Multi-programming Cross Platform Benchmarking for Quantum Computing Hardware

Siyuan Niu, Aida Todri-Sanial

HAL Id: lirmm-03690212

https://hal-lirmm.ccsd.cnrs.fr/lirmm-03690212v1

Submitted on 8 Jun 2022

Multi-programming Cross Platform Benchmarking for Quantum Computing Hardware

Siyuan Niu

LIRMM, University of Montpellier
161 Rue Ada, Montpellier, France
siyuan.niu@lirmm.fr

Aida Todri-Sanial

LIRMM, University of Montpellier, CNRS
161 Rue Ada, Montpellier, France
aida.todri@lirmm.fr

Abstract—With the rapid development of quantum hardware technologies, benchmarking the performance of quantum computers has become attractive. In this paper, we propose a new aspect of benchmarking quantum computers by evaluating the limitation of hardware utilization using a multi-programming mechanism – a technique that simultaneously executes multiple circuits in a quantum machine. This is the first attempt to compare the evaluation of multi-programming on trapped-ion and superconducting devices. Based on the experimental results, performing multi-programming on a trapped-ion device demonstrates better results than a superconducting machine without losing any fidelity to independent executions.

Index Terms—Multi-programming, NISQ evaluation, Benchmarking

I. Introduction

Quantum computing is a rapidly growing research field in recent years. Various quantum platforms have been developed based on different technologies, for example, superconducting, trapped-ion, photonics, neutral-atom, etc. The two leading technologies are superconducting and trapped-ion devices. Users can access the real quantum machines through online services released by companies such as Google, IBM, or Quantinuum.

The near-term quantum machines with at most around 100 qubits are qualified as Noisy Intermediate-Scale Quantum (NISQ) hardware [7]. The hardware topology is limited and the quantum operations are prone to errors. Even with these hardware constraints, the fast improvement of quantum hardware leads to two frequently asked questions: (1) Which metric to use to characterize the performance of a quantum computer? (2) How to use a quantum computer more efficiently given the currently available hardware resources while obtaining reliable results?

To address the first question, quantum volume (QV) [1] is one of the most used metrics to benchmark NISQ devices proposed by IBM. The best QV that IBM quantum machine obtained is 128, whereas the trapped-ion device produced by Quantinuum was announced to achieve 4096 for QV. Other single-number metrics such as Q-Score [3] or application-based benchmarks such as SupermarQ [8] have also been developed. As for the second question, the multi-programming

mechanism was introduced for superconducting and DWAVE machines by enabling parallel circuits or QUBO executions on one quantum chip simultaneously [2], [5]. The multi-programming mechanism can help improve hardware utilization, reduce the overall problem-solving time, and even save costs for charging quantum machines.

In our paper, we combine the two questions by benchmarking the hardware ability to enable multi-programming mechanism across two different platforms: a superconducting device from IBM and a trapped-ion device from Quantinuum. First, we compare the characteristics of the two machines. Second, we evaluate the multi-programming mechanism and report the fidelity difference compared with the standalone circuit execution mode.

II. Comparison of Quantum Hardware Characteristics

TABLE I
CHARACTERISTICS OF THE TWO QUANTUM DEVICES.

Vendors	IBM	Quantinuum
Backend name	ibmq_mumbai	H1-2
Technology	Superconducting	Trapped-ion
Qubits	27	12
QV	128	4096
Gate sets	Rz, SX, X, CX	U_{1q}, Rz, ZZ
1Q Error (%)	0.02	0.01
2Q Error (%)	4.5	0.35
RO Error (%)	2.9	0.4
Topology	Nearest-neighbor	All-to-all
Parallel two-qubit operation	All connected qubits	3 parallel zones

We compare ibmq_mumbai, a 27-qubit IBM quantum machine with the largest QV, with a 12-qubit System Model H1-2 Quantinuum device. Their characteristics are shown in Table I. H1-2 machine demonstrates more reliable quantum operations compared to ibmq_mumbai, the 1Q, 2Q, and readout error being improved by 2x, 12.9x, and 7.2x. Moreover, the QCCD architecture [6] used by the H1-2 machine enables multiple interaction zones so that the parallel two-qubit operations become possible.

III. Multi-programming Mechanism Benchmarking

The multi-programming mechanism was first proposed to apply to the superconducting device because of its relatively large number of qubits. The fidelities of circuits are decreased due to a limited number of reliable qubits and the higher probability of crosstalk. Even though the size of the current trapped ion device is relatively small, it is still interesting to investigate if the same issues of crosstalk and non-uniform reliable qubits occur in the trapped-ion device. Additionally, since all of the online services for trapped-ion machines are charged per circuit, executing multiple circuits in one execution can save budget as well. In this section, we first execute two circuits on both devices in parallel to report the fidelity difference between simultaneous and standalone executions. Second, we simultaneously apply QAOA twice to evaluate its performance in solving the Max-Cut problem using multi-programming mechanism.

A. Simultaneous Executions of Two Circuits

TABLE II
INFORMATION OF BENCHMARKS.

ID	Benchmark	Qubits	Gates	CX
1	BV3	3	9	2
2	BV4	4	12	3
3	peres_3	3	16	7
4	toffoli	3	15	6
5	3_17_13	3	36	17
6	4mod5-v1_22	5	21	11
7	mod5mils_65	5	35	16
8	alu-v0_27	5	36	17
9	decod24-v2_43	4	52	22

The benchmarks used to evaluate the performance of multi-programming are collected from the state of the art [2], [4], shown in Table II. For ibmq_mumbai, we apply QuMC [4] multi-programming algorithm and set the optimization level of qiskit transpiler to the highest level of three. The version of Qiskit is 0.25.0 and the number of shots is set to 8192. Whereas for Quantinuum H1-2, the transpilation process is not accessible by users and we only set the number of shots to 100 to maintain a reasonable budget. Moreover, we use the metric Probability of Successful Trials (PST), defined as the number of trials giving the correct answer divided by the total number of trials, to represent the fidelity of the circuit.

The results of executing two benchmarks simultaneously or independently are shown in Fig. 1. In general, Quantinuum H1-2 machine obtains more reliable results than ibmq_mumbai. The simultaneous executions on ibmq_mumbai (labeled as ibmq_mul) decrease the fidelities by 3.4% compared to standalone executions (labeled as ibmq_indp). Whereas for Quantinuum H1-2, similar results are obtained
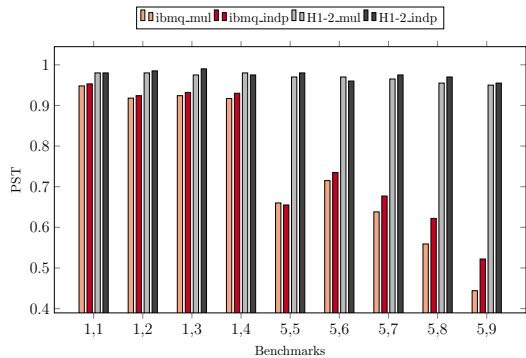

Fig. 1. Results of executing two benchmarks simultaneously and independently on ibmq_mumbai and Quantinuum H1-2.

comparing simultaneous (labeled as H1-2_mul) with independent executions (labeled as H1-2_indp), with only a difference of 0.5% on fidelities, but the budget of simultaneous executions is reduced by 31%. Therefore, enabling multi-programming on Quantinuum trapped-ion devices can improve the hardware utilization, reduce the total circuit runtime (execution time + waiting time), and save the budget without losing fidelity, which also demonstrates the low impact of crosstalk.

B. Simultaneous Executions of QAOA

Inspired by the parallel executions of multiple QUBO problems on DWAVE machine [5], we perform QAOA algorithm twice on the two machines at the same time and evaluate its performance in solving Max-Cut problem.

For demonstration, we construct an QAOA ansatz for a 4-node rectangle graph and optimize the parameters using *COBYLA* optimizer. We execute two of the same circuits with optimized parameters on the two quantum chips. For Quantinuum H1-2, if we execute the QAOA ansatz once independently, the probabilities of obtaining the correct answers "0101" and "1010" are the highest in the probability distribution, and the sum of the two probabilities is 50%. When executing two of the same ansatz in parallel, both of the ansatz circuits are able to have the highest probabilities for the correct answers, the sum being 64% and 47%, respectively. Also, the budget is reduced by 30%. Whereas for ibmq_mumbai, executing QAOA ansatz once independently and twice simultaneously obtain correct answers with highest probabilities as well, the sum being 42.5% (independent), 38.1%, and 40.3% (the last two for simultaneous executions).

IV. Conclusion

Cross platform benchmarking becomes an active research topic with the rapid improvement of different quantum technologies. Various metrics have been proposed to characterize the performance of a quantum computer. We propose a novel aspect of evaluating the hardware limitation by performing multi-programming mechanism, a technique that enables to execute multiple circuits in parallel to improve the hardware utilization and reduce the total circuit runtime. We first execute

two small benchmarks on IBM superconducting quantum chip and Quantinuum trapped-ion machine. Second, we run two QAOA ansatz circuits in parallel on the two machines to solve the Max-Cut problem. Based on the results, we found that trapped-ion devices are more suitable for multi-programming mechanism, without losing fidelities compared with independent executions, and the cost budget is reduced significantly.

REFERENCES

[1] Andrew W Cross, Lev S Bishop, Sarah Sheldon, Paul D Nation, and Jay M Gambetta. Validating quantum computers using randomized model circuits. *Physical Review A*, 100(3):032328, 2019.

[2] 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.

[3] Simon Martiel, Thomas Ayral, and Cyril Allouche. Benchmarking quantum coprocessors in an application-centric, hardware-agnostic, and scalable way. *IEEE Transactions on Quantum Engineering*, 2:1–11, 2021.

[4] Siyuan Niu and Aida Todri-Sanial. Enabling multi-programming mechanism for quantum computing in the nisq era. *arXiv preprint arXiv:2102.05321*, 2021.

[5] Elijah Pelofske, Georg Hahn, and Hristo N Djidjev. Parallel quantum annealing. *Scientific Reports*, 12(1):1–11, 2022.

[6] Juan M Pino, Jennifer M Dreiling, Caroline Figgatt, John P Gaebler, Steven A Moses, MS Allman, CH Baldwin, M Foss-Feig, D Hayes, K Mayer, et al. Demonstration of the trapped-ion quantum ccd computer architecture. *Nature*, 592(7853):209–213, 2021.

[7] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[8] Teague Tomesh, Pranav Gokhale, Victory Omole, Gokul Subramanian Ravi, Kaitlin N. Smith, Joshua Viszlai, Xin-Chuan Wu, Nikos Hardavellas, Margaret R. Martonosi, and Frederic T. Chong. SupermarQ: A scalable quantum benchmark suite.