with any emerging field is always a difficult and almost foolhardy task. The precise timing and nature of any advances is hard to say. With that in mind, our aim here is to speculate on possible themes that could emerge over the next few years, and to provide food for thought. Our opinions are based solely on the perspective of where we stand today and we would advise you to treat them as such.

We can break our speculations down into five natural sub-questions:

i. **What advances in QC hardware will we see?**
ii. **What advances in QC software will we see?**
iii. **What advances in QC algorithms will we see?**
iv. **What advances in QC applications will we see?**
v. **What advances in QaaS will we see?**

Let's unpack each of these in turn.

What advances in QC hardware will we see?

There are two, perhaps uncontroversial, statements we can make about where the QC hardware field will go.

In order of certainty, these are:

i. We will see significant revenue and capital investment inflows into the various QC hardware providers. This will be a result of the burgeoning QaaS market, and as investors spy the opportunity to invest in the 'next big thing'.

ii. We will see quantum hardware continue to improve, both in terms of numbers of qubits and reduction in problems caused by noise (both with error rates falling and error correction schemes that can realise logical qubits from a larger number of physical qubits). Many of the most-visible quantum hardware providers today such as IBM and IonQ have published roadmaps showing exactly how they expect their devices to improve over the next five years.

More speculatively, over the medium-to-long term, we would also expect to see consolidation in the QC hardware provider market – a point also echoed by McQuarrie et al in their recent review of the commercial aspects of QC. This is the natural evolution of any market. As some approaches to physically representing qubits prove to be difficult to scale, those QC hardware providers will cease to trade. Alternatively, acquisition of QC hardware start-ups by more established tech companies and/or other QC hardware providers will naturally lead to consolidation.

Recently the industry has seen some dramatic movement, with electronics giant Honeywell to spin out Honeywell Quantum Solutions - a leader in quantum hardware which will then merge with Cambridge Quantum Computing (CQC). CQC bring complementary expertise as one of the most established players in quantum software and algorithm development. The new combined company (at time of writing as yet unnamed) will be a powerhouse in the quantum industry landscape, not to mention an attractive investment opportunity.

In the short-term, there may be growth in the QC hardware market with cash inflows into the field attracting new startups. We do not expect this expansion to be large – creating a quantum computer is capital intensive, requiring highly specialised skillsets that are currently in short supply. 'First-mover advantage' is a strong differentiator, however, and so we would anticipate that new entrants to the hardware market will attempt to distinguish themselves by using less common physical representations of qubits such as topological based qubits that use the fractional quantum Hall effect, or qubits based on Majorana fermions.

What advances in QC software will we see?

The successful proof-of-concept demonstration of quantum computers, such as D-Wave selling its first commercial quantum annealer in 2011, led to an explosion in associated QC software vendors, a point highlighted in the **recent review by McQuarrie et al**.

This is a trend we expect to continue, for several reasons:

i. Unlike QC hardware production, QC software production is considerably less capital intensive.

ii. The opening up of a QaaS market will allow a greater number of QC software start-ups to enter the market, without the need to have a tie-up or special relationship with a QC hardware provider.

iii. There is a large number of potential domains to which QC software solutions can be applied, and so, in contrast to the QC hardware market, the potential software market space – within which solution vendors can differentiate themselves – is bigger. The ever-widening set of application domains will act as a pull on the QC software market.

iv. As appropriate high-level QC language abstractions are created, it will be easier to scale the QC software skill-base compared to scaling the QC hardware skill-base. Existing software developers, data scientists and machine-learning engineers can adapt to QC with support from the appropriate language tools. In contrast, individuals entering the QC hardware domain will need to acquire a reasonable understanding of quantum mechanics.

As we alluded to above, an inevitable part of that market pull will be QC software providers offering higher-level language abstractions, a necessity if the QC field is to scale. Currently, there is a rich ecology of QC languages – recently surveyed by **Heim et al**. These range from Microsoft's **Q#** language, which provides a relatively higher level of abstraction with choice of gates and precise circuit design hidden, to languages such as IBM's **qiskit** and **Cirq** from Google which give the user more fine-grained control.

As with any programming challenge, there is a trade-off to be made between high-level abstraction and low-level control. High-level abstraction may lead to inefficient circuit design, a particularly important issue in these early days of QaaS where the number of qubits available is limited. In contrast, low-level control increases the cognitive overhead put on the user, and the skillset required. Certainly, some form of high-level abstraction will have to occur, and so we expect QC programming languages to be a fluid area over the next one to five years. Indeed, we are already beginning to see new entrants such as **Silq** into the high-level end of the field.

At the other end of the scale, we will also see advances in quantum firmware – the software layer that sits just above the actual physical hardware, and mediates between the high-level code of the programmer and the specific hardware being used. The firmware acts as another abstraction layer so that the qubits the high-level code interacts with have superior performance compared to the qubits in the bare hardware. Quantum firmware provides an alternative to Quantum Error Correction (QEC) as a means of dealing with the consequences of noisy qubits with fragile coherence. As **Ball et al** highlight, QEC can be resource costly, and so this acts as an incentive for the development of efficient quantum firmware.

What advances in QC algorithms will we see?

We have already emphasised that a large number of QC solutions make use of a small number of QC algorithms. These include the Quantum Fast Fourier Transform (QFFT) that underpins Shor's algorithm, Grover's algorithm that underpins several optimisation routines, and the adiabatic theorem that underpins the quantum annealing approaches to optimisation. These well-used algorithms form what one might think of as the current design patterns in QC.

Will we see a new fundamental algorithm or design pattern emerge over the next few years? Probably not. We say this based not on detailed evidence, but more because it is difficult to see from where such a new algorithm might emerge. The number of fundamentally distinct families of algorithms is small, despite it being nearly 40 years since the field of QC was first established. The rate at which fundamentally new QC algorithms emerge is modest and this forms our prior expectation. Recent investments at national level by countries such as the **UK**, **USA**, and **China** will undoubtedly increase this rate, but it is likely to remain low.

What we are more likely to see are significant improvements and refinements in the many variants of algorithms that have already been developed, as they are applied to more and more realistic and challenging problems. For example, we are already seeing the modification of Grover's Adaptive Search (GAS) algorithm from QUBO problems to **quadratic (and higher order) optimisation problems that include equality and inequality constraints**.

What advances in QC applications will we see?

It is inevitable that we will see an increase in the number of domains that QC is applied to. As QC moves through its hype cycle there will be an undoubted increase in the number of unachievable promises about the potential of QC made by some vendors. Towards the end of the next three to five years we will see a reduction in the froth as users figure out what works and what doesn't. In the short-term (one to three years) we will see the biggest advances in the areas where progress is easiest.

We anticipate these to be:

1. **Emulation of quantum-mechanical systems:** this is an application area we have not yet mentioned. It is the application of QC to the simulation of quantum systems themselves, such as in **quantum chemistry**. Here the quantum computer is being used as an accurate and faithful facsimile of the system we are trying to understand. Similar uses of QC have been made in the fields of **drug discovery** and **materials science**. The use of QC here is perhaps uncontroversial. Consequently, we would expect significant further advances in the use of QC in these fields, and any other fields where quantum-mechanical effects cannot be ignored.
2. **Commercial optimisation problems:** as highlighted previously, combinatorial optimisation problems are a natural application of QC and we expect to see significant growth in terms of the number of different applications. There will be advances in the application of both quantum annealers and gate-based quantum computers to commercial optimisation problems. This is because both technologies have their advantages and drawbacks – quantum annealers such as those from D-Wave offer significantly more qubits but are restricted to problems that can easily be mapped to finding the minimum of an energy landscape.
3. **Genuine quantum machine learning (QML) applications:** despite the elevated hype associated with QML, this area will still naturally attract a lot of attention, particularly from academic research groups. However, it is likely that any advances will remain within the academic sphere for the time being – it is unlikely to lead to a short-term revolution in the kind of tooling that a Data Scientist uses for their everyday tasks.

What advances in QaaS will we see?

We discuss the QaaS market last, primarily because its evolution and success will be dependent upon what happens in relation to the other issues we have raised. Firstly, we will certainly see an increase in the size of the hardware available via QaaS, and we will see a reduction in costs. Secondly, we will see an increase in the number of users in the short-term.

How sustainable the increase in the QaaS market is will depend primarily upon how quickly hardware capabilities can grow. If QaaS hardware improves, it will leave users free to focus on solving their problems.

A key turning point to look out for will be if and when users are able to integrate quantum solutions into a wider workflow. This will require and drive enhancements in the capacity, stability, and performance of QaaS.

The bottlenecks to QaaS sustainability will then be:

i. **whether users can achieve some form of genuine near-term quantum advantage, and;**

ii. **the ease of use of the various language tools.**

To some extent, the creation of QaaS by the likes of AWS and Microsoft Azure appears to be a 'chicken-and-egg' experiment. Certainly, the next half decade will prove to be critical – and extremely interesting – for that experiment.

Conclusion

It is an exciting time to embrace quantum computing. The next few years will likely see dramatic advances – some expected, some unforeseen – in hardware, approaches, and the ability to integrate the power of QC into commercial solutions. But the promise of a paradigm shift in computation tomorrow requires a shift in your mindset and capabilities today.

Thank you for taking the time to read this paper, and we hope that you have fun bounding up the quantum learning curve!

dunnhumby THE WORLD'S FIRST CUSTOMER DATA SCIENCE PLATFORM

dunnhumby is the global leader in Customer Data Science, empowering businesses everywhere to compete and thrive in the modern data-driven economy. We always put the Customer First. Our mission: to enable businesses to grow and reimagine themselves by becoming advocates and champions for their Customers.

With deep heritage and expertise in retail — one of the world's most competitive markets, with a deluge of multi-dimensional data — dunnhumby today enables businesses all over the world, across industries, to be Customer First.

The dunnhumby Customer Science Platform is our unique mix of technology, software and consulting enabling businesses to increase revenue and profits by delivering exceptional experiences for their Customers – in-store, offline and online. dunnhumby employs over 2,000 experts in offices throughout Europe, Asia, Africa, and the Americas working for transformative, iconic brands such as Tesco, Coca-Cola, Meijer, Procter & Gamble, Raley's, L'Oreal and Monoprix.

Connect with us to start the conversation

dunnhumby.com