Implementing Quantum Virtual Machines for Enhanced Resource Utilization in Quantum Computing

Aaron Bordeaux[1], *Runzhou Tao*[2], *Hongzheng Zhu*[2], *Jason Nieh*[2]
[1]Department of Computer Science, Princeton University, Princeton, NJ, USA
[2]Department of Computer Science, Columbia University, New York, NY, USA

Abstract: This paper introduces HyperQ, a novel system designed to enhance the efficiency and performance of quantum cloud computing by implementing quantum virtual machines (qVMs). We integrated HyperQ into the IBM Quantum Platform and evaluated its effectiveness from provider and user perspectives. Our evaluation focused on throughput, resource utilization, latency reduction, and fidelity maintenance improvements. Using a subset of QASMBench benchmarks, we demonstrated that HyperQ significantly increases throughput and utilization by up to six and seven times, respectively while reducing latency by a similar factor. Importantly, these improvements do not compromise the fidelity of quantum program execution. HyperQ's ability to perform both space and time scheduling of quantum workloads showcases its potential to effectively optimize near-term quantum computing resources. Our results indicate that HyperQ is a scalable and practical solution for enhancing quantum cloud services, providing substantial performance gains without requiring significant changes to existing quantum hardware or compilation tools.

Methods: To evaluate HyperQ, we implemented it on the IBM Quantum Platform and conducted experiments using small and medium benchmarks from QASMBench. We created two benchmark sets: "small-only," containing small circuits for fidelity evaluation, and "small&med," combining small and medium circuits for assessing scheduling performance. Each benchmark simulated real-world workloads by randomly queuing program workloads multiple times. We tested three configurations: the standard IBM Quantum Platform, HyperQ with only space scheduling, and HyperQ with both space and time scheduling. The experiments were conducted on IBM's Brisbane quantum computer with a 127-qubit Eagle chip, using a desktop PC for compilation, qVM scheduling, and post-processing. We measured throughput, utilization, latency, and fidelity to assess HyperQ's performance improvements.

Results: The implementation of HyperQ on the IBM Quantum Platform demonstrated significant improvements across all measured metrics. HyperQ increased throughput by up to six times compared to the baseline IBM Quantum Platform, with better performance for small-only benchmarks due to the efficient simultaneous execution of smaller qVMs. Utilization rates also improved, showing a near sevenfold increase for small-only benchmarks and a fivefold increase for small&med benchmarks. HyperQ reduced average program latency by up to six times, primarily by decreasing queue waiting times. Despite the added overhead of qVM scheduling, the latency improvements remained substantial. Fidelity measurements indicated that HyperQ introduced negligible additional noise, maintaining high fidelity comparable to the baseline IBM Quantum approach, with all programs staying well below the L1 distance threshold of 1. In summary, HyperQ achieved higher throughput, better utilization, reduced latency, and maintained high fidelity in quantum program execution.

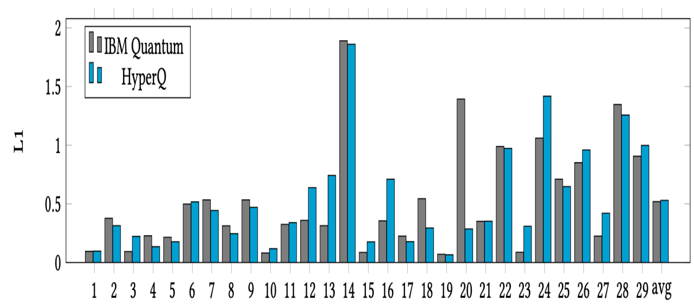

Figure 10. Fidelity Comparison

Conclusion: HyperQ successfully introduces quantum virtual machines to enhance the efficiency of quantum cloud computing. By implementing HyperQ on the IBM Quantum Platform, we achieved significant improvements in throughput, utilization, and latency, while maintaining high fidelity in program execution. These results demonstrate HyperQ's potential to optimize quantum computing resources, making it a promising solution for the growing demands of quantum cloud services.

Acknowledgment: I thank Columbia University SURE for the opportunity to conduct this research. Special thanks to Tiffany A. Moore, Dr. Jason Nieh, and my mentors Runzhou Tao and Hongzheng Zhu for their invaluable guidance and support.

References:

Poulami Das, Swamit S Tannu, Prashant J Nair, and Moinuddin Qureshi. A case for multi-programming quantum computers. In Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, pages 291–303, 2019.

Siyuan Niu and Aida Todri-Sanial. Enabling multi-programming mechanism for quantum computing in the nisq era. Quantum, 7:925, 20