EE 522 Capstone: A Reference Implementation for a Quantum Message Passing Interface (QMPI)

STUDENTS: Yue Shi, Tommy Nguyen

Distributed Quantum Computing & Quantum MPI

- **Scalability:** Distributed Quantum Computing and QMPI can be used to connect multiple quantum processors to overcome the difficulty of finite qubit resources, especially in the NISQ era where noise is correlated with qubit number
- **Interoperability**: QMPI is an important step for creating a standard communication protocol between different quantum computers. This ensures interoperability between quantum devices from different hardware, just as MPI does for classical computers. This is vital in developing a global quantum internet, where quantum computers can communicate and collaborate effectively
- **Quantum Advantage:** Distributed Quantum Computing and QMPI are crucial for harnessing the full potential of quantum computers. By distributing quantum resources, latency can be decreased for parallelable computations, thereby enhancing the quantum advantage

Quantum Message Passing Interface (QMPI)

The QMPI is a communication framework for distributed quantum computation inspired by the classical MPI used in high-performance classical computing. We have developed the QMPI framework in Qiskit and built an API for users

Point to Point Communication

Based on remote EPR pairs, there are two communication protocols for inter-node communication: *Cat_Comm* and *TP_Comm*. Below illustrates how to use these two schemes to implement a remote cx gate with source qubit q_0 residing in quantum node A and destination qubit q_0' in node B with EPR pair q_{c0} and q_{c0}' [2][3]

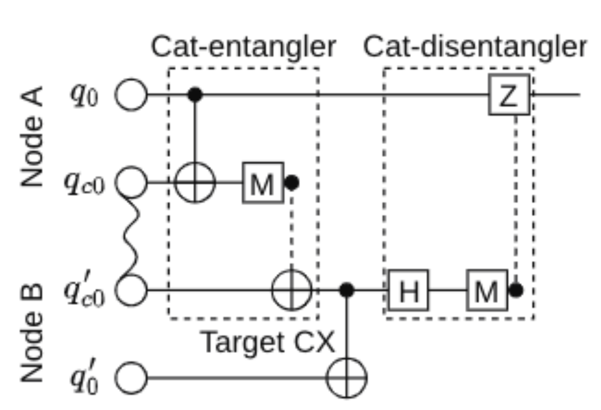


Fig. 1. Cat_Comm protocol [2]

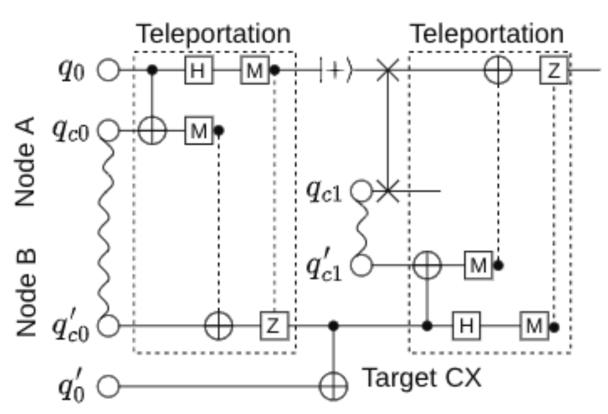


Fig. 2. TP_Comm protocol [2]

- *Send* and *Unsend* are implemented with *TP_Comm* using 2 buffered EPR pairs for each node
- *Receive* is implemented using feedback from the classical measurements in *TP_Comm* for blocking functionality
- We designed an interface to replace any quantum circuit with remote multi-qubit gates into one that uses QMPI's point to point communication

Collective Communication

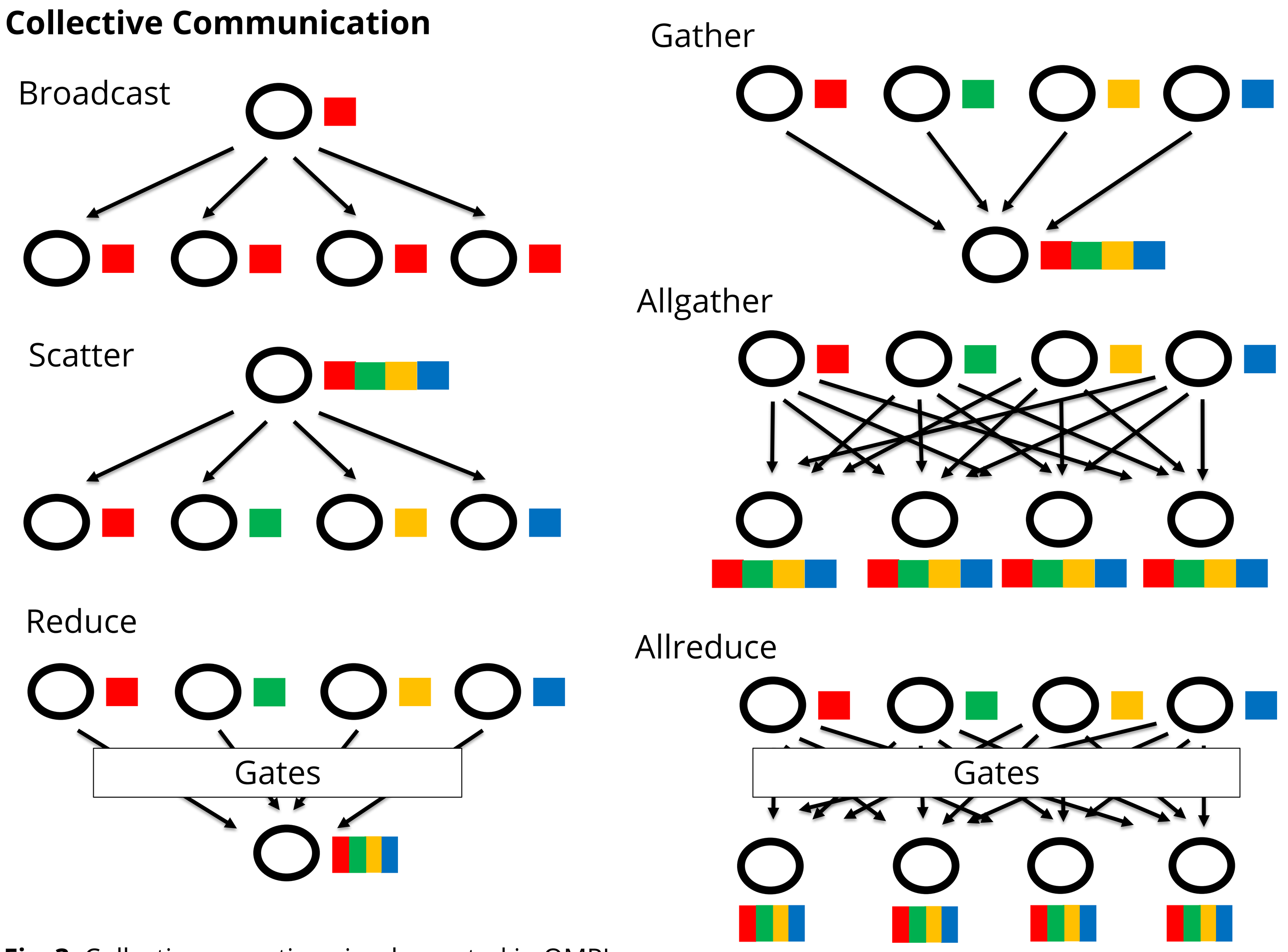


Fig. 3. Collective operations implemented in QMPI

Potential Applications:

- Transverse Ising Model – *Broadcast, Scatter*
- Quantum Phase Estimation – *Reduce/Gather*
- Quantum Key Distribution – *Allgather*
- VQE/QAOA – *Scatter, Reduce*
- Quantum Arithmetic – *Allreduce*

A Strategy for improving Collective Communication Efficiency

On top of the *Auto_Comm* and *Coll_Comm* frameworks [2][3], we develop a new strategy for quantum circuits with MCT (multi-controlled Toffoli) gates. This can reduce resource consumption and improve the fidelity of collective communication in distributed quantum systems.

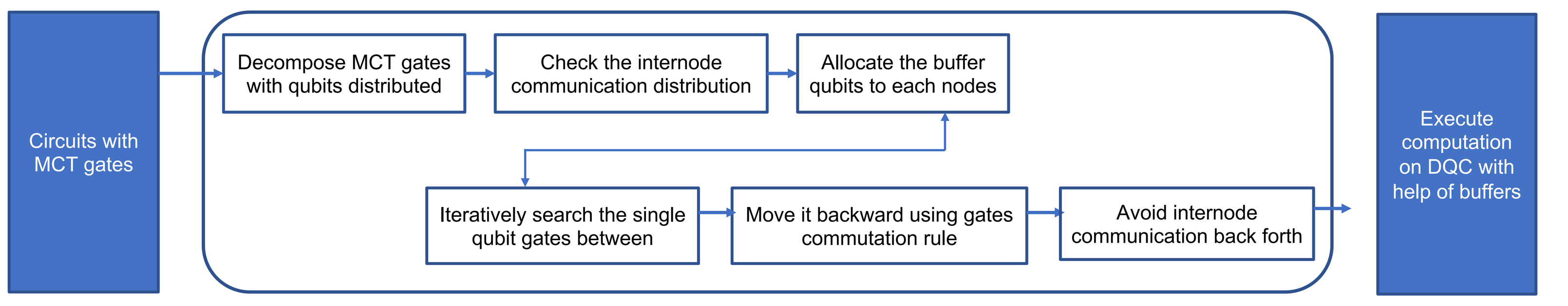


Fig. 4. Schematic of MCT strategy

Multi-Controlled Toffoli (MCT) Experimental Setup

- Platform: simulations implemented on QASM simulator
- Benchmark circuit: Function rd32 circuit imported from RevLib (as shown below) [4]
- Noise model: Built according to the parameters in the Coll_Comm paper [3]
- Baseline: Coll_Comm DQC compiler

Operation	Latency	Fidelity
Single-qubit gates	$t_{1q} \sim 0.1$ CX	$f_{1q} \approx 99.99\%$
CX and CZ gates	$t_{2q} = 1$ CX	$f_{2q} \approx 99.80\%$
Measure	$t_{ms} \sim 5$ CX	$f_{ms} \approx 99.60\%$
Intra-cluster EPR preparation	$t_{iep} \sim 12$ CX	$f_{iep} \approx 98\%$
Intra-cluster classical comm	$t_{icb} \sim 1$ CX	—
Inter-cluster EPR preparation	$t_{oep} \sim 1000$ CX	$f_{oep} \approx 90\%$
Inter-cluster classical comm	$t_{ocb} \sim 100$ CX	—

Fig. 5. Latency and fidelities of gates [2]

- Metrics: total number of consumed EPR pairs and fidelity of computation

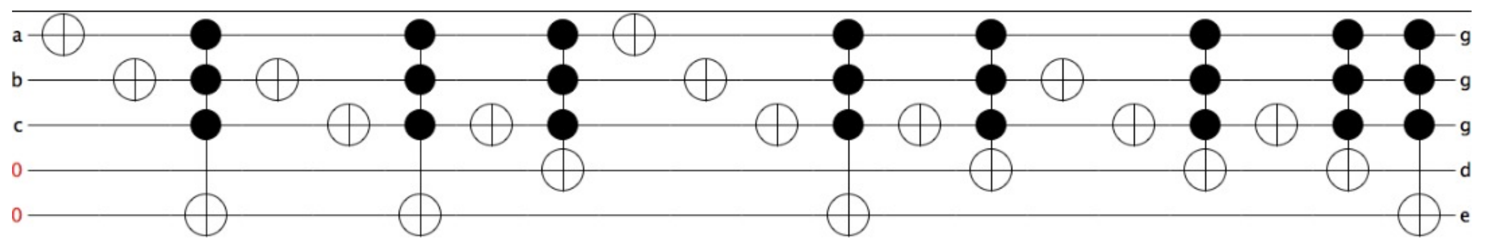
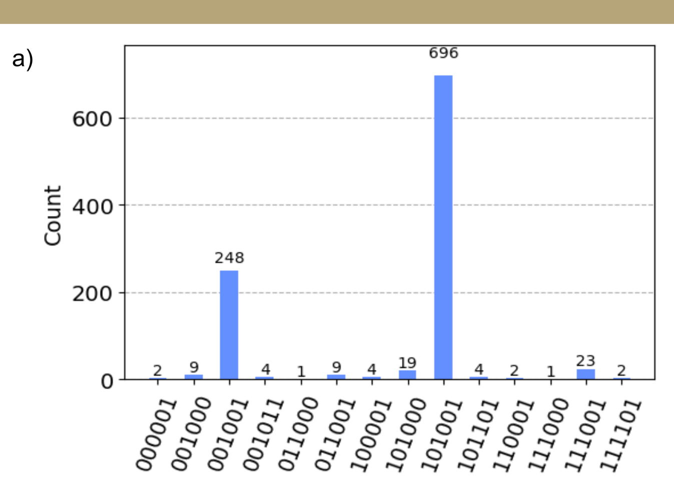


Fig. 6. Benchmark circuit rd32 (adder) taken from RevLib [4]

Simulation Results

- Our MCT technique produces the correct result with 68% fidelity when simulating the benchmark circuit and uses 5 EPR pairs. The fidelity greatly improved over the baseline (44%), and the usage of remote EPR pairs significantly decreased.
- The MCT strategy works for other quantum circuits with MCT gates with improvements on the distributed quantum system.
- We implemented Quantum Phase Estimation with Gather and the Trotterization of the Transverse Ising Model with Scatter to test our QMPI framework

a)



b)

metrics / methods	EPR pair cost	Fidelity
Using buffers	14	0.442
Optimized	5	0.680

Fig. 7. (a) Simulation result using our MCT technique, (b) and metrics comparation

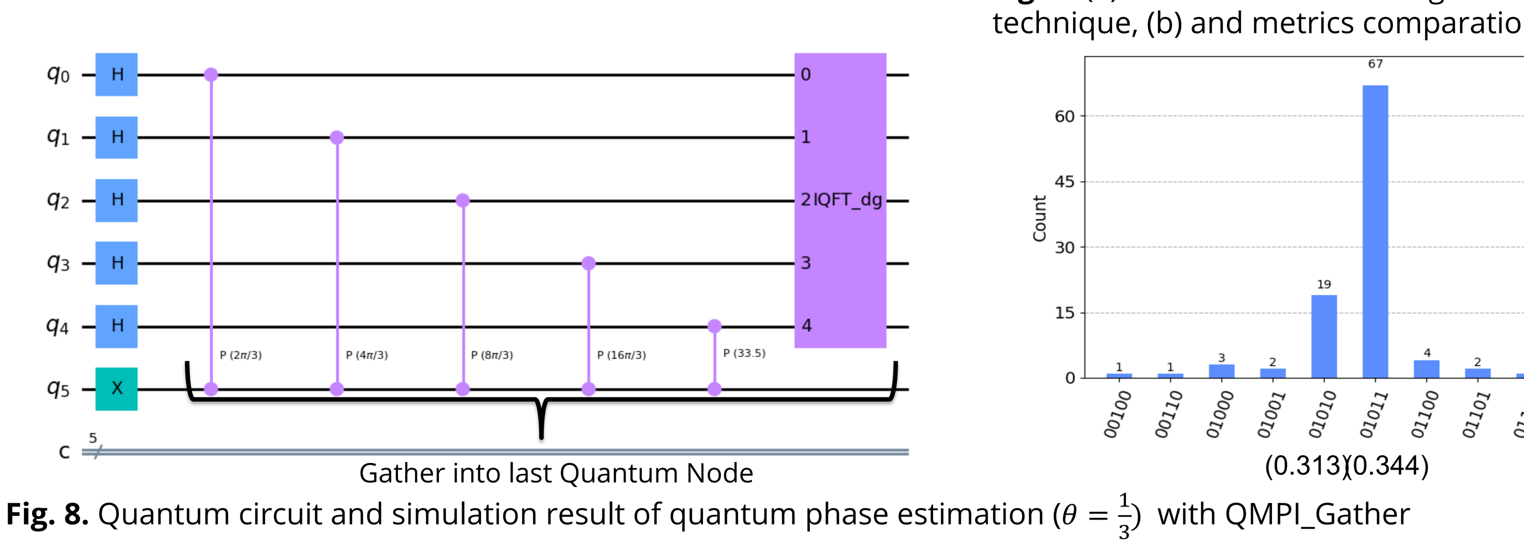


Fig. 8. Quantum circuit and simulation result of quantum phase estimation ($\theta = \frac{1}{3}$) with QMPI_Gather

Future Work, References, and Acknowledgments

1. Develop a compiler combining our MCT strategy with the QMPI framework
2. Implement more applications within the QMPI framework and MCT and compare fidelities, latencies, and # of EPR pairs
3. Further optimizations using strategies within *Auto_Comm* and *Coll_Comm* that would be automatic within QMPI
4. Generalize QMPI for quantum architectures that doesn't assume all-to-all connectivity

Acknowledgement

This material is based upon work supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Co-design Center for Quantum Advantage (C 2QA) under contract number DESC0012704. The Pacific Northwest National Laboratory is operated by Battelle for the U.S. Department of Energy under contract DE-AC05-76RL01830.

References

[1] Thomas Haner et. al., Distributed quantum computing with QMPI. Proceedings of international conference for high performance computing, network, storage and analysis. November 2021, No.16.

[2] Anbang Wu et. al., AutoComm: A Framework for enabling efficient communication in distributed quantum program. 55th IEEE/ACM International Symposium on Microarchitecture (MICRO), Chicago, IL, USA, 2022, pp. 1027–1041.

[3] Anbang Wu et. al., CollComm: Enabling efficient collective quantum communication based on EPR buffering. arXiv: 2208.06724v2

[4] https://www.revlib.org

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

ADVISERS: Ang Li, Samuel Stein, Tim Stavenger, Marvin Warner (PNNL); Martin Roetteler (Microsoft); Sara Mouradian, Peter J. Pauzauskie (UW)

SPONSOR: ELECTRICAL & COMPUTER ENGINEERING DEPARTMENT, UNIVERSITY OF WASHINGTON, PACIFIC NORTHWEST NATIONAL LABS

PNNL-SA-185584