2025 IEEE International Conference on Cluster Computing Workshops (CLUSTER Workshops) | 979-8-3315-1256-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/CLUSTERWorkshops65972.2025.11164201

NetQMPI: a practical MPI-inspired library for distributed quantum computing over NetQASM SDK

F. Javier Cardama[1] and Tomás F. Pena[1,2]
[1]*Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS)*
[2]*Departamento de Electrónica e Computación (DEC) - Universidade de Santiago de Compostela (USC)*

***Abstract*—Distributed quantum computing (DQC) is emerging as a practical solution to overcome the scalability limits of individual quantum processors by interconnecting them via quantum networks. However, programming such systems remains a significant challenge due to the need to coordinate quantum operations, manage entanglement, and synchronize classical communication. This paper proposes NetQMPI as a Python library that enables high-level programming of distributed quantum applications by building on top of two key foundations: NetQASM, which abstracts the physical network layer to interface with quantum nodes and simulators such as NetSquid and SimulaQron; and NetQIR, which proposes a layered abstraction model suitable for defining a quantum version of the classical Message Passing Interface (MPI). NetQMPI provides an MPI-style programming model tailored to quantum systems. It introduces primitives such as `qsend` and `qrecv` that abstract quantum communication details like entanglement generation, and protocols. Collective operations such as `qscatter` or `qgather` are also included. This allows developers to focus on distributed algorithm design while relying on NetQMPI to handle low-level operations efficiently.**

***Index Terms*—distributed quantum computing, intermediate representation, high-performance computing, quantum networks, mpi**

I. Introduction

Quantum computing has experienced substantial progress over the past decade, driven by advances in quantum hardware, control systems, and algorithm design [1]. Nevertheless, the development of scalable quantum algorithms remains severely constrained by the limited number of qubits available in current quantum processing units (QPUs). Even the most advanced quantum hardware platforms today offer only a few hundred noisy qubits, imposing stringent limits on algorithmic depth and practical problem sizes [2].

Distributed Quantum Computing (DQC) has emerged as a compelling architectural paradigm to address these scalability limitations. By interconnecting multiple QPUs via quantum networks capable of distributing entanglement and enabling inter-node quantum operations, DQC allows a collection of smaller devices to operate as a logically larger system. This model relies on both quantum and classical channels to coordinate computation, teleport quantum states, and share measurement outcomes between spatially separated nodes [3].

While conceptually powerful, programming DQC systems remains a formidable challenge. Developers must contend with the full complexity of the quantum network stack, including entanglement generation protocols, classical synchronization of distributed processes, and intricate physical-layer behaviors. These difficulties underscore the need for robust software abstractions that decouple algorithmic logic from network operations, enabling developers to design distributed applications without requiring low-level quantum network engineering [4].

In this paper, we introduce NetQMPI, a Python-based library designed to fill this gap. NetQMPI is built top the NetQASM SDK, which provides a low-level instruction set architecture (ISA) for programming quantum networks [5]. While NetQASM offers a socket-like interface to request entanglement, issue gate instructions, and coordinate communication between nodes, NetQMPI defines a higher-level message-passing interface inspired by classical MPI. This abstraction enables developers to structure distributed quantum programs using familiar constructs such as ranks, `qsend`/`qrecv` operations while relying on the NetQASM layer to manage entanglement generation, teleportation, and classical messaging transparently.

II. NetQMPI Overview

Executing distributed quantum algorithms requires quantum networks capable of transferring quantum states and distributing entangled resources across nodes. To support scalable software development in this setting, several abstraction layers have been proposed. NetQASM provides a low-level instruction set architecture (ISA) for quantum network programming, offering modular support for entanglement generation, classical control, and synchronization. Its backend-agnostic design allows deployment on simulators such as NetSquid [6] or SquidASM [7] without requiring users to handle low-level physical modeling.

Building on this foundation, NetQIR [8] introduces a distributed quantum intermediate representation that decouples algorithmic, compiler, and communication concerns. In parallel, QMPI [9] proposes a message-passing model inspired by classical MPI. However, one of its key limitations is that it is not quantum-inspired in a realistic sense: it includes collective primitives such as broadcast or allgather that violate the no-cloning theorem and thus cannot be physically implemented. NetQMPI addresses this by offering a practical and executable quantum message-passing interface compatible with the NetQASM stack and grounded in valid quantum information-theoretic operations.

III. NetQMPI: Design and Features

NetQMPI addresses this need by realizing an MPI-style programming interface that respects quantum mechanical constraints while leveraging the full power of the NetQASM execution model. NetQMPI supports teleportation-based communication primitives such as qsend and qrecv, enabling the transfer of quantum states between nodes via shared entanglement and classical messaging. It follows a Single Program, Multiple Data (SPMD) approach where a single logical script is annotated with rank-based conditionals, and NetQMPI compiles this into individual NetQASM programs per node. This design aligns with the NetQIR architectural vision while being executable on current NetQASM-based simulation stacks. The implementation of this library can be found open-source in its GitHub[1] repository or by installing it with `pip`[2].

Importantly, NetQMPI provides a concrete implementation of the high-level abstractions proposed by NetQIR, while retaining compatibility with the modular, backend-independent philosophy of NetQASM. As simulators like SquidASM evolve to expose richer performance and fidelity metrics, NetQMPI stands as a portable and extensible foundation for developing and benchmarking distributed quantum applications in a scalable and architecture-neutral way. The following are some of its features:

- **Communication Abstractions:** NetQMPI provides `qsend` and `qrecv` functions that abstract the process of sending and receiving quantum information between nodes. It also provides other functions such as `qscatter` or `qgather`.
- **Process Management:** It manages process ranks and communication contexts, facilitating the development of scalable quantum applications.
- **Simulator Integration:** NetQMPI is compatible with NetQASM-supported simulators like NetSquid and SimulaQron, allowing for testing and validation of quantum network applications.

IV. Example of use: qsend and qrecv

To illustrate the most basic functionality of NetQMPI's capabilities, a simple program involving two quantum nodes has been implemented and showed in the Figure 1. Node 0 is responsible for creating qubit with a given state (in this case $|+\rangle$), which it then sends to node 1 using a `qsend`. Node 1 will have to execute a `qrecv` to receive the information. It is important to note that, due to the non-cloning theorem, the qubit in node 0 has been destroyed and, therefore, no longer belongs to that node.

[1]https://github.com/NetQIR/net-qmpi

[2]`pip install netqmpi`

```python
def main(COMM_WORLD: QMPICommunicator = None):
    rank = COMM_WORLD.get_rank()

    with COMM_WORLD.connection:
        if rank == 0:
            q = Qubit(COMM_WORLD.connection)
            q.H()
            COMM_WORLD.qsend([q], 1)
        else:
            [qubit_recv] = COMM_WORLD.qrecv(0)
            measurement = qubit_recv.measure()
            COMM_WORLD.connection.flush()
            print(f"measure: {measurement}")
```

Fig. 1. Example of using `qsend` and `qrecv` with NetQMPI

V. Conclusions

NetQMPI simplifies the development of distributed quantum applications by providing high-level abstractions over NetQASM SDK. These high-level abstractions are based on proposals already studied in the literature such as NetQIR or QMPI. By handling the complexities of quantum communication internally, it enables developers to focus on application logic, accelerating the development of scalable quantum software. Future work includes extending NetQMPI's functionality and integrating it with emerging quantum hardware platforms.

References

[1] M. A. Serrano, J. A. Cruz-Lemus, R. Perez-Castillo, and M. Piattini, "Quantum software components and platforms: Overview and quality assessment," *ACM Computing Surveys*, vol. 55, p. 164, 12 2022. [Online]. Available: https://dl.acm.org/doi/10.1145/3548679

[2] M. Fellous-Asiani and et al., "Limitations in quantum computing from resource constraints," *PRX Quantum*, vol. 2, p. 040335, Nov 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PRXQuantum.2.040335

[3] D. Barral, F. J. Cardama, and et al., "Review of distributed quantum computing: From single qpu to high performance quantum computing," *Computer Science Review*, vol. 57, p. 100747, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013725000231

[4] J. Illiano and et al., "Quantum internet protocol stack: A comprehensive survey," *Computer Networks*, vol. 213, p. 109092, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622002250

[5] A. Dahlberg and et al., "NetQASM—a low-level instruction set architecture for hybrid quantum–classical programs in a quantum internet," *Quantum Science and Technology*, vol. 7, p. 035023, 6 2022. [Online]. Available: https://iopscience.iop.org/article/10.1088/2058-9565/ac753fhttps://iopscience.iop.org/article/10.1088/2058-9565/ac753f/meta

[6] T. Coopmans, R. Knegjens, and et al., "Netsquid, a network simulator for quantum information using discrete events," *Communications Physics 2021 4:1*, vol. 4, pp. 1–15, 7 2021. [Online]. Available: https://www.nature.com/articles/s42005-021-00647-8

[7] B. van der Vecht, S. Wehner, and A. Dahlberg, "SquidASM github repository," 2023.

[8] F. J. Cardama, J. Vázquez-Pérez, and et al., "NetQIR: An extension of QIR for Distributed Quantum Computing," 2025. [Online]. Available: https://arxiv.org/abs/2408.03712

[9] T. Häner and et al., "Distributed quantum computing with QMPI," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3458817.3476172