

Distributed Quantum Computing: Applications and Challenges

Juan C. Boschero, Niels M. P. Neumann, Ward van der Schoot, Thom Sijpesteijn, and Robert Wezeman¹

The Netherlands Organisation for Applied scientific Research (TNO), The Hague,
The Netherlands

Abstract. Quantum computing is presently undergoing rapid development to achieve a significant speed-up promised in certain applications. Nonetheless, scaling quantum computers remains a formidable engineering challenge, prompting exploration of alternative methods to achieve the promised quantum advantage. An example is given by the concept of distributed quantum computing, which aims to scale quantum computers through the linking of different individual quantum computers. Additionally, distributed quantum computing opens the way to new applications on the longer term. This study seeks to give an overview of this technology on an application-level, considering both use cases and implementation considerations. In this way, this work aims to push forward the field of distributed quantum computing, aiming for real-world distributed quantum systems in the near future.

Keywords: Quantum computing, applications, distributed quantum computing, DQC, use-cases, challenges

1 Introduction

Computing power has grown exponentially since the early 1960s following the predictions of Moore’s law, which asserts that the number of transistors on a computer chip doubles roughly every two years [39]. Unfortunately, it is starting to become apparent that this exponential growth is reaching its limit. The already minuscule sizes of transistors limit the unbounded increase of the number of transistors on chips. Luckily, a new form of computing, called quantum computing, has arisen recently, which promises to continue this exponential growth.

Quantum computing is a form of computing based on quantum mechanics, a theory of physics relevant at the smallest scales. Quantum mechanics describes phenomena such as superposition and entanglement, which do not arise classically. Hence, quantum computers can perform fundamentally different operations than the ones performed by classical devices. With clever design of these operations, quantum computing makes it possible to reach up to exponential speed-ups in various fields, such as machine learning, chemistry and optimisation [23].

Quantum computing is currently under rapid development with an apparent exponential scaling in the number of qubits, the basic building blocks. This

scaling is similar to what Moore’s law predicted for classical computers [19], and what classical computers are currently following. Many applications require thousands of qubits for quantum computing to become of practical relevance, especially when accounting for decoherence effects. While modern quantum devices can achieve up to 1000 qubits [19], the quality of these qubits remains low. Numerous applications require qubits and operations with extremely low error rates. Currently available quantum devices, often referred to as noisy intermediate-scale quantum (NISQ) devices [50], are noisy and suffer from loss of information.

Because of this, practical large-scale algorithms are still years away, even with the current scaling of qubits and error rates. Instead of scaling the resources on a single device, multiple quantum devices can be combined to attain increased computational power with the given quantum devices. This idea is called *distributed quantum computing* (DQC), and works similarly to the classical distributed settings often used in high-performance computing (HPC). Depending on the noise rates of quantum communication links, DQC offers a possible faster path to scaling quantum hardware [26]. Additionally, DQC opens the path to new applications where different parties can collaborate and thereby solve more complex problems than when running algorithms themselves.

The first ideas of DQC are already applied to scale current quantum hardware [12]. An example are the Heron chips by IBM, which use the chiplet architecture [33], in which communication between different chips is used to scale the overall computational power [19, 26]. Furthermore, one upcoming technique is given by circuit cutting, in which larger circuits are cut up into smaller circuits to run on smaller quantum hardware [38, 37, 59]. While this is not the same as DQC, it is the same idea of using techniques to solve larger problems on a quantum computer than one would expect based on the resources alone [49, 52].

There are different ways of achieving DQC. In the broadest definition, DQC can be seen as solving a problem using multiple quantum chips or devices. The devices can be in close proximity, situated on separate chips within the same computer, or they can be located kilometres apart. This definition also includes settings where different quantum devices independently solve separate parts of a problem, and where the solutions to the individual parts are combined to give the answer to the full problem. If the outcome of the individual parts is classical, this distributed settings requires only classical communication between the devices. In this work, the focus lies on the case where the different quantum devices cooperate in a quantum way using quantum communication and where interaction between the devices is necessary before extracting the individual answers. Because of that, DQC is more specifically defined as the execution of a quantum circuit distributed over different parties, in which at least one gate is distributed over different devices, and applied through quantum communication.

Quantum communication entails the sharing of quantum information between different parties. While it is a key ingredient for DQC, it also plays a vital role in other applications, such as Quantum Key Distribution [3, 15] and the quantum internet [5, 29, 61]. These two examples specifically use quantum communication to transmit information from one party to another. In DQC,

however, quantum communication is a prerequisite for performing joint computations.

Generally speaking, there are two advantages of using DQC over quantum computing on a single device. The first advantage has already been mentioned above: by allowing cooperation between different quantum devices, the total amount of computing resources increases, resulting in a more powerful computational unit. One could enhance computational power in the near future by leveraging the diverse strengths of various quantum technologies through the combination of devices based on different technologies, thus advancing DQC capabilities. The second advantage is that it allows for collaborative computing. If the different quantum devices are controlled by different parties, each party could input their own data on their respective quantum device. DQC would then allow these parties to perform operations on their joint data, thus allowing collaborative insights among the parties. In the long term, DQC thus opens the way to completely new applications.

Until recently, most of the theoretical research in the field of DQC has been focused on distributing specific algorithms. In experimental studies, the emphasis has been on very fundamental aspects such as distributing gates across systems at distances ranging from a few meters [11] to several kilometers [35]. Recently, a couple of works have started to give an overview of the field of DQC, with works shown in [6] and [1], being most notable. In the first survey, the authors provide an overview of DQC by looking at four different perspectives: algorithms, networking, compiling and simulation. The work focuses mostly on the low-level challenges associated with these different aspects of DQC. The second work gives a full overview of the state of DQC from a rather technical perspective. It is a great overview for anyone trying to understand the workings of DQC, touching upon hardware, architecture, compilation and algorithm design.

In contrast, the present work aims to give an overview of the application-level aspects of DQC. Specifically, we focus on end-users and applications, detailing what DQC implies in practice and what the associated challenges and advantages are. Typically, a close interaction with classical high performance computers is required when running quantum algorithms. Emphasizing the forward-looking nature of this work is crucial, given that the existing DQC infrastructure is still in its early experimental stages. Furthermore, as this problem (and its solution) extends to distributed settings, we will not explicitly discuss this integration.

This work is structured as follows. In Section 2, we introduce two notions of DQC and explain their differences. We then discuss different use cases for both types of DQC in Section 3. In Section 4, we detail some (technical) considerations around the development and practice of DQC. In particular, we discuss how quantum algorithms, quantum communication and quantum hardware are relevant to a distributed framework. Lastly, we conclude our overview with a discussion and outlook on the future of DQC.

Table 1. Nomenclature for different types of distributed QC

Paradigm name	Resources	Data
Regular QC	local	local
Resource DQC	distributed	local
Data DQC	distributed	distributed

2 Types of DQC

As mentioned in the introduction, there are two main reasons to opt for distributed operations. These reasons extend beyond the field of DQC. The first reason is that your own computational resources do not suffice for solving the problem. The second reason is that you do not have all data available for solving the problem. Both reasons translate to different applications and infrastructures. Below we name and explain both types in the quantum setting, after which we discuss potential applications for both in the next chapter.

We call the first type *resource DQC*. Here, the available quantum resources are insufficient or unsuitable for running the algorithm and we turn to other parties or devices to collectively achieve the required resources. This type closely relates to a client-server model in classical distributed settings [2]. The used resources can even originate from different servers. From the perspective of the client, no difference should exist between one server and multiple linked servers. The insufficient resources can differ in type and extend beyond just an insufficient number of qubits. For certain algorithms, it could be beneficial to distribute (part of) the computation so that one can take advantage of hardware properties of the other device. For example, reducing the circuit depth by distributing the computation to a device with a larger gate set or a better qubit connectivity.

We call the second type *data DQC*. Here, the data is distributed over multiple parties and quantum computing and quantum communication can link the data to perform computations on all data collaboratively. This type of distributed computing closely relates to classical multi-party computation [7], with the added benefit that quantum computations can directly be performed on the data [41].

Encoding data onto quantum computers is not a straightforward process. It is an active area of study, typically involving the initialisation or preparation of quantum states at the start of a circuit to align with classical tensor data [67], or utilising quantum feature maps to encode classical inputs as quantum data [36]. This embedded data would subsequently be communicated through entangled qubits to other devices, as detailed in Section 4.2.

The two types of DQC may also appear combined and in a hierarchical setting. For instance, in a data DQC setting, one party may implement resource DQC to perform their local computations. Table 1 contains these two types of DQC, as well as regular quantum computing, and lists whether the resources and the data are local or distributed for each entry. Note that a setting with only local quantum resources and distributed data falls outside our scope for DQC (i.e., with at least one distributed quantum gate) and is thus omitted.

3 Different Use Cases

This section describes four potential applications of DQC. As practically any quantum algorithm can benefit from resource DQC, we stress that the list is incomplete. It should, however, be mentioned that certain applications are more naturally suitable for DQC than others. The next chapter gives some indication as to what makes an algorithm suitable for DQC. The four presented use cases are examples of suitable algorithms and provide insights in the potential of DQC.

Quantum Machine Learning: Machine learning (ML) is a leading contemporary research domain with applications in diverse fields, such as image and video recognition, natural language processing, recommendation systems, and fraud detection [54]. In its simplest form, machine learning focuses on training models with a certain amount of training data, such that the resulting model can be applied to unseen data to predict relevant properties related to this data. These ML tools have transcended academia and are now accessible to the general public, exemplified by instances like DALL-E2 [45], which generates images from text descriptions, and ChatGPT [44], a conversational artificial intelligence model capable of engaging in human-like interactions. The downside of these models is that their continual upkeep and refinement requires processing large data sets, resulting in the use of significant computational resources. The number of floating-point operations needed to train state-of-the-art models has been shown to be growing exponentially [40], which can quickly become a problem because of the unrealistic long training times.

Quantum Machine Learning (QML) is the field that tries to combine quantum computing with machine learning. A potential advantage of QML is speeding up computation, i.e., the quantum algorithm that solves the specific ML problem runs faster than a classical alternative algorithm. A possible speed-up is however not the only dimension that needs to be considered for assessing true meaningful advantages of QML. For an elaborate discussion we refer to chapter 9 of [56].

Both forms of DQC defined in Section 2 can be beneficial for QML. Resource DQC can facilitate the implementation of larger QML models that surpass the qubit capacity of an individual quantum device. Thus, this type of DQC can accommodate the execution of more complex models demanding increased qubit connectivity or adherence to a specialised gate-set. Data DQC, on the other hand, allows different parties to supply input to an QML algorithm collaboratively. This can be useful when different parties own different parts of the data that are needed to train a model. In the context of classification, this could for example be the case when one party possesses particular types of distorted images of an object, while another party knows the actual type of object due to having access to other resources such as sound. Combining their data, these two parties are able to train a ML model that is able to classify the distorted images.

In the NISQ era, so-called *variational quantum algorithms* (VQAs) are promising methods of quantum machine learning. VQAs are hybrid quantum-classical algorithms, where a parameterised circuit is run on a quantum computer and the parameter optimisation occurs in a classical outer loop. Two well-known VQAs are the highly structured QAOA [18] and the Variational Quantum Eigensolver

(VQE) [62]. VQAs are expected to be especially relevant in the NISQ era, as many believe the parameters can be optimised to accommodate for the noise present. In addition, because of their simple form, variational algorithms require only few qubits and small depth compared to more traditional fault-tolerant quantum algorithms [8]. Because of this, many if not most of the current quantum machine learning approaches are based on variational algorithms.

Recently, distributing quantum computations in the context of VQAs has seen its first research papers. Two examples are the resource DQC version of VQE by [13] and [28]. Many other research in distributed versions of VQAs tends to distribute the classical part of the computation (e.g., [16, 34, 42]), but this falls outside the scope of this present paper.

Secure computations: *Multi-party computation* refers to a field of research with the goal to allow two or more parties to perform analyses or computations on their joint data without revealing their own data. Security is a vital requirement for many applications in multi-party computation. This concerns the information exchanged between different parties and that the exchanged data cannot be exploited to disclose any unintended information about either party.

Quantum computers offer inherent security of quantum states, as quantum states are destroyed upon measurement. This naturally makes quantum technologies suitable for applications in security. Most quantum algorithms will be hybrid, i.e., combining classical and quantum routines, such as variational algorithms. The inherent security does not transfer to the classical routines, hence resulting in potential security issues. However, these security issues also arise the equivalent classical algorithms. The use of quantum only increases the security of the algorithm, specifically for the information stored in quantum states.

DQC adds extra security to the implementation of quantum computers. Firstly, in the context of resource DQC, it removes the need for sending input and algorithm data to the quantum computer provider. Currently, most quantum computers can only be accessed through cloud providers. To use cloud-based quantum computers, users have to send all instructions to the quantum computer provider and that way essentially share all information about their input and their algorithm. This removes all potential security offered by quantum devices. By using resource DQC, input and algorithm data only need to be inputted to a local quantum device, after which DQC allows the use of the cloud-based quantum computer and its required extra resources. No data is shared with the cloud provider because of the security obtained through quantum states [27].

Secondly, in the context of data DQC, multi-party computation are possible without explicit data sharing between the different parties. In such settings, each party inputs their data into their local quantum device, after which a distributed computation is performed. Then, no single party directly learns data that is inputted by other parties. In addition, the algorithm can be modified so that no party learns what algorithm was actually run and which problem was actually solved, both while executing the algorithm, as well as when measuring the qubits and processing the measurement outcomes. Note however that each party does learn some part of the algorithm and quantum state, so no hard guarantees

can be given on the security resulting from this way of distributing quantum computations. The classical communication required between quantum devices, however, does not leak the quantum state information of the remote qubits.

Breaking cryptography: Cryptography is a field of research that enables many processes in ICT, such as secure communication, data protection and authenticated messaging. The algorithms to protect these processes depend on certain mathematical problems which are believed to be hard to solve using current classical computers. Because of this, cryptography has managed to protect vital ICT infrastructures for over forty years.

Quantum computers will have a significant impact, as they prove valuable in breaking cryptographic protocols. Grover's algorithm [20] weakens symmetric cryptography using a quadratically faster unstructured search algorithm, whereas Shor's algorithm [57] breaks most asymmetric cryptography by exploiting the structure used in the protocol to achieve an exponential speed-up.

Both algorithms could benefit from a resource DQC implementation. Grover's algorithm is not naturally suited for DQC implementations, as its traditional implementation requires many qubits communicating with one another. However, an exact version of the algorithm exists where many function evaluations are performed in parallel [69]. This version requires fewer qubits communicating with one another, making the algorithm more suitable for distributed settings.

The situation is quite different for Shor's algorithm. The phase estimation part of the algorithm consists of multiple controlled phase gates that all commute. In addition, only certain qubits need to communicate with one another in the algorithm. These two characteristics make it that Shor's algorithm is naturally suited for distributed settings. The work of [64] shows a possible implementation of a distributed Shor's algorithm and discusses the potential speed-up.

Data DQC seems to be of no relevance for both algorithms, as neither of them requires input data distributed over or owned by different parties.

Quantum interferometry: Interferometers are instruments that utilise the interference of superimposed waves for extracting information. These instruments find applications in various fields [22].

Current interferometers convert the information from the incoming wave packets in classical data, which proved optimal for applications requiring small amounts of interferometers measuring a source with a high signal to noise ratio [22]. However, classical computations with interferometric data tend to encounter challenges, particularly in fields where the signal-to-noise ratio is low and multiple interferometers are required, like astronomy [4]. Quantum computers can solve issues in such scenarios as the quantum information of a wave package, or photon, can be directly transcribed onto a qubit resulting in less resolution noise. This gives rise to new quantum interferometers.

If one were to have access to different interferometers, the same signal could be measured from different locations and in more detail. A central interferometer could then combine the light incoming from the separate interferometers, after which the central interferometer extracts the required information. By having access to data from multiple interferometers, this extracted information would

be more accurate. However, classical methods require the incoming light to be brought together physically, which is susceptible to noise.

Quantumly, the case is different, leading to a data DQC use case. Given multiple quantum interferometers, the quantum information measured by a single quantum interferometer could directly be translated onto its local quantum device. Different quantum computers located at the different interferometers can then run a DQC algorithm. The result of this algorithm yields the required information, without having to physically bring the different signals together. This shows that data DQC can improve the performance of quantum interferometers.

It should be noted that different to the aforementioned use cases, the advantage of DQC mainly comes from the quantum conversion of information rather than computation. Still, computation is required to process the signals, for which a quantum computer is naturally suitable. Because of this, this use case is still considered as a DQC use case, be it with a different focus.

4 Technical considerations

DQC promises to bring many advantages, yet it is far from an out-of-the-box solution. Quantum computers themselves already have their limitations and will likely only bring (great) benefits to specific fields. Even within these fields, DQC is unlikely to speed up all quantum algorithms. When applying DQC, there are different technical considerations to take into account that influence whether the use case benefits from DQC, and if so, how it should be used specifically. This section discusses the technical considerations that currently seem most important. The considerations are grouped top-down in three categories: algorithms, quantum communication architecture and quantum computing hardware.

4.1 Algorithms

Many practical aspects come into play when developing (distributed) quantum algorithms. This imposes a limit to the sort of quantum algorithms that benefit from running on distributed systems.

First, algorithms that contain many gates between different pairs of qubits are inherently unsuitable for DQC implementations. Noise in the communication links dominate imperfections in quantum devices and hence form the limiting factor in implementations of DQC. An efficient distributed quantum algorithm should therefore minimize the amount of communication between different devices. Effectively, most operations should be performed on the same device, which is possible if the qubits can be split up into multiple groups, so that most of the operations only happen within each of the separate qubit groups. Algorithms that require many different pairs of qubits to interact, such as in quantum volume circuits [10], therefore only marginally benefit from distributing quantum computations. Specifically for data DQC, this means that algorithms in which the data stored at different quantum computers needs to be communicated between different parties a significant amount of times, are unsuitable for DQC.

On the other hand, algorithms suitable for standard circuit cutting techniques might also be suitable for distributed settings, as the overhead due to the circuit cutting translates roughly to the communication complexity in distributed algorithms. One must note that some protocols retain a high success probability in a distributed setting, even with noisy communication links [47].

Second, algorithms that contain many blocks of commuting operations are naturally suitable for DQC implementations. This is due to the fact that commuting operations can be applied in parallel. So in DQC settings, every individual quantum computer can run one block of commuting operations in parallel. To apply these commuting operations in parallel, there needs to be a way to link the different operations. This is achieved in [25] using a non-local quantum fan-out gate, a gate that naturally extends the non-local CNOT gate, introduced in [17], to multiple targets [65]. As such, this reduces the depth of some algorithms significantly, as subsequent operations are replaced by parallel ones. A prime example is the phase estimation algorithm, which allows us to parallelize Shor’s factoring algorithm [57] to constant depth [25].

4.2 Quantum Communication Architecture

To turn single-device quantum computing into DQC, connections between different quantum devices are required. Different considerations have to be considered when setting up such connections. These considerations concern either the quantum network topology as a whole, or the coupling of specific quantum devices. The following two subsections will detail the considerations concerning both.

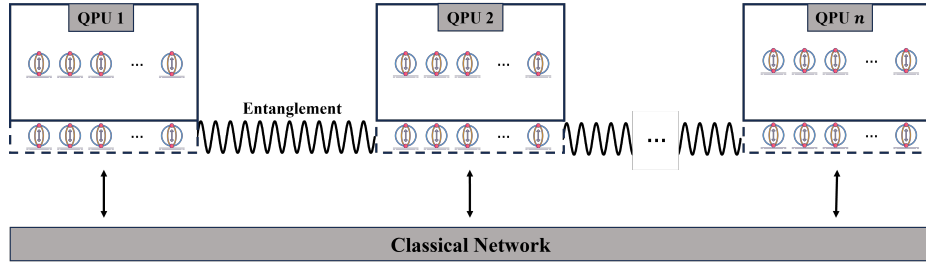


Fig. 1. Illustration of a fundamental quantum computing network topology featuring n quantum computers. Each square symbolizes an abstracted quantum computing processing unit (QPU), equipped with an unspecified number of computational qubits within the solid square, and an additional qubit within each dashed box, denoting the communication qubit. These communication qubits facilitate the establishment of entanglement between individual QPUs. Additionally, the QPUs are interconnected via a classical network, essential for executing distributed algorithms.

Quantum Network Topology In DQC, each quantum device uses two types of qubits: *Computation qubits* used to perform the quantum algorithm; and,

communication qubits used to communicate with other quantum devices. With the communication qubits, we can set up entanglement between different devices. Therefore, a quantum computer must forfeit at least one qubit to establish communication with another quantum computer, rendering it unavailable for directly computing the algorithm. Moreover, the quantum devices necessitate a pre-established classical network among them to execute distributed algorithms. This is illustrated in Figure 1, which depicts a fundamental quantum network topology between quantum computers.

Which qubit is used for which task is an important decision, as the total number of qubits is limited per device and the coherence times of the qubits is still limited.

For each pair of entangled communication qubits between two devices, the two devices can perform a joint non-local 2-qubit gate. After the 2-qubit gate has been performed, the entanglement must be set up again before it can be used for a new non-local operation. With more communication qubits, users can perform more non-local operations before having to regenerate entanglement between the communication qubits. As a down side, the computational power of a device decreases, as less computation qubits are available.

For each quantum device, some qubits are more suited for communication, whereas others are better for computation. Which qubit is best for which task depends on for instance the number of local (SWAP) operations required and the fidelity of the qubits when connecting them to a remote quantum network. The optimal choice can even depend on time [60]. In addition, during a quantum algorithm, a qubit can change between being a communication or computation qubit, to accommodate which role fits best for which qubit at any moment.

A closely related question is which quantum devices are directly coupled in the setting with more than two quantum devices,. If two quantum devices have a shared entangled pair of qubits, this pair can be used to perform non-local quantum operations between them directly. If two devices do not have a shared entangled pair of qubits, non-local quantum operations can only be performed through a chain of entangled qubits on other quantum devices or via entanglement swapping operations. The non-local operation will then destroy all pairs of entangled qubits. A direct quantum link between the device thus saves entanglement generation efforts. On the other side, every quantum connection between two different quantum devices requires a fresh communication qubit. The choice for which quantum links exist between which devices is also something that can be optimised during the execution of an algorithm.

Both considerations need to be taken into account when setting up a distributed quantum system. For example, a designer can reduce the total number of communication qubits by not directly connecting two quantum devices when only a few non-local operations need to be implemented between them. On the other hand, a designer can choose to set up many quantum connections between quantum devices that have to perform many non-local operations between them. Again, the moment in time, the quantum algorithm, and the individual quantum

computer characteristics influence which devices are directly connected and how many qubits will be used for each connection.

Coupling of quantum devices Shared entangled pairs of qubits help couple different quantum devices. As different physical phenomena (see next section) can help produce qubits, different physical connections between the different types of qubits are necessary.

In general, we distinguish between direct and indirect coupling. Direct coupling happens when the two quantum devices directly interact with each other, resulting in the required entanglement. Direct coupling mainly works for resource DQC when the devices are physically close. In data DQC, the different quantum systems most likely are physically separated. In this case, indirect coupling can be used. Indirect coupling uses an intermediate system to set up the entanglement. First, local entanglement between the intermediate system and one of the two devices is set up. The qubit of the intermediate system is then transported to the other quantum device, and entangled with a qubit of the other device. This protocol establishes entanglement between the two quantum devices.

Despite the lower dissipation and decoherence demonstrated by direct coupling [66], indirect coupling currently serves as the primary approach for quantum communication [63]. For the purpose of quantum information transfer, current attention is directed towards optomechanical resonators because they can link the vibrations of a system (phonons) to particles of light (photons). These photons can subsequently be transmitted through optical fibres [21] or free space. Quantum information can accurately be transported across 143km in experimental settings [24]. In addition, the teleportation of a CNOT gate between logical qubits has been realised [9]. Superconducting cables can transport qubits by indirectly coupling onto co-planar wave guides connected to the superconducting quantum computers as demonstrated in [68], where a GHZ was correctly transported between state between two superconducting quantum computers.

4.3 Quantum Computing Hardware

The quantum hardware naturally significantly impacts the implementation of DQC, and different aspects have to be taken into account. Currently, a wide variety of quantum devices are in development. These devices differ significantly in for instance used qubit technology, qubit topology, number of qubits and many more. In this section, we discuss which of these aspects are relevant for DQC.

With the rise in quantum information science, a wide variety of quantum architectures emerged. As of today, the systems that have undergone experimental controllable and coherent quantum manipulation include trapped ions [43], atoms [53], spin systems [58] and superconducting circuits [30, 46]. Moreover, among the various quantum architectures currently under development are postulated options such as magnonic qubits [31] and skyrmion qubits [51]. These quantum systems are built with the specifications set by [14] in mind, which states that a competitive quantum computer must strive to achieve:

1. A scalable physical system with well characterised qubits;
2. The ability to initialise the qubit states to a simple state, i.e., $|000\dots\rangle$;
3. Decoherence times (much) longer than gate operation times;
4. A “universal” set of quantum gates;
5. A qubit-specific measurement capability;
6. The ability to interconvert stationary and flying qubits;
7. The ability to properly transmit flying qubits from one location to another.

The first five specifications form the basic requirements of a local quantum computer, while specification 6 and 7 detail the needs for proper communication between quantum systems. Different quantum systems excel at different requirements. As an example, superconducting qubits, and especially transmon qubits, show a considerable advantage for requirements 1 and 2. Ion trap qubits show considerable advantages for requirement 3. Currently, no single standardised system exists that consistently outperforms others in *all* aspects.

Note that specifications 6 and 7 are of special relevance to DQC compared to local quantum computations. Although specifications 6 and 7 are heavily quantum communication related, it is important to note that trapped ion systems and photons excel in this field. Transmitting photonic [32] and trapped ion [48] qubits has been achieved at long distances using free space or fibre optic cables.

Although the specifications in [14] are important for basic functionalities, they do not necessarily translate to practical applicability. To measure how well a quantum device can solve partial problems, quantum metrics are required. Various quantum metrics exist, each with their own focus and use case [55].

Both the above specifications and quantum metrics are relevant for the design of DQC algorithms. In classical computing, deploying an algorithm across multiple nodes is often unfavourable due to startup costs, security concerns, and the added complexities in overheads. However, due to the diverse architectures offered in quantum computing, distributed quantum computing often possesses a larger pay off when compared to its classical counterpart as the advantages of the different architectures can be leveraged. Considering the strengths and weaknesses of the different quantum devices, the quantum algorithm can be instantiated such that the strength of each of the devices is utilised, while avoiding each of the weaknesses. For example, pairing a computer architecture possessing long coherence times with an architecture with fast operation times allows one system to store information while the other executes the operations of an algorithm. The mapping of algorithmic qubits to physical qubits should thus take these aspects into account. It should also be noted that some quantum metrics allow the benchmarking of distributed quantum systems, which could show useful insights when deciding which system to use for which part.

5 Discussion and outlook

In this work, we presented an overview of distributed quantum computing (DQC) and its use cases. We first identified two types of DQC: *Resource* DQC, where the local quantum computation resources are insufficient; and, *Data* DQC, where the

relevant information is distributed over multiple parties and quantum computers are used to allow computations on the shared data. Combinations between the two also exist, as well as hierarchical settings.

In Section 3, we discussed the benefits that DQC may possess in the areas of quantum machine learning, secure computing, breaking encryption, and quantum interferometry. A clear view of the precise value that DQC will bring is still unclear, as it depends on the nature of the hardware used in the future.

Section 4 discussed different technical considerations around quantum computing in a distributed setting. We discussed the effects of DQC on quantum algorithms, noting that communication will be a bottleneck when distributing algorithms. The number of devices over which the algorithm is distributed will affect the way performance and the losses due to the communication overhead between the devices, furthermore, more devices give more degrees of freedom to optimally implement the algorithm. In the context of quantum communication architecture, we discuss the difference between computation and communication qubits, as well as the consequences of directly and indirectly coupling quantum devices. With direct coupling, a quantum link between two devices exists, whereas with indirect coupling, entanglement between two devices has to be set up via one or more other (intermediate) devices. We ended the section by considering quantum hardware, noting that DQC is currently only achievable using photon or trapped ion systems as these systems easily allow the transition of in-transit quantum states to quantum states useful for computation.

In the near future, resource DQC offers a solution to scaling up quantum computational power, by linking together multiple smaller quantum devices. Furthermore, data DQC opens the path to new applications, where multiple parties can collaboratively solve a problem with their combined data. Especially data DQC requires quantum devices of higher quality than currently available. Currently, the main challenges lie in improving the quantum hardware and improving (the capabilities of) the quantum network. The first implementations on actual distributed hardware will however require dedicated implementations and low-level optimizations.

Distributed quantum computing does introduce an additional overhead, both in the number of required qubits, as well as in the communication between different devices. Yet, DQC also offers a new path towards scaling up quantum hardware capabilities and it opens the way to novel applications of quantum computers. With this work, we presented a first overview of DQC, incorporating technical considerations as well as algorithmic considerations and use-cases. In this way, we hope that this overview forms a unified language for research on DQC, and gives first glimpses into its potential impact and aspects to take into account. This will hopefully accelerate the field of DQC, paving the way towards proper implementation of DQC in the short and long term.

References

1. Barral, D., Cardama, F.J., Díaz, G., Faílde, D., Llovo, I.F., Juane, M.M., Vázquez-Pérez, J., Villasuso, J., Piñeiro, C., Costas, N., Pichel, J.C., Pena, T.F., Gómez, A.:

- Review of distributed quantum computing. from single qpu to high performance quantum computing (2024), <https://arxiv.org/abs/2404.01265>
2. Benatallah, B., Casati, F., Toumani, F.: Web service conversation modeling: a cornerstone for e-business automation. *IEEE Internet Computing* 8(1), 46–54 (2004)
 3. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. *Theoretical Computer Science* 560, 7–11 (dec 2014)
 4. Boffin, H.M.J., Hussain, G., Berger, J.P., Schmidtbreick, L. (eds.): *Astronomy at High Angular Resolution, Astrophysics and Space Science Library*, vol. 439 (2016)
 5. Cacciapuoti, A.S., Caleffi, M., Tafuri, F., Cataliotti, F.S., Gherardini, S., Bianchi, G.: Quantum internet: Networking challenges in distributed quantum computing. *IEEE Network* 34(1), 137–143 (2020)
 6. Caleffi, M., Amoretti, M., Ferrari, D., Cuomo, D., Illiano, J., Manzalini, A., Cacciapuoti, A.S.: *Distributed Quantum Computing: a Survey* (2022)
 7. Canetti, R., Feige, U., Goldreich, O., Naor, M.: Adaptively secure multi-party computation. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. p. 639–648. STOC '96, Association for Computing Machinery, New York, NY, USA (1996)
 8. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., Coles, P.J.: Variational quantum algorithms. *Nature Reviews Physics* 3(9), 625–644 (aug 2021)
 9. Chou, K.S., Blumoff, J.Z., Wang, C.S., Reinhold, P.C., Axline, C.J., Gao, Y.Y., Frunzio, L., Devoret, M.H., Jiang, L., Schoelkopf, R.J.: Deterministic teleportation of a quantum gate between two logical qubits. *Nature* 561(7723), 368–373 (Sep 2018), <http://dx.doi.org/10.1038/s41586-018-0470-y>
 10. Cross, A.W., Bishop, L.S., Sheldon, S., Nation, P.D., Gambetta, J.M.: Validating quantum computers using randomized model circuits. *Phys. Rev. A* 100, 032328 (Sep 2019)
 11. Daiss, S., Langenfeld, S., Welte, S., Distant, E., Thomas, P., Hartung, L., Morin, O., Rempe, G.: A quantum-logic gate between distant quantum-network modules. *Science* 371(6529), 614–617 (2021), <https://www.science.org/doi/abs/10.1126/science.abe3150>
 12. Davarzani, Z., Zomorodi, M., Houshmand, M.: A hierarchical approach for building distributed quantum systems. *Scientific Reports* 12(1) (9 2022), <http://dx.doi.org/10.1038/s41598-022-18989-w>
 13. DiAdamo, S., Ghibaudi, M., Cruise, J.: Distributed quantum computing and network control for accelerated VQE. *IEEE Transactions on Quantum Engineering* 2, 1–21 (2021)
 14. DiVincenzo, D.P.: The Physical Implementation of Quantum Computation. *Fortschritte der Physik* 48(9-11), 771–783 (sep 2000)
 15. Djordjevic, I.B.: Chapter 15 - quantum key distribution. In: Djordjevic, I.B. (ed.) *Quantum Information Processing, Quantum Computing, and Quantum Error Correction*, pp. 703–784. Academic Press, second edn. (2021)
 16. Du, Y., Qian, Y., Wu, X., Tao, D.: s. *IEEE Transactions on Quantum Engineering* 3, 1–16 (2022)
 17. Eisert, J., Jacobs, K., Papadopoulos, P., Plenio, M.B.: Optimal local implementation of nonlocal quantum gates. *Phys. Rev. A* 62, 052317 (Oct 2000)
 18. Farhi, E., Goldstone, J., Gutmann, S.: *A Quantum Approximate Optimization Algorithm* (2014)
 19. Gambetta, J.: *IBM Quantum Computing: Roadmap* (12 2023)

20. Grover, L.K.: A Fast Quantum Mechanical Algorithm for Database Search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. p. 212–219. STOC '96, Association for Computing Machinery, New York, NY, USA (1996)
21. Hafezi, M., Chang, D., Gritsev, V., Demler, E., Lukin, M.: Photonic quantum transport in a nonlinear optical fiber. *Europhysics Letters* (epl) 94 (07 2009)
22. Hariharan, P.: Basics of interferometry. Elsevier Academic Press, Boston, 2nd ed. edn. (2006)
23. Hassija, V., Chamola, V., Goyal, A., Kanhere, S.S., Guizani, N.: Forthcoming applications of quantum computing: peeking into the future. *IET Quantum Communication* 1(2), 35–41 (2020)
24. Herbst, T., Scheidl, T., Fink, M., Handsteiner, J., Wittmann, B., Ursin, R., Zeilinger, A.: Teleportation of entanglement over 143 km. *Proceedings of the National Academy of Sciences* 112(46), 14202–14205 (nov 2015)
25. Høyer, P., Špalek, R.: Quantum fan-out is powerful. *Theory of Computing* 1(5), 81–103 (2005)
26. IBM: Expanding the IBM Quantum roadmap to anticipate the future of quantum-centric supercomputing (may 2022)
27. Jozsa, R.: An introduction to measurement based quantum computation (2005)
28. Khait, I., Tham, E., Segal, D., Brodutch, A.: Variational Quantum Eigensolvers in the Era of Distributed Quantum Computers (2023)
29. Kimble, H.J.: The quantum internet. *Nature* 453(7198), 1023–1030 (Jun 2008), <http://dx.doi.org/10.1038/nature07127>
30. Kjaergaard, M., Schwartz, M.E., Braumüller, J., Krantz, P., Wang, J.I.J., Gustavsson, S., Oliver, W.D.: Superconducting Qubits: Current State of Play. *Annual Review of Condensed Matter Physics* 11(1), 369–395 (2020)
31. Lachance-Quirion, D., Tabuchi, Y., Gloppe, A., Usami, K., Nakamura, Y.: Hybrid quantum systems based on magnonics. *Applied Physics Express* 12(7), 070101 (jun 2019)
32. Lago-Rivera, D., Rakonjac, J.V., Grandi, S., de Riedmatten, H.: Long distance multiplexed quantum teleportation from a telecom photon to a solid-state qubit. *Nature Communications* 14(1) (apr 2023)
33. LaRacuenta, N., Smith, K.N., Imany, P., Silverman, K.L., Chong, F.T.: Modeling short-range microwave networks to scale superconducting quantum computation (2023), <https://arxiv.org/abs/2201.08825>
34. Li, J., Alam, M., Ghosh, S.: Large-Scale Quantum Approximate Optimization via Divide-and-Conquer. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 42(6), 1852–1860 (2023)
35. Liu, X., Hu, X.M., Zhu, T.X., Zhang, C., Xiao, Y.X., Miao, J.L., Ou, Z.W., Liu, B.H., Zhou, Z.Q., Li, C.F., Guo, G.C.: Distributed quantum computing over 7.0 km (2023)
36. Lloyd, S., Schuld, M., Ijaz, A., Izaac, J., Killoran, N.: Quantum embeddings for machine learning (2020)
37. Mitarai, K., Fujii, K.: Constructing a virtual two-qubit gate by sampling single-qubit operations. *New Journal of Physics* 23(2), 023021 (feb 2021), <https://dx.doi.org/10.1088/1367-2630/abd7bc>
38. Mitarai, K., Fujii, K.: Overhead for simulating a non-local channel with local channels by quasiprobability sampling. *Quantum* 5, 388 (Jan 2021), <http://dx.doi.org/10.22331/q-2021-01-28-388>
39. Moore, G.E.: Cramming more components onto integrated circuits. *Electronics* 38(8), 114–117 (4 1965)

40. Narayanan, D., Shoeybi, M., Casper, J., LeGresley, P., Patwary, M., Korthikanti, V., Vainbrand, D., Kashinkunti, P., Bernauer, J., Catanzaro, B., Phanishayee, A., Zaharia, M.: Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC '21, Association for Computing Machinery, New York, NY, USA (2021)
41. Neumann, N.M.P., Wezeman, R.S.: Distributed quantum machine learning. In: Innovations for Community Services, pp. 281–293. Springer International Publishing (2022)
42. Niu, Y.F., Zhang, S., Ding, C., Bao, W.S., Huang, H.L.: Parameter-parallel distributed variational quantum algorithm. *SciPost Physics* 14(5) (May 2023)
43. Noel, C., Niroula, P., Zhu, D., Risinger, A., Egan, L., Biswas, D., Cetina, M., Gorshkov, A.V., Gullans, M.J., Huse, D.A., Monroe, C.: Measurement-induced quantum phases realized in a trapped-ion quantum computer. *Nature Physics* 18(7), 760–764 (jun 2022)
44. OpenAI: ChatGPT. URL: <https://openai.com/chatgpt> (2022)
45. OpenAI: DALL-E2. URL: <https://openai.com/dall-e-2> (2022)
46. Orlando, T.P., Mooij, J.E., Tian, L., van der Wal, C.H., Levitov, L.S., Lloyd, S., Mazo, J.J.: Superconducting persistent-current qubit. *Phys. Rev. B* 60, 15398–15413 (Dec 1999)
47. Otero, A.R., Neumann, N.M.P., van der Schoot, W., Wezeman, R.: Noise robustness of a multiparty quantum summation protocol (2023), accepted at ICCS 2024
48. Palani, D., Hasse, F., Kiefer, P., Boeckling, F., Schroeder, J.P., Warring, U., Schaetz, T.: High-fidelity transport of trapped-ion qubits in a multilayer array. *Physical Review A* 107(5) (may 2023)
49. Piveteau, C., Sutter, D.: Circuit knitting with classical communication. *IEEE Transactions on Information Theory* pp. 1–1 (2023)
50. Preskill, J.: Quantum Computing in the NISQ era and beyond. *Quantum* 2, 79 (Aug 2018)
51. Psaroudaki, C., Panagopoulos, C.: Skyrmion Qubits: A New Class of Quantum Logic Elements Based on Nanoscale Magnetization. *Physical Review Letters* 127(6) (aug 2021)
52. Rennela, M., Brand, S., Laarman, A., Dunjko, V.: Hybrid divide-and-conquer approach for tree search algorithms. *Quantum* 7, 959 (Mar 2023), <https://doi.org/10.22331/q-2023-03-23-959>
53. Saffman, M.: Quantum computing with atomic qubits and Rydberg interactions: progress and challenges. *Journal of Physics B: Atomic, Molecular and Optical Physics* 49(20), 202001 (oct 2016)
54. Sarker, I.H.: Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science* 2(3), 160 (2021)
55. van der Schoot, W., Wezeman, R., Eendebak, P.T., Neumann, N.M.P., Phillipson, F.: Evaluating three levels of quantum metrics on quantum-inspire hardware. *Quantum Information Processing* 22(12) (Dec 2023), <http://dx.doi.org/10.1007/s11128-023-04184-x>
56. Schuld, M., Petruccione, F.: Machine Learning with Quantum Computers. *Quantum Science and Technology*, Springer Cham (2021)
57. Shor, P.W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* 26(5), 1484–1509 (1997)
58. Struck, P.R., Burkard, G.: Spin Quantum Computing, pp. 71–103. Springer Netherlands, Dordrecht (2016)

59. Tang, W., Tomesh, T., Suchara, M., Larson, J., Martonosi, M.: Cutqc: using small quantum computers for large quantum circuit evaluations. In: Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. ASPLOS '21, ACM (Apr 2021), <http://dx.doi.org/10.1145/3445814.3446758>
60. Wang, Z., Li, J., Xue, K., Cheng, S., Yu, N., Sun, Q., Lu, J.: An Asynchronous Entanglement Distribution Protocol for Quantum Networks. *IEEE Network* 36(5), 40–47 (2022)
61. Wehner, S., Elkouss, D., Hanson, R.: Quantum internet: A vision for the road ahead. *Science* 362(6412), eaam9288 (2018), <https://www.science.org/doi/abs/10.1126/science.aam9288>
62. de Wolf, R.: Quantum computing: Lecture notes (2023)
63. Xiang, Z.L., Ashhab, S., You, J.Q., Nori, F.: Hybrid quantum circuits: Superconducting circuits interacting with other quantum systems. *Reviews of Modern Physics* 85(2), 623–653 (apr 2013)
64. Yimsiriwattana, A., Jr., S.J.L.: Distributed quantum computing: a distributed Shor algorithm. In: Donkor, E., Pirich, A.R., Brandt, H.E. (eds.) *Quantum Information and Computation II*. vol. 5436, pp. 360 – 372. International Society for Optics and Photonics, SPIE (2004)
65. Yimsiriwattana, A., Lomonaco, S.J.: Generalized GHZ States and Distributed Quantum Computing (2004)
66. Zhang, G., James, M.R.: Direct and Indirect Couplings in Coherent Feedback Control of Linear Quantum Systems. *IEEE Transactions on Automatic Control* 56(7), 1535–1550 (jul 2011)
67. Zhang, X.M., Li, T., Yuan, X.: Quantum state preparation with optimal circuit depth: Implementations and applications. *Phys. Rev. Lett.* 129, 230504 (Nov 2022), <https://link.aps.org/doi/10.1103/PhysRevLett.129.230504>
68. Zhong, Y., Chang, H.S., Bienfait, A., Dumur, É., Chou, M.H., Conner, C.R., Grebel, J., Povey, R.G., Yan, H., Schuster, D.I., Cleland, A.N.: Deterministic multi-qubit entanglement in a quantum network. *Nature* 590(7847), 571–575 (feb 2021)
69. Zhou, X., Qiu, D., Luo, L.: Distributed exact Grover’s algorithm. *Frontiers of Physics* 18(5) (Aug 2023)