

RESEARCH ARTICLE | MARCH 20 2025

S-QGPU: Shared quantum gate processing unit for distributed quantum computing

Shengwang Du  ; Yufei Ding  ; Chunming Qiao 



AVS Quantum Sci. 7, 013803 (2025)

<https://doi.org/10.1116/5.0241972>

 CHORUS



Articles You May Be Interested In

Shopping for a time-sharing service

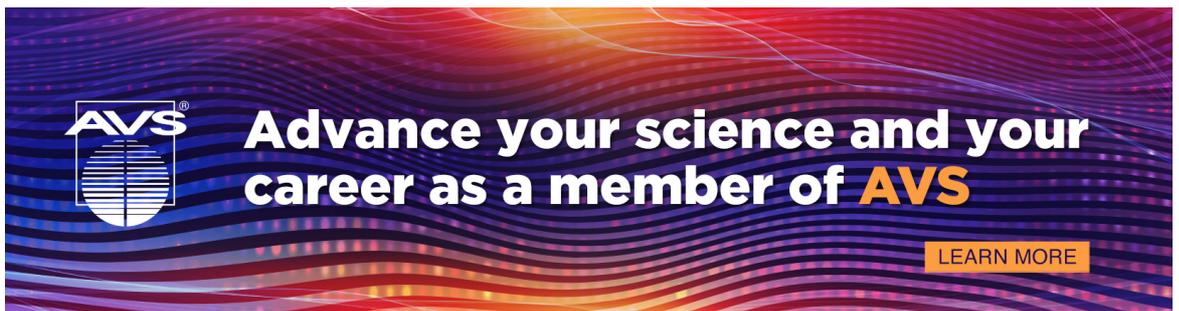
Physics Today (July 1970)

1956 Nobel Prize in Physics

Physics Today (January 1957)

Public service and human contributions

Physics Today (October 1967)

Advance your science and your career as a member of AVS

[LEARN MORE](#)

S-QGPU: Shared quantum gate processing unit for distributed quantum computing

Cite as: AVS Quantum Sci. **7**, 013803 (2025); doi: [10.1116/5.0241972](https://doi.org/10.1116/5.0241972)

Submitted: 1 October 2024 · Accepted: 5 March 2025 ·

Published Online: 20 March 2025



View Online



Export Citation



CrossMark

Shengwang Du,^{1,2,3,4,a)}  Yufei Ding,^{5,b)}  and Chunming Qiao^{6,c)} 

AFFILIATIONS

¹Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana 47907, USA

²Department of Physics and Astronomy, Purdue University, West Lafayette, Indiana 47907, USA

³Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, Indiana 47907, USA

⁴Department of Physics, The University of Texas at Dallas, Richardson, Texas 75080, USA

⁵Department of Computer Science and Engineering, University of California San Diego, La Jolla, California 92093, USA

⁶Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, New York 14260, USA

^{a)}Electronic mail: dusw@purdue.edu

^{b)}Electronic mail: yufeiding@ucsd.edu

^{c)}Electronic mail: qiao@buffalo.edu

ABSTRACT

We propose a distributed quantum computing (DQC) architecture in which individual small-sized quantum computers are connected to a shared quantum gate processing unit (S-QGPU). The S-QGPU comprises a collection of hybrid two-qubit gate modules for remote gate operations. In contrast to conventional DQC systems, where each quantum computer is equipped with dedicated communication qubits, S-QGPU effectively pools the resources (e.g., the communication qubits) together for remote gate operations, and, thus, significantly reduces the cost of not only the local quantum computers but also the overall distributed system. Our preliminary analysis and simulation show that S-QGPU's shared resources for remote gate operations enable efficient resource utilization. When not all computing qubits (also called data qubits) in the system require simultaneous remote gate operations, S-QGPU-based DQC architecture demands fewer communication qubits, further decreasing the overall cost. Alternatively, with the same number of communication qubits, it can support a larger number of simultaneous remote gate operations more efficiently, especially when these operations occur in a burst mode.

© 2025 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1116/5.0241972>

I. INTRODUCTION

Quantum computing represents a paradigm shift in computational power, promising to solve problems that are currently intractable for classical computers. Theoretically, an ideal quantum computer with 300 qubits possesses a staggering 2^{300} -dimensional Hilbert space, rendering classical computing techniques inaccessible, even if every atom in the observable universe were to be converted into classical bits. Such a fully connected N -qubit gate-based universal quantum computer requires that a unitary controlled gate operation can be directly performed between any two arbitrary qubits. However, in the real world, the connectivity of a local quantum computer is always limited by physical constraints. Therefore, the number of qubits in a quantum computer is not a “precise” measure of its overall computing

power.¹ For example, in a superconductor quantum computer, its real computing power is always much lower than what appears to be based on the number of qubits, because a superconductor qubit can only interact with its nearby qubits (nearest-neighbor coupling), and hence, completion of a gate operation between two distant qubits must go through many sequential intermediate gates.² For this reason, although IBM has released a 1121-qubit processor called Condor in 2023,³ they are all still far away from being sufficient to support practically useful applications beyond scientific interests. Trapped-ion⁴ and single neutral-atom-based^{5,6} quantum computers have similar challenges.

Due to many physical constraints and technological limitations, it is difficult, if not impossible, to build a monolithic fully connected quantum computer with a very large number (N) of qubits, and let

alone to guarantee a direct controlled gate operation can be performed between any two arbitrary qubits. Extending from N to $N + 1$ in such a quantum computer is more than just physically adding one more qubit. Therefore, a conventional wisdom is that the cost of such a fully connected local quantum computer increases *exponentially* as the number of qubits increases. Consequently, there is a growing interest in exploring distributed quantum computing (DQC) systems that can interconnect many small-size, cost-effective local quantum computers, or processors.^{7–14}

Nevertheless, the central challenge in constructing an effective DQC system lies in facilitating remote gate operations that involve computing qubits from different nodes. To accomplish this, several components are typically required in addition to a quantum network comprising quantum links, switches, and repeaters to support inter-node quantum communication.¹⁵ These components include (1) transducers for efficient conversion between local qubits and flying photonic qubits and (2) auxiliary qubits, referred to as *communication qubits* hereafter, used primarily to temporarily store the quantum state information for remote gate operations. It is worth noting that when two nodes are connected, each equipped with N computing qubits, via classical means, the dimension of the combined Hilbert space is, at most, $2 \times 2^N = 2^{(N+1)}$. This is significantly smaller than the 2^{2N} dimension achievable by an ideal DQC system with these two nodes.

In most conventional DQC architectures, each local quantum computer is equipped with additional communication qubits dedicated to establishing remote entanglement links.^{13,14} The presence of these communication qubits substantially escalates the cost of individual local quantum computer nodes. Furthermore, in these schemes, remote gate operations based on entanglement are post-selected via measurement, rendering them inherently non-deterministic. In this study, we introduce a novel DQC architecture based on *Shared Quantum Gate Processing Unit* (S-QGPU), built upon a recent breakthrough concept known as the hybrid two-qubit gate module.¹⁶ This module not only performs transduction between local qubits and photonic qubits but also facilitates controlled gate operations. In the S-QGPU-based architecture, nodes are interconnected to a centralized shared collection of these hybrid two-qubit gate modules, where all remote gate operations are executed. In contrast to the conventional architectures, the implementation of S-QGPU leads to a significant reduction in the costs associated with local quantum computing nodes. Moreover, the remote gate operations conducted within the S-QGPU framework are deterministic, eliminating the inherent non-determinism found in traditional DQC architectures.

While the S-QGPU DQC was inspired by recent advances in the neutral atom platform,¹⁶ the architecture is intentionally hardware-agnostic, making it compatible with various quantum computing technologies. Implementing the S-QGPU requires efficient quantum transducers at both local nodes and centralized shared gate modules to facilitate seamless conversion between local computing qubits and flying photonic qubits. In superconducting platforms, long-distance interconnections can employ optical-wavelength transducers,^{17,18} while microwave transducers¹⁹ suffice for short-distance connections. Similarly, for trapped-ion-based DQC, transducers that convert between ion qubits and photonic qubits are essential.²⁰ For neutral atoms, the required transduction processes are detailed in Ref. 16. Establishing a standardized wavelength for interconnecting photonic qubits across platforms would further enhance the universality of the

S-QGPU, enabling it to serve diverse quantum computing nodes with varying hardware implementations.

This article is structured as follows. In Sec. II, we describe the DQC architecture based on S-QGPU. In Sec. III, we analyze the cost of S-QGPU-based scheme with full pairing capacity, as a comparison with that of the corresponding conventional scheme as well as a monolithic quantum computer. In Sec. IV, we extend our cost analysis to a more general case where not all the computing qubits are remotely paired. In Sec. V, we evaluate and compare the performances of S-QGPU-based and conventional architectures when the inter-node communication pattern is burst. Finally, we conclude in Sec. VI.

II. ARCHITECTURE MODELS

To describe our proposed S-QGPU-based architecture for DQC, we first look at the conventional entanglement-communication-based scheme, as illustrated in Fig. 1(a). In this setup, each local node is equipped with not only computing qubits but also communication qubits for supporting remote gate operations. To perform a remote (2-qubit) controlled gate operation involving the first computing qubit, denoted by p on node 1, and the second computing qubit, denoted by q , on node 2, we first entangle a communication qubit, p' , on node 1 with a communication qubit, q' , on node 2, shown as the dashed lines in Fig. 1(a). Then, the remote gate operation between the two computing qubits p and q can be accomplished with local operations involving p and p' on nodes 1, followed by transmission of the Bell state measurement of the resulting p' from node 1 to node 2 using a classical channel, and finally, another local operation involving q and q' on node 2 based on the received measurement result. In this architecture, the nodes need to be interconnected with each other via a quantum network consisting of links and switches to route (or distribute) entanglement. In addition, establishing entanglement between two nodes typically involves transduction at both the sources that generate entangled photon pairs, and each (destination) node that stores one of the entangled photons using a communication qubit. This entanglement-communication-based DQC architecture can be implemented with either the widely recognized Cat-entangler scheme or the teleportation scheme,^{13,14} which are inherently non-deterministic as their remote gate operations are post-selected based on measurement.

Our proposed S-QGPU-based architecture is depicted in Fig. 1(b). In this new architecture, each node has computing qubits but no communication qubits are needed to support remote gate operations. Instead, it uses a collection of hybrid two-qubit gates¹⁶ called S-QGPU as shared resources to support remote gate operations. All the nodes are connected to the centralized S-QGPU, and transduction is performed at all the nodes and the S-QGPU. More specifically, to perform a remote gate operation involving computing qubit p on node 1 and computing qubit q on node 2, we first convert p and q into flying photonic qubits (a process often known as quantum transduction), which are subsequently transmitted to any one of many hybrid two-qubit gate modules, available at S-QGPU for processing. Upon completion of the remote gate operation, the module (in S-QGPU) then returns the resulting two qubits (also as photons) to nodes 1 and 2, respectively. In addition, the module will be reset, so that it can be used to perform another remote gate operation involving any two computing qubits on any two nodes in the future. In S-QGPU-based architecture, the remote gate operations are deterministic.

Note that S-QGPU-based DQC architecture requires (1) efficient conversion (transduction) between local computing qubits and flying

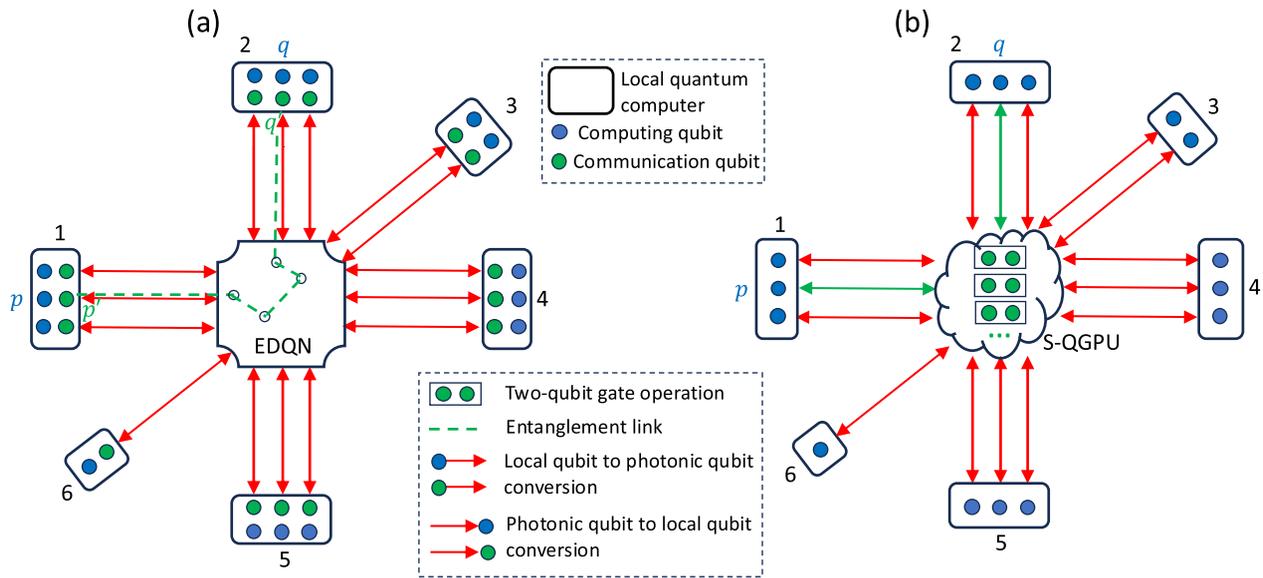


Fig. 1. Two distributed quantum computing architectures. (a) Conventional entanglement-communication-based architecture where each node has both computing qubits and communication qubits, and the nodes are interconnected via an entanglement distribution quantum network (EDQN). The green-color dashed lines represent established entanglement links. (b) S-QGPU-based architecture, where all communication qubits, paired into a collection of two-qubit gate units, are shared by the nodes which have only computing qubits. The green-color lines represent established photon channels.

photonic qubits and (2) each two-qubit gate module capable of performing a complete set of 4×4 unitary operations. While we are not limiting system implementation to any particular platform, the atom-photon hybrid modules proposed by Oh *et al.*¹⁶ could meet these requirements. In addition, from a technological point of view, S-QGPU requires more advanced implementation than the conventional entanglement-communication-based architecture, and in particular, the qubits in the hybrid modules are more functional than typical communications qubits. Nevertheless, for ease of presentation, we still treat the qubits in the hybrid modules to be the same as the communication qubits in the following discussions.

In this work, we will compare the proposed S-QGPU-based DQC with the conventional entanglement-communication-based architecture in terms of cost and performance. We show that for an ideal system with M nodes, each having N computing qubits, and total having MN communication qubits to accommodate full pairing capacity, S-QGPU-based system results in a lower cost when M and N are sufficiently large. This is because the dependence of the cost of a local quantum computer on its number of qubits is nonlinear, while the cost of S-QGPU is linear on its number of modules. The cost of transducers, quantum communication channels, and optical switches have linear or polynomial cost functions of M and N .

In addition, we also show that when not all MN computing qubits need to be engaged in remote gate operations at the same time, S-QGPU-based architecture can realize a more significant reduction in the number of communication qubits than what is possible with the conventional entanglement-communication-based architecture, and accordingly, additional cost-savings on top of an already lower cost. Moreover, in terms of performance comparison, we show that when the number of communication qubits is limited (and the same) in both architectures, S-QGPU-based scheme is more efficient when

remote gate operations are unevenly distributed at either the computing qubit level or the node level, or in other words, when the pattern for inter-node communication (supporting remote gate operations) is burst. Such a more significant reduction in the number of communication qubits, and accordingly additional reduction in the cost-savings, as well as performance improvement under a bursty communication pattern, all come as a well-recognized benefit from sharing of the resources, which, in our context, refers to the communication qubits (and the hybrid two-qubit gate modules) supporting remote gate operations. We provide a detailed analysis in Secs. III–V.

III. COST COMPARISON: A SPECIAL CASE WITH FULL PAIRING

In this section, we consider a distributed system with M nodes, each having N computing qubits. To simplify the presentation, we assume that MN is an even number and consider the case where all MN computing qubits can be fully paired for remote gate operations at the same time, which requires $MN/2$ hybrid two-qubit gate modules in S-QGPU. For a fully connected local quantum computer, extending the number of computing qubits from N to $N + 1$ is not simply a matter of adding one additional qubit. It also necessitates establishing direct connections (two-qubit controlled-gate operations) to all other N qubits. Consequently, the number of connections, or the complexity of the quantum computer, grows exponentially as N increases. Assuming that the cost is proportional to the number of pairwise connections, the cost of a fully connected N -qubit local quantum computer can be estimated as $\epsilon(a^N - 1)$, where $\epsilon > 0$ and $a > 1$ are cost parameters (coefficients). Note that these parameters may evolve as technological advancements occur. With the S-QGPU being an assembly of $MN/2$ hybrid modules, each being two-qubit gates, its estimated cost is $(MN/2)\epsilon(a^2 - 1)$ (this is because the cost of each two-qubit

gate can be considered as being equivalent to that of a two-qubit node). To ensure full pairing interconnectivity of the total MN computing qubits, MN quantum communication channels are required between the local nodes and the S-QGPU, so is an $MN \rightarrow MN$ optical switch. The quantum communication channel cost, including conversion between local qubits and photonic qubits, is approximated as MNb , with $b > 0$ indicating the cost for each quantum channel. The cost of an $MN \rightarrow MN$ optical switch, which is purely classical, is estimated as $(MN)^2 d$ where $d > 0$ is the cost per switching path. The overall cost, denoted by C_S , can be formulated as follows:

$$C_S(M, N) = M\varepsilon(a^N - 1) + \frac{1}{2}MN\varepsilon(a^2 - 1) + MNb + (MN)^2 d. \quad (1)$$

For the conventional entanglement-communication-based architecture shown in Fig. 1(a), it requires N communication qubits per node. Because each node has $2N$ qubits to accommodate both computation and communication, the cost of a system, denoted by C_E , is given by

$$C_E(M, N) = M\varepsilon(a^{2N} - 1) + MNb + (MN)^2 d. \quad (2)$$

For reference, the cost of building a monolithic fully connected local quantum computer with MN qubits, denoted by C_0 , is

$$C_0(M, N) = \varepsilon(a^{MN} - 1). \quad (3)$$

From the above-mentioned equations, it is evident that despite the fact that the overall cost of any approach is an exponential function of N , both S-QGPU-based and conventional entanglement-communication-based systems have a cost that is polynomial to the number of nodes, M , whereas the monolithic computer has a cost that is exponential to M as well. Accordingly, when building a system with a large number of computing qubits (and thus, M could be large), both distributed approaches will have a lower cost than the monolithic approach. Among the two distributed architectures, S-QGPU-based has a smaller exponent than the conventional architecture, and accordingly, a lower cost when N is large enough.

To quantitatively analyze and compare the costs, we take the following estimations. Assuming that a two-qubit quantum computer costs \$5000, and a 50-qubit fully connected quantum computer costs \$4 000 000, we have $\varepsilon = \$21\,476$, and $a = 1.110\,32$. For illustration purpose, we (somewhat arbitrarily) assume $b = \$10\,000$ per quantum communication channel and $d = \$100$ per optical switching path.

Let us first set $M = 6$. As depicted in Fig. 2, substantial cost disparities emerge among the three schemes: $C_S(6, N) < C_E(6, N) < C_0(6, N)$. In this case, the turning point for $C_S \simeq C_L \simeq C_0$ occurs at $N = 5$. As $N > 5$, the cost of S-QGPU-based scheme starts to reduce cost significantly as compared to the conventional scheme. It is possible to achieve 300-qubit distributed quantum computing at a cost of \$37 M.

The costs are jointly determined by both M and N , as shown in the contour plots in Fig. 3, where a number associated with each (log-scale) cost curve represents the exponent associated with the overall system cost. As can be seen, with a small N , the cost of S-QGPU-based scheme is nearly identical to that of the corresponding conventional scheme. As N increases, the cost of the nodes is significantly reduced in the S-QGPU-based architecture, and thus, it becomes more

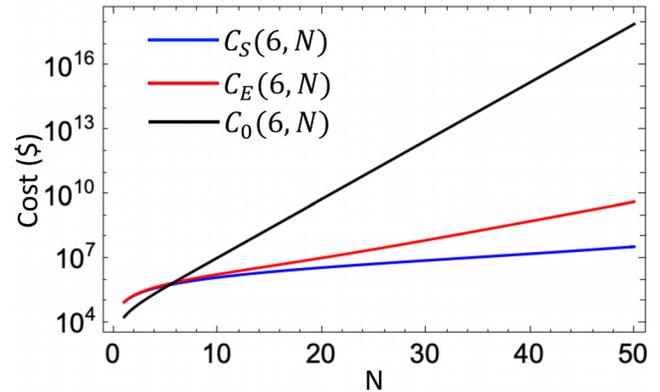


Fig. 2. Costs $C_S(6, N)$, $C_E(6, N)$, and $C_0(6, N)$ with fixed node number $M = 6$ as functions of N , the number of computing qubits at each node. Parameters: $\varepsilon = \$21\,476$, $a = 1.110\,32$, $b = \$10\,000$, and $d = \$100$.

cost-effective. At a large N , the exponential terms dominate in Eqs. (1)–(3), leading to the approximate relations

$$\frac{C_S(M, N)}{C_0(M, N)} \simeq Ma^{N-MN} \simeq Ma^{-MN}, \quad (4)$$

and

$$\frac{C_S(M, N)}{C_E(M, N)} \simeq a^{N-2N} = a^{-N}. \quad (5)$$

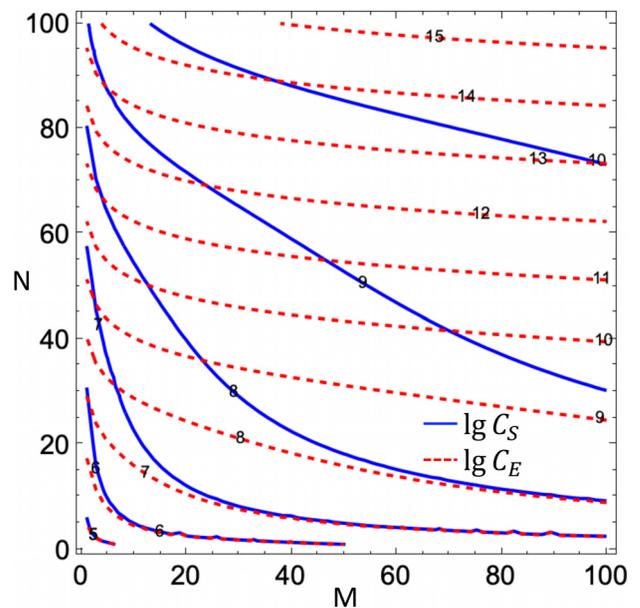


Fig. 3. Contour plots of $\lg[C_S(M, N)]$ and $\lg[C_E(M, N)]$. The solid (blue) and dashed (red) lines are equal-cost curves for $\lg C_S$ and $\lg C_E$, respectively. Parameters: $\varepsilon = \$21\,476$, $a = 1.110\,32$, $b = \$10\,000$, and $d = \$100$.

IV. COST COMPARISON: A GENERAL CASE WITH PARTIAL PAIRING

So far, we have examined a special case where the DQC system is designed with full pairing capacity such that all the MN computing qubits can be involved in some remote gate operations at the same time. In such a case, the total number of communication qubits needed, denoted by Q , is MN , and the maximum number of remote gate operations that can be supported concurrently, denoted by R , is $MN/2$, in both architectures. Essentially, at each of the M nodes, all its N computing qubits are allowed to be involved in certain remote gate operations at the same time. In this section, we examine a more general case with partial pairing, where the system is designed to support a class of applications that require that (at most) $x \leq N$ computing qubits from each of $y \leq M$ nodes be involved in remote gate operations at any given time. It is clear that in the special full pairing case studied earlier, we have $x = N$ and $y = M$. Note that when $x < N$ and/or $y < M$, not all MN computing qubits engage in remote gate operations concurrently at the same time. In S-QGPU-based architecture, we just need $R = xy/2 \leq MN/2$ two-qubit gate units with a total $2R = xy$ equivalent shared communication qubits. In the conventional entanglement-communication-based architecture, however, as we do not know which y (of M) nodes are involved in simultaneous remote gate operations, we have to assume all M nodes possibly engage and, thus, assign redundant communication qubits.

We note that not all applications require MN computing qubits be fully paired in remote gate operations at the same time. Accordingly, if none of the applications to be run on the system would require more than R pairs of remote gate operations simultaneously, then one may consider the system’s effective capacity to be large enough to create an illusion that its computing capacity is on the order of 2^{MN} .

In the following discussion, let Q_E and Q_S be the total number of communication qubits needed by the system with entanglement-communication-based and S-QGPU-based architectures, respectively. As to be shown next, when $x < N$ and/or $y < M$, we have $Q_E > Q_S$, although in the special case (where $x = N$ and $y = M$), we have $Q_E = Q_S = MN$. This means that, in general, the cost saving achieved by S-QGPU-based architecture over the conventional entanglement-communication-based architecture can be even more significant than in the special full pairing case.

Below, we discuss the general partial pairing case where $Q_E, Q_S \geq xy = 2R$ in more detail. Assuming as before that the nodes are homogeneous (i.e., having the same configuration in terms of the number of computing and communication qubits), we examine two subcases:

- Subcase E:** Evenly distributed remote gate operations. In this subcase, given at most $R = xy/2$ simultaneous remote gate operations at any given time, there are up to $y \leq M$ nodes that are engaged in concurrent remote gate operations. In addition, among these y nodes, the same number of computing qubits at each node, up to x , is involved in remote gate operation at the same time as other computing qubits at other $y - 1$ nodes. In other words, the computing qubits involved in remote gate operations are evenly distributed over the y nodes.
- Subcase U:** Unevenly distributed remote gate operations. In this subcase, for a given $R \leq N$, it is possible that at only one node, all R computing qubits are engaged in remote gate operations,

whereas the other R computing qubits are arbitrarily distributed over the remaining $y - 1$ nodes. If $R > N$, then it is possible that at one or a few nodes, all N computing qubits at each node are engaged in remote gate operations, whereas at some other nodes, fewer computing qubits are engaged in remote gate operations concurrently. In other words, the computing qubits involved in remote gate operations are unevenly distributed over the y nodes.

In subcase E, the conventional entanglement-communication-based architecture would need to dedicate x communication qubits at each node, resulting in a total of $Q_E = Mx$ communication qubits as we do not know which y nodes would engage, whereas in the proposed S-QGPU-based architecture, the total number of communication qubits needed is only $Q_S = 2R = xy$. When $y < M$, it is clear that $Q_S < Q_E$. As shown in Fig. 4(a), Q_S and Q_E follow different slopes as functions of x .

In subcase U, when $R \leq N$, the conventional entanglement-communication-based architecture would need to dedicate R communication qubits at each node, resulting in a total of $Q_E = MR$ communication qubits, whereas in the proposed S-QGPU-based architecture, the total number of communication qubits needed is still $Q_S = 2R$ only. When $R > N$, the number of communication qubits needed at each node in entanglement-communication-based architecture plateaus at N , resulting in a total of $Q_E = MN$ communication qubits, whereas in the proposed S-QGPU-based architecture, the total number of communication qubits needed remains as $Q_S = 2R$ only. The comparative results are illustrated in Fig. 4(b).

We can generalize the cost function in Eq. (1) as follows (for both subcases E and U):

$$C_s^{EU}(M, N) = \epsilon M(a^N - 1) + \epsilon R(a^2 - 1) + bMN + d(MN \cdot 2R). \tag{6}$$

Similarly, in subcase E, we can generalize Eq. (2) as follows:

$$C_E^E(M, N) = \epsilon M(a^{N+x} - 1) + bMN + d(MN \cdot Mx), \tag{7}$$

and in subcase U we can generalize Eq. (2) as follows when $R \leq N$ [note that when $R > N$, there is no change to Eq. (2)],

$$C_E^U(M, N) = \epsilon M(a^{N+R} - 1) + bMN + d(MN \cdot MR). \tag{8}$$

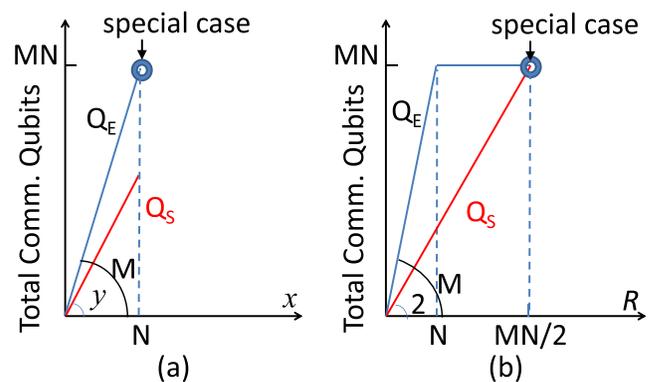


Fig. 4. Comparing the number of communications qubits needed in two architectures in the general case. (a) Subcase E, Q_E vs Q_S as functions of x . (b) Subcase U, Q_E vs Q_S as functions of R .

V. PERFORMANCE COMPARISON WITH BURST REMOTE GATE OPERATIONS

In addition to the notable size and cost scalability of the proposed S-QGPU-based architecture, this section delves into showcasing its heightened performance scalability across various ranges of bursty remote gate operations. Bursty inter-processor communication^{21–23} is prevalent in classical data centers^{24–26} and distributed computing systems.^{27–29} Recent studies^{30,31} unveiled a similar phenomenon in DQC. Prior work³⁰ defined burst communication as interactions between a single qubit and multiple nodes, examining this pattern in benchmarks, such as the quantum Fourier transform (QFT) program and the quantum approximate optimization algorithm (QAOA). Furthermore, their study highlighted how compiler optimizations can amplify such burst communication patterns. A more recent study³¹ extended the exploration of burst communication by examining inter-node communication patterns. This work analyzed benchmarks, such as the quantum ripple carry adder (RCA), the QFT algorithm, and Grover’s algorithm, further deepening our understanding of burst communication scenarios in distributed quantum systems.

Without loss of generality, in this paper, we focus on both qubit-level and node-level burstiness to study the benefit of our proposed DQC architecture. With the former, each computing qubit will engage in remote gate operations at any given time with a certain probability. With the latter, each node will engage in remote gate operations with other nodes at any given time with a certain probability. Here, when a node engages in remote gate operations, all of its computing qubits are engaged in remote gate operations. From the perspective of the quantum network that interconnects different nodes with each other in the conventional entanglement-communication-based architecture or with the S-QGPU in the proposed architecture, a higher probability at either the qubit-level or node-level represents a heavier communication load and higher degree of burstiness, resulting in a higher demand for the use of communication qubits to support remote gate operations.

With a limited number of communications qubits, it is likely that not all of the desired remote gate operations can be performed at once. Accordingly, it may take some computing qubits multiple time steps to complete its remote gate operations, resulting in a longer communication latency. In this section, we conduct simulations to evaluate the respective performance of S-QGPU-based and the conventional entanglement-communication-based architecture by assuming that they have the same number of nodes (M) which is a parameter in the simulation, with the same (and fixed) number of computing qubits per node ($N = 50$), and the same total number of communication qubits $Q_E = Q_S = 10M$.

Our assessment of performance under bursty communication patterns involves determining the latency incurred for all computing qubits to complete their requests for remote gate operations that are generated at a given time, providing profound insight into the operational prowess and efficiency of an architecture under dynamic workloads (in terms of remote gate operations). Specifically, let L_r be the average latency under realistic settings (with a limited $Q_E = Q_S = 10M$) and L_i be the latency in the ideal setting (with an unlimited number of communication qubits), we define the communication satisfaction ratio (CSR) to be

$$CSR = \frac{L_r}{L_i}. \tag{9}$$

Figure 5 gives the comparison of communication efficiency between entanglement-communication based architecture (where each of the M nodes has 10 dedicated communication qubits), and the proposed S-QGPU-based architecture (where $10M$ communication qubits are shared among the M nodes) under qubit-level burst communications. As mentioned earlier, here, the burst ratio encapsulates the likelihood of a particular computing qubit actively participating in remote gate operation during each and every discrete time step. Specifically, a local computing qubit has a probability of $x\%$ to initiate a remote gate operation with any qubits on other nodes. In the conventional entanglement-communication based architecture, such a remote gate operation leverages a communication qubit from both the source node and the target node (for creating the entanglement). In contrast, in the proposed S-QGPU shared architecture, such an operation requires a qubit pair from the shared pool.

In the simulations, we systematically vary M while keeping both N and Q . Two subfigures illustrate the results for different communication settings: Fig. 5(a)—the conventional entanglement-communication

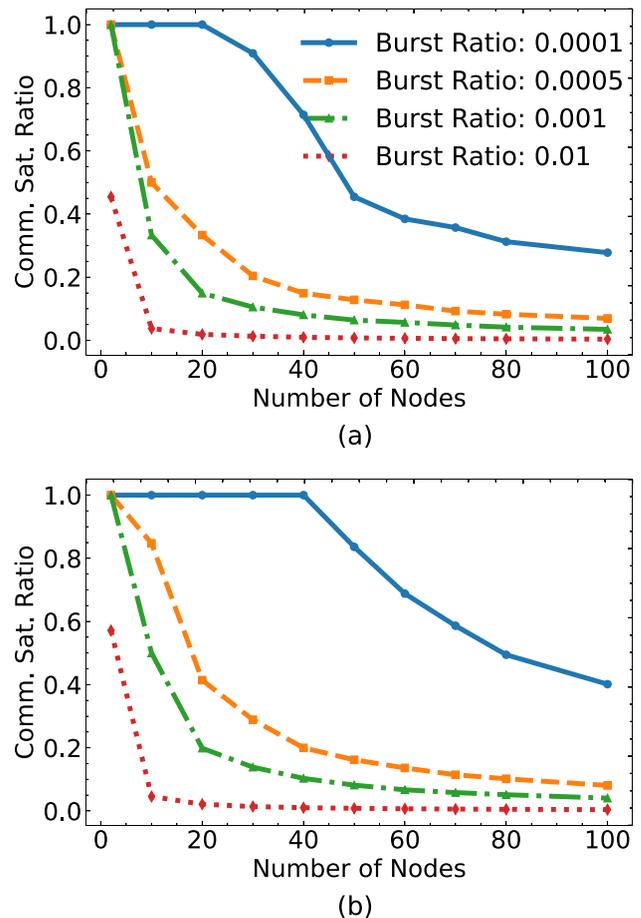


Fig. 5. Comparison of communication satisfaction ratio (CSR) between dedicated and shared qubits under qubit-level burst communication when scaling up with the number of nodes. (a) The conventional entanglement-communication-based architecture with dedicated communication qubits. (b) The S-QGPU-based architecture with shared communication qubits. Legend for (b) mirrors (a).

based architecture with dedicated communication qubits on each node and Fig. 5(b)—our S-QGPU-based architecture with a pool of two-qubit gate modules shared among all the nodes. Each curve represents a specific burst ratio at the qubit level and shows how the communication satisfaction ratio varies with the number of nodes. It is evident that our S-QGPU-based architecture consistently outperforms its counterpart. Regardless of the number of nodes considered, the S-QGPU-based architecture consistently achieves a higher CSR. Moreover, our architecture enables scaling to a greater number of nodes while maintaining a desired CSR. This performance superiority is attributed to the intrinsic efficiency derived from sharing communication qubits as resources in our S-QGPU-based architecture, manifested further in the presence of bursty remote gate operation patterns with limited resources.

Figure 6 further gives the comparison of communication efficiency between the two architectures under node-level burst communication. As mentioned earlier, the node-level burst ratio signifies the

probability of an entire node engaging in remote communication with all other nodes during discrete time steps. This abstraction captures the collective activity of a node as a whole, encompassing all the individual qubits within the node. Node-level burstiness amplifies the importance of the core advantage of sharing communication qubits across nodes, offered by the S-QGPU-based architecture. The overlapping curves in Fig. 6(a) highlight that, regardless of the burst ratio, once a single node exceeds its communication capacity, other nodes cannot share unused communication resources. This inevitably leads to a decline in the overall CSR, which is independent of the burst ratio in this specific setting. Therefore, the overlapping nature of these curves is expected and serves to illustrate the inherent limitation of systems with dedicated communication qubits. In contrast, Fig. 6(b) presents the results from our proposed approach with shared communication qubits. Sharing enables balancing communication demands across nodes at different times, significantly improving CSR under higher burst ratios. More specifically, with the same total number of communication qubits as in the case of Fig. 6(a), a higher burst ratio corresponds to a greater number of nodes that can be served effectively. Figures 6(a) and 6(b) together highlight the advantage of our shared communication qubit approach over the conventional method.

VI. CONCLUSION

In summary, we have introduced a new DQC architecture, which leverages multiple small quantum computer nodes to achieve a high quantum computing power by interconnecting them to a *shared* quantum gate processing unit (S-QGPU) that performs all remote gate operations. Unlike conventional DQC architectures based on entanglement communication, wherein remote gate operations are accomplished via teleportation or cat-entanglers,^{13,14} the proposed S-QGPU approach for remote gate operations is deterministic and does not depend on any measurement-based post selection.

The proposed S-QGPU-based architecture offers a more scalable solution with improved cost-effectiveness and operational efficiency, when the number of fully connected qubits on each quantum computer is limited by technology. Through extensive cost analysis, we have shown that S-QGPU-based DQC architecture significantly reduces both cost and resource requirements in both full and partial pairing scenarios compared to the conventional entanglement-communication-based architecture. In addition, our simulation-based performance evaluation has further revealed the superior communication efficiency of S-QGPU-based architecture in handling bursty requests for remote gate operations both at the qubit-level and the node-level. Overall, this study has demonstrated a tremendous potential to both scale up and scale out quantum computing power significantly using the S-QGPU-based architecture, though much further research still needs to be done.

Finally, we also note that although a hybrid two-qubit gate module with efficient transduction interfaces between local qubits and flying photonic qubits¹⁶ is currently being considered a key element of the envisioned S-QGPU, the proposed concept of sharing resources (such as these modules) itself to enable a novel DQC architecture applies equally well to other qubit and transduction technologies. For example, the S-QGPU architecture can also be implemented in superconductor quantum computing platform with microwave links to overcome the constraints on qubit connectivity.^{32,33}

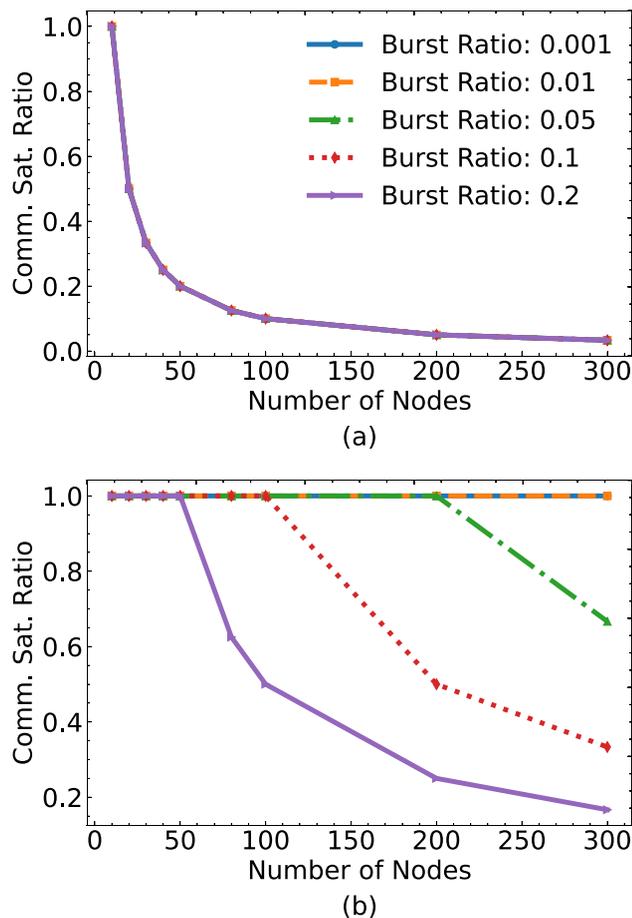


Fig. 6. Comparison of communication satisfaction ratio between dedicated and shared qubits under node-level burst communication when scaling up with the number of nodes. (a) The conventional entanglement-communication-based architecture with dedicated communication qubits. (b) The S-QGPU-based architecture with shared communication qubits. Legend for (b) mirrors (a).

ACKNOWLEDGMENTS

S.D. acknowledges the support from AFOSR (No. FA9550-22-1-0043), DOE (No. DE-SC0022069), and NSF (Nos. 2500662 and 2228725). Y.D. acknowledges the support from NSF (No. 2048144), Cisco Research, and Robert N. Noyce Trust. C.Q. acknowledges the support from NSF (No. 2403203).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Shengwang Du: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal). **Yufei Ding:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal). **Chunming Qiao:** Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, *Phys. Rev. A* **100**, 032328 (2019).
- ²J. M. Gambetta, J. M. Chow, and M. Steffen, *npj Quantum Inf.* **3**, 2 (2017).
- ³J. Gambetta, “The hardware and software for the era of quantum utility is here,” IBM Quantum Research Blog (2023), <https://www.ibm.com/quantum/blog/quantum-roadmap-2033>.
- ⁴C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, *Appl. Phys. Rev.* **6**, 021314 (2019).
- ⁵D. S. Weiss and M. Saffman, *Phys. Today* **70**(7), 44 (2017).
- ⁶L. Isenhower, E. Urban, X. L. Zhang, A. T. Gill, T. Henage, T. A. Johnson, T. G. Walker, and M. Saffman, *Phys. Rev. Lett.* **104**, 010503 (2010).
- ⁷L. Gyongyosi and S. Imre, *Sci. Rep.* **11**, 5172 (2021).
- ⁸A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, *IEEE Network* **34**, 137 (2020).
- ⁹R. Van Meter and S. J. Devitt, *Computer* **49**, 31 (2016).
- ¹⁰D. Cuomo, M. Caleffi, and A. S. Cacciapuoti, *IET Quantum Commun.* **1**, 3 (2020).
- ¹¹J. Avron, O. Casper, and I. Rozen, *Phys. Rev. A* **104**, 052404 (2021).
- ¹²Y. Zhao and C. Qiao, *IEEE/ACM Trans. Networking* **31**, 2777 (2023).
- ¹³A. Yimsiriwattana and S. J. Lomonaco, Jr., *AMS Contemp. Math.* **381**, 131 (2005).
- ¹⁴J. Eisert, K. Jacobs, P. Papadopoulos, and M. B. Plenio, *Phys. Rev. A* **62**, 052317 (2000).
- ¹⁵H. J. Kimble, *Nature* **453**, 1023 (2008).
- ¹⁶E. Oh, X. Lai, J. Wen, and S. Du, *Adv. Quantum Technol.* **6**, 2300007 (2023).
- ¹⁷M. Mirhosseini, A. Sipahigil, M. Kalaei, and O. Painter, *Nature* **588**, 599 (2020).
- ¹⁸X. Han, W. Fu, C. Zou, L. Jiang, and H. X. Tang, *Optica* **8**, 1050–1064 (2021).
- ¹⁹R. W. Andrews, R. W. Peterson, T. P. Purdy, K. Cicak, R. W. Simmonds, C. A. Regal, and K. W. Lehnert, *Nat. Phys.* **10**, 321 (2014).
- ²⁰D. Hucul, I. V. Inlek, P. Ghaemi, Q. Quraishi, and C. Monroe, *Nat. Phys.* **11**, 37 (2015).
- ²¹N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papan, and A. Vahdat, *ACM SIGCOMM Comput. Commun. Rev.* **40**, 339 (2010).
- ²²L. Xu, H. G. Perros, and G. Rouskas, *IEEE Commun. Mag.* **39**, 136 (2001).
- ²³S. B. Yoo, *J. Lightwave Technol.* **24**, 4468 (2006).
- ²⁴D. Shan, F. Ren, P. Cheng, R. Shu, and C. Guo, “Micro-burst in data centers: Observations, analysis, and mitigations,” in *IEEE 26th International Conference on Network Protocols (ICNP)* (IEEE, 2018), pp. 88–98.
- ²⁵T. Benson, A. Anand, A. Akella, and M. Zhang, *ACM SIGCOMM Comput. Commun. Rev.* **40**, 92 (2010).
- ²⁶M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, *ACM SIGCOMM Comput. Commun. Rev.* **40**, 63 (2010).
- ²⁷A. M. Alakeel and S. Arabia, *Int. J. Comput. Sci. Inf. Secur.* **10**, 153–160 (2010).
- ²⁸Y. Drougas and V. Kalogeraki, “Accommodating bursts in distributed stream processing systems,” in *IEEE International Symposium on Parallel & Distributed Processing* (IEEE, 2009), pp. 1–11.
- ²⁹N. Abramson, “The aloha system: Another alternative for computer communications,” in *Proceedings of the November 17–19, 1970, Fall Joint Computer Conference, AFIPS’70 (Fall)* (Association for Computing Machinery, 1970), pp. 281–285.
- ³⁰A. Wu, H. Zhang, G. Li, A. Shabani, Y. Xie, and Y. Ding, “Autocomm: A framework for enabling efficient communication in distributed quantum programs,” in *55th IEEE/ACM International Symposium on Microarchitecture (MICRO)* (IEEE, 2022), pp. 1027–1041.
- ³¹A. Wu, Y. Ding, and A. Li, “Qucomm: Optimizing collective communication for distributed quantum computing,” in *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture* (IEEE, 2023), pp. 479–493.
- ³²N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, *Proc. Natl. Acad. Sci. U. S. A.* **114**, 3305 (2017).
- ³³R. Stassi, M. Cirio, and F. Nori, *npj Quantum Inf.* **6**, 67 (2020).