Joint Network and Computing Resource Optimisation in Distributed Quantum Computing

Sima Bahrani*, Rui Wang, Juan Parra-Ullauri, Romerson D. Oliveira, Reza Nejabati, Dimitra Simeonidou

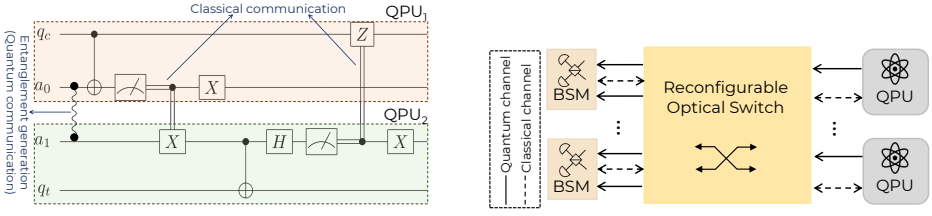
High Performance Networks Group, Merchant Venturers Building, University of Bristol, UK.
**sima.bahrani@bristol.ac.uk*

Abstract: We propose an orchestration framework to optimize network and computing resources and minimize degradation from quantum and classical communication in distributed quantum computing interconnect networks.

1. Introduction

Quantum computing is an emerging technology that holds promise for solving complex problems much faster than traditional computers. However, current quantum processors have limitations in computational power [1]. Distributed quantum computing (DQC) has been proposed as a solution to address these scalability issues [1, 2]. DQC involves partitioning and executing quantum algorithms across multiple interconnected quantum processing units (QPUs) to leverage their combined computing power. A key requirement for DQC is performing quantum operations between qubits on separate QPUs (termed a *remote gate*), which can be accomplished using entangled qubit pairs. For instance, Fig. 1a illustrates the circuit diagram for implementing a controlled-NOT (CNOT) gate between two *computing qubits*, namely q_c and q_t, positioned remotely [3]. This implies that both quantum and classical communication are necessary in a DQC interconnect network.

Operating as a distributed system, DQC requires efficient orchestration of network components through tasks like scheduling, compiling, partitioning of a quantum circuit (QC) to two or more parts (termed *circuit partitioning*), and allocation of network and computing resources (termed *resource allocation*) to different tasks. This work focuses specifically on circuit partitioning and resource allocation. Existing literature has explored aspects of these tasks. For instance, in [4, 5] circuit partitioning approaches based on graph partitioning methods have been proposed. However, these works mainly concentrate on optimizing the partitioning of individual quantum circuits into a fixed number of segments. In [6], a straightforward resource allocation method has been proposed that assumes identical QPUs. This work takes the next step by considering the joint optimization of resource allocation and circuit partitioning for short-range DQC interconnect networks. The optimization problem considers two main goals: 1) minimizing errors from quantum and classical communication delays, and 2) optimizing quantum processor utilisation. The first goal relates to enhancing computation accuracy by accounting for qubit decoherence during delays for entanglement generation and classical measurement information transmission. The second goal focuses on efficiently utilizing computing resources. To simultaneously address these two objectives, a multi-objective optimization algorithm based on mixed-integer linear programming (MILP) has been proposed. The proposed solution can significantly improve the DQC utilisation.



(a) (b)

Fig. 1: (a) Circuit diagram of remote CNOT gate. (b) Quantum network model (physical layer).

2. Network model

The quantum network model depicted in Fig. 1b is considered in this work. At the core of the network lies a reconfigurable optical switch which dynamically connects different network components such as QPUs and Bell state measurement (BSM) modules. This dynamic network configuration allows adapting the network topology to match the specific demands of quantum algorithms. The BSM modules are utilized to generate heralded entanglement within the network. In the following, the parameters associated with this network model are described.

We assume a network with J QPUs, where P quantum circuits are to be assigned to them. The sets of QPUs and QCs are denoted by $\{QPU_1, QPU_2, \ldots, QPU_J\}$ and $\{QC_1, QC_2, \ldots, QC_P\}$ respectively. The number of computing qubits in QPUs is represented by the vector $N = [n_1, \ldots, n_J]$, while the average/median of the memory decoherence time of the QPUs' qubits is denoted by $T = [t_1, \ldots, t_J]$. As for the QCs, the circuit width and the number of partitions for QCs are represented by the vectors $W = [w_1, \ldots, w_P]$ and $K = [k_1, \ldots, k_P]$, respectively.

3. Resource allocation and circuit partitioning problem

In this section, we formulate the problem of jointly optimizing resource allocation and circuit partitioning, considering the two objectives mentioned in Sec. 1.

Let us assume a specific circuit partitioning and resource allocation instance, $\mathscr{A}$, resulting in k_p partitions and $n_p^{(\mathrm{rg})}$ remote gates for QC_p, where p ranges from 1 to P. Additionally, each of these circuit partitions is assigned to a QPU. We define the matrix $X_{P\times J}$ with elements x_{pj}, where x_{pj} is the number of qubits from QC_p assigned to the QPU_j. In the following, we present the mathematical formulation of the objectives for our optimization problem.

First, we will focus on the first objective. For a qubit stored for time τ, the decoherence process can be modelled using a depolarizing noise model, as follows [7]:

$$\rho(t_0+\tau) = r(\tau,t_{\mathrm{dec}})\rho(t_0) + \frac{1-r(\tau,t_{\mathrm{dec}})}{3}(\sigma_{\mathrm{z}}\rho(t_0)\sigma_{\mathrm{z}} + \sigma_{\mathrm{x}}\rho(t_0)\sigma_{\mathrm{x}} + \sigma_{\mathrm{y}}\rho(t_0)\sigma_{\mathrm{y}}), \tag{1}$$

where t_{dec} denotes the memory decoherence time of the quantum memory, and $r(\tau,t_{\mathrm{dec}}) = 0.5(1+e^{-\tau/t_{\mathrm{dec}}})$. In the above equation, $\rho(t_0)$ and $\rho(t_0+\tau)$ are the state of the qubit at time t_0 and $t_0+\tau$, respectively. The parameters σ_{x}, σ_{y}, and σ_{z} denote the Pauli matrices. It can be inferred from Eq. (1) that the probability that the state of the qubit remains intact is equal to $r(\tau,t_{\mathrm{dec}})$. Now, we consider the quantum circuit QC_p with w_p qubits partitioned to k_p parts and assigned to the QPUs according to $\mathscr{A}$. For a single remote gate event, the probability that the latencies from entanglement generation and classical communication, denoted by T_{eg} and T_{mr}, does not alter the state of a qubit from QC_p allocated to QPU_j is given by $a_j = r(T_{\mathrm{eg}},T_j)\times(r(T_{\mathrm{mr}},T_j))^2$, where the power 2 is due to two rounds of classical data transmission, according to Fig. 1a. Taking all of the qubits of QC_p and the number of remote gates into account, the probability of no error occurring due to these delays can be written as $P_e = 1-\prod_{j=1}^{J} a_j^{x_{pj}n_p^{(rg)}}$. We define the cost corresponding to $\mathscr{A}$ for QC_p to be $C(\mathscr{A},p) = P_e$. Simplifying this cost function by utilising the logarithmic function, and taking all QCs into account, the first objective is defined as $\mathrm{obj}_1 = \sum_{p=1}^{P} n_p^{(rg)} \sum_{j=1}^{J} b_j x_{pj}$, where $b_j = -\log_2 a_j$. Next, we define the second objective. To formulate this objective function, a vector, v, with binary elements and length J is defined, where the jth element, v_j, is equal to one if $\sum_{p=1}^{P} x_{pj} > 0$. The number of used QPUs is then given by $\mathrm{obj}_2 = \sum_{j=1}^{J} v_j$.

It is worth highlighting that objective 1 and objective 2 are tightly related, as both aims align with reducing the number of remote gates. On one hand, utilizing fewer QPUs will unavoidably result in fewer remote gates. On the other hand, as we can see from the mathematical definition of objective 1, a smaller number of remote gates substantially reduces the probability of error. It is also worth noting that executing remote gates requires quantum and classical communication resources. As such, this optimization enhances network resource utilization as well.

The parameter $n_p^{(rg)}$ is complex, as it depends on the allocation vector $[x_{p1},x_{p2},\ldots,x_{PJ}]$ and the properties of QC_p. This makes the optimization problem nonlinear and intractable, exceeding polynomial time solvability. To handle the complexity, the next section proposes a method to find a good approximate solution.

4. Proposed multi-objective optimisation method

To simplify the optimization problem, a new parameter correlated with $n_p^{(rg)}$ is introduced. Specifically, a graph is constructed for QC_p, where vertices represent qubits and edges are two-qubit gates. The parameter g_p is defined as $(\mathrm{mincut}+\mathrm{maxcut})/2$, where mincut (maxcut) gives the minimum (maximum) number of crossing edges when partitioning the graph vertices to two parts. Thus, g_p characterizes the connectivity of the QC_p. Our method substitutes $n_p^{(rg)}$ with $g_p(k_p-1)$. This incorporates both the number of partitions and the nature of the quantum circuit into the formulation. With further simplifications, the multi-objective minimization problem becomes:

$$\min\{\sum_{p=1}^{P}\sum_{j=1}^{J} g_p b_j x_{pj}, \sum_{p=1}^{P} k_p, \sum_{j=1}^{J} v_j\}, \tag{2}$$

which can be solved using MILP model and ε-constraint method. Here again, to incorporate the number of partitions in our problem formulation, the matrix $E_{P\times J}$ is defined with binary elements e_{pj}, where e_{pj} is equal to one if $x_{pj} > 0$. The total number of circuit partitions, k_p, is then achieved by $\sum_{p=1}^{P}\sum_{j=1}^{J} e_{pj}$. The MILP-based formulation for this optimization problem is presented in Table I. The parameters Z and B in Table I represent the constraints on the number of partitions and the number of QPUs, respectively. Additionally, M is a positive constant that is larger than the elements of N.

The proposed algorithm has two key phases. First, the MILP formulation in Table I is used to derive a solution for matrix X, which allocates qubits from multiple QCs to individual QPUs. Next, each quantum circuit is independently partitioned to meticulously minimize the number of remote gates.

5. Evaluation and simulation results

This section evaluates the proposed resource allocation and partitioning algorithm through simulations on a 6-QPU quantum network. The QPU capacities n_j and decoherence times t_j are randomly chosen based on superconducting

quantum processor properties, with $n_j \in \{5,\ldots,27\}$ and $t_j \in \{35,\ldots,400\}$. We use benchmark quantum circuits from the Munich toolkit [8] to establish the set of QCs. This set encompasses four distinct QC types: Quantum Fourier Transform (QFT), Deutsch-Jozsa (DJ), Variational Quantum Eigensolver (VQE), and GHZ state. In our evaluation, two primary scenarios are considered: 1) Six total QCs - two QFT, two DJ, one VQE, one GHZ 2) Four QCs - one of each type. The width of each QC, w, is randomly chosen from $\{8,\ldots,15\}$. The parameters T_{eg} and T_{mr} are assumed to be 10 μs and 0.4 μs respectively.

The proposed algorithm is compared to a benchmark algorithm that assigns each QC to the QPU with the most available qubits. This approach fills QPUs round-by-round. For each scenario, random selection of the QPUs and the QCs is repeated for 1000 times. In each iteration, the QCs are partitioned and allocated to the QPUs using both the proposed and benchmark algorithms. To solve the MILP problem in the proposed method, Python MIP is used.

Table 1: MILP-based formulation

$$\min \sum_{p=1}^{P}\sum_{j=1}^{J} g_p b_j x_{pj}$$

s.t.

$$\sum_{j=1}^{J} x_{pj} = w_p \quad \text{for} \quad p = 1,2,\ldots,P$$

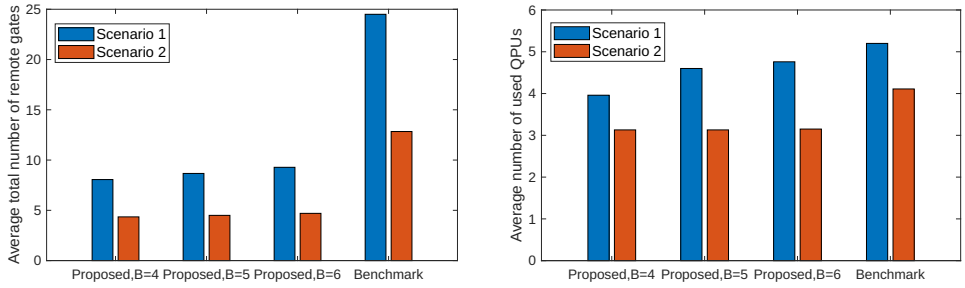
$$\sum_{p=1}^{P} x_{pj} \le n_j \quad \text{for} \quad j = 1,2,\ldots,J$$

$$0 \le x_{pj} \le M e_{pj}, \;\; e_{pj} \le x_{pj} \;\; \text{for,} \;\; p = 1,2,\ldots,P \quad j = 1,2,\ldots,J$$

$$v_j \le \sum_{p=1}^{P} x_{pj}, \;\; \sum_{p=1}^{P} x_{pj} \le M v_j \;\; \text{for} \quad j = 1,2,\ldots,J$$

$$\sum_{p=1}^{P}\sum_{j=1}^{J} e_{pj} \le Z, \;\; \sum_{j=1}^{J} v_j \le B$$

Figure 2 compares the performance of the proposed algorithm to the benchmark algorithm under two scenarios. The metrics shown are the average total number of remote gates and the average number of used QPUs. As highlighted in Sec. 3, the number of remote gates is directly related to both network resources and the errors arising from quantum and classical links. Therefore, this metric is an appropriate figure of merit for evaluating the performance of our proposed algorithm. The proposed algorithm is evaluated with three different constraints on the maximum number of QPUs (B = 4, 5, and 6). The results demonstrate that the proposed algorithm reduces the average total number of remote gates substantially compared to the benchmark in both scenarios. Additionally, the proposed algorithm utilizes fewer QPUs on average in all cases. By tuning the constraint on QPU usage, the proposed approach achieves significant improvements in both remote gate count and QPU utilization over the benchmark.



(a) (b)

Fig. 2: Simulation results for the proposed and benchmark algorithms and different scenarios. (a) Average total number of remote gates. (b) Average number of used QPUs.

6. Conclusion

This work addressed resource allocation and circuit partitioning for DQC interconnect networks. The proposed MILP-based approach aims to optimize two key metrics: 1) errors from quantum and classical communication and 2) QPU usage. Simulations demonstrate the method's ability to significantly enhance QPU utilization and network resources compared to benchmarks, as well as reducing the errors arising from quantum and classical links. The proposed algorithm provides an important advance towards realizing efficient DQC networks.

Acknowledgement

We acknowledge the funding support from the Quantum Communication Hub (EP/T001011/1).

References

1. M. Caleffi *et al.*, "Distributed quantum computing: a survey," *arXiv preprint arXiv:2212.10609*, 2022.
2. J. Ang *et al.*, "Architectures for multinode superconducting quantum computers," *arXiv preprint arXiv:2212.06167*, 2022.
3. A. Yimsiriwattana *et al.*, "Distributed quantum computing: A distributed shor algorithm," vol. 5436. SPIE, 2004, pp. 360–372.
4. O. Daei *et al.*, "Optimized quantum circuit partitioning," *International Journal of Theoretical Physics*, vol. 59, no. 12, pp. 3804–3820, 2020.
5. P. Andres-Martinez *et al.*, "Automated distribution of quantum circuits via hypergraph partitioning," *Physical Review A*, vol. 100, no. 3, p. 032308, 2019.
6. R. Parekh *et al.*, "Quantum algorithms and simulation for parallel and distributed quantum computing," in *2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS)*. IEEE, 2021, pp. 9–19.
7. G. C. Lorenzo *et al.*, "Finite-key analysis for memory-assisted decoy-state quantum key distribution," *New Journal of Physics*, vol. 22, no. 10, p. 103005, 2020.
8. https://www.cda.cit.tum.de/mqtbench/.