ILP-Based Scheduling for Linear-Tape Model Trapped-Ion Quantum Computers

Extended Abstract

Xin-Chuan Wu
University of Chicago
xinchuan@uchicago.edu

Yongshan Ding
University of Chicago
yongshan@uchicago.edu

Yunong Shi
University of Chicago
yunong@uchicago.edu

Yuri Alexeev
Argonne National Laboratory
yuri@alcf.anl.gov

Hal Finkel
Argonne National Laboratory
hfinkel@anl.gov

Kibaek Kim
Argonne National Laboratory
kimk@anl.gov

Frederic T. Chong
University of Chicago
Argonne National Laboratory
chong@cs.uchicago.edu

ABSTRACT

Quantum computing (QC) is emerging as a potential post-Moore high-performance computing (HPC) technology. Trapped-ion quantum bits (qubits) are among the most leading technologies to reach scalable quantum computers that would solve certain problems beyond the capabilities of even the largest classical supercomputers. In trapped-ion QC, qubits can physically move on the ion trap. The state-of-the-art architecture, linear-tape model, only requires a few laser beams to interact with the entire qubits by physically moving the interacting ions to the execution zone. Since the laser beams are limited resources, the ion chain movement and quantum gate scheduling are critical for the circuit latency. To harness the emerging architecture, we present our mathematical model for scheduling the qubit movements and quantum gates in order to minimize the circuit latency. In our experiment, our scheduling reduces 29.47% circuit latency on average. The results suggest classical HPC would further improve the quantum circuit optimization.

KEYWORDS

Quantum Computing, Post-Moore High-Performance Computing, Emerging Architecture, Integer Linear Programming, AMPL, Trapped-Ion Qubits.

1 INTRODUCTION

Quantum Computing (QC) aims to solve certain computational problems beyond the capabilities of even the largest classical high-performance computing (HPC). By leveraging quantum mechanical principles (superposition and entanglement), QC algorithms demonstrate potential to revolutionize areas such as machine learning [1], quantum chemistry [8, 14], and cryptography [15]. Thus, QC is expected as a post-Moore HPC technology.

Trapped-ion technologies are among the most promising systems for practical QC. In trapped-ion QC, an ion is the physical expression of a quantum bit (qubit), and each ion can be physically shuttled over trap surfaces [6, 10–13]. Acousto-optic modulators

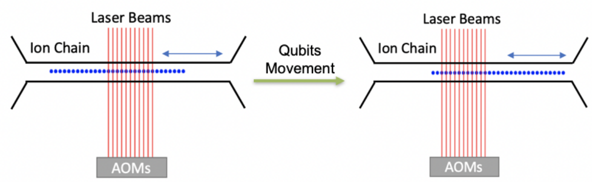

Figure 1: Linear-tape model trapped-ion quantum computers.

(AOMs) generate the laser beams to perform single-qubit rotation gates and XX-gates between arbitrary pairs of qubit ions [5, 16].

In the traditional trapped-ion architecture, since each ion requires a dedicated laser beam, the number of laser beams must be equal to the number of ions. The scalability of trapped-ion QC is then limited by the complexity of modulating the frequency and amplitude of each laser beam.

Linear-tape model is the state-of-the-art technology to achieve scalable QC [7]. This architecture only requires a few laser beams to interact with the entire qubits on the linear ion crystal by physically moving the target ions to the execution zone, as Figure 1. The ions are trapped in a linear chain. Changing the voltages can transport the ion chain [2, 9]. Since only a part of the ions can be located in the range of laser beams at each time step, the entire ion chain will physically move back and forth during the execution of quantum circuits.

This architecture brings interesting scheduling problems (Figure 2) that are similar to classical disk I/O scheduling problems. In that setting, we have existing scheduling algorithms, FIFO and elevator. However, since we have multiple laser beams and two-qubit gates in our QC scheduling problem, FIFO and elevator cannot fit our situation to give the optimal solution for minimizing the circuit latency.

In this paper, we propose our integer linear programming (ILP) approach for scheduling ion movement and quantum gates to support the linear-tape model trapped-ion QC architecture. Our scheduling algorithm is called STRIQC (Scheduling for TRapped-Ion QC).

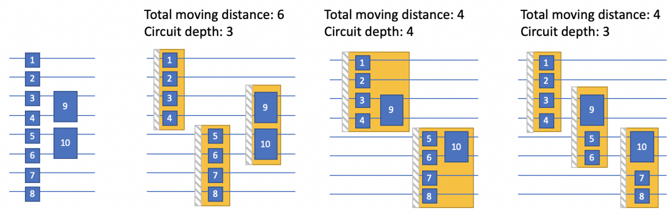

Figure 2: An example of scheduling problem for linear-tape trapped-ion machine.

2 STRIQC

2.1 Notations

Let G be the set of gates. Let T be the set of program time steps. For $g \in G$, we denote q_g to express applying the gate g on the qubit q. For two-qubit gates, q_g is the control qubit, and q'_g is the target qubit. To denote gate dependency, we use a binary relation $\prec$ on the gates. For two gates $g_1 \prec g_2$ if g_2 depends on g_1. For $t \in T, g \in T$, we employ a binary variable $c_{t,g} = 1$ if g is scheduled at t. Otherwise, $c_{t,g} = 0$. For $t \in T$, the moving distance of the ion chain at time t is denoted by m_t, and h_t is the laser head position. t_g and t_s are constant parameters for gate time and unit shuttling time, respectively. n is the number of qubits, and k is the number of laser beams, and d is the circuit depth.

2.2 ILP Model

Objective:

$$\text{Minimize } d \times t_g + \sum_{i=1}^{d}(m_i \times t_s) \tag{1}$$

Constraints:

$$\forall g \in G, \sum_{t \in T} c_{t,g} = 1 \tag{2}$$

$$\forall g_1 \prec g_2 \in G, \sum_{t \in T}(c_{t,g_1} \times t) < \sum_{t \in T}(c_{t,g_2} \times t) \tag{3}$$

$$\forall t \in T, g \in G, h_t \le c_{t,g} \times min(q_g, q'_g) + (1 - c_{t,g}) \times (n - k + 1) \tag{4}$$

$$\forall t \in T, g \in G, c_{t,g} \times (h_t + k - 1) \ge c_{t,g} \times max(q_g, q'_g) \tag{5}$$

$$\forall g \in G, d \ge \sum_{t \in T}(c_{t,g} \times t) \tag{6}$$

$$\forall t \in T, -m_t \le h_t - t_{t-1} \le m_t \tag{7}$$

We implement our model with AMPL [4], and STRIQC is integrated as a part of the QC compilation process. For existing quantum applications, solving the scheduling problem for the whole circuit at once may take long time to finish the optimization. To compile the circuit within a reasonable period, we realize the optimized compilation by dividing the circuit into multiple small circuit blocks, and solve the scheduling problem for each small circuit, and then merge the results as the final executable quantum assembly.

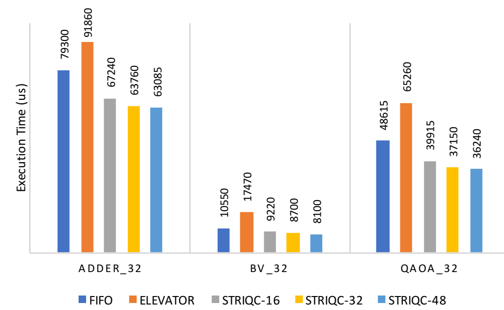

Figure 3: Circuit execution time.

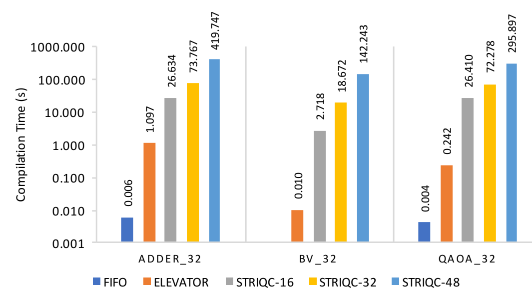

Figure 4: Circuit compilation time.

3 RESULTS

We use Bernstein-Vazirani, quantum approximate optimization algorithm (QAOA), adder, and quantum fourier transform (QFT) as our benchmarks. Each of them uses 32 qubits in the program.

We evaluate the overall circuit execution time on our timing model simulation. We set each gate time as $100\mu s$ [3], and the unit shuttling time is $5\mu s$ [9]. We assume the device is a 32-qubit machine with 16 laser beams.

Figure 3 shows the circuit execution time (the lower is better). We perform STRIQC with different size of circuit blocks. STRIQC-N means the circuit is divided into multiple N-gate blocks, and then we apply STRIQC to schedule the circuit block. We compare STRIQC with FIFO and elevator algorithms. The results show that STRIQC gives a better scheduling of ion movements and gate operations in terms of circuit latency. If we solve the circuits with larger blocks, we can get more efficient scheduling. However, it takes longer time to finish the compilation. Figure 4 shows the compilation time for different size of circuit blocks.

4 CONCLUSION

Linear-tape model is a novel trapped-ion QC architecture. To support this emerging QC (post-Moore HPC) architecture, we propose an ILP-based scheduling for quantum circuits. STRIQC generates the optimal sequence of ion movement operations and quantum gates in terms of latency. With larger circuit block, STRIQC can generate the better scheduling. This indicates that we will get benefits from classical HPC systems to achieve optimal compilation.

5 ACKNOWLEDGMENTS

This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. This research was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations - the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, to support the nation's exascale computing imperative. The material was supported by the U.S. Department of Energy, Office of Science, and supported by the National Science Foundation under Grant No. 1619253. This work is funded in part by EPiQC, an NSF Expedition in Computing, under grant CCF-1730449. This work is also funded in part by NSF PHY-1818914.

REFERENCES

[1] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. 2017. Quantum machine learning. *Nature* 549, 7671 (2017), 195.

[2] RB Blakestad, C Ospelkaus, AP VanDevender, JM Amini, J Britton, D Leibfried, and David J Wineland. 2009. High-fidelity transport of trapped-ion qubits through an X-junction trap array. *Physical review letters* 102, 15 (2009), 153002.

[3] Colin D Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M Sage. 2019. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews* 6, 2 (2019), 021314.

[4] Robert Fourer, David M Gay, and Brian W Kernighan. 1993. AMPL. A modeling language for mathematical programming. (1993).

[5] Nikodem Grzesiak, Reinhold Blümel, Kristin Beck, Kenneth Wright, Vandiver Chaplin, Jason M Amini, Neal C Pisenti, Shantanu Debnath, Jwo-Sy Chen, and Yunseong Nam. 2019. Efficient Arbitrary Simultaneously Entangling Gates on a trapped-ion quantum computer. *arXiv preprint arXiv:1905.09294* (2019).

[6] Hartmut Häffner, Christian F Roos, and Rainer Blatt. 2008. Quantum computing with trapped ions. *Physics reports* 469, 4 (2008), 155–203.

[7] IONQ. 2018. IonQ harnesses single-atom qubits to build the world's most powerful quantum computer. https://ionq.co/. (2018). Online.

[8] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. 2017. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* 549, 7671 (2017), 242.

[9] H Kaufmann, T Ruster, Christian Tomás Schmiegelow, Marcelo Alejandro Luda, V Kaushal, J Schulz, D Von Lindenfels, F Schmidt-Kaler, and UG Poschinger. 2017. Scalable creation of long-lived multipartite entanglement. *Physical review letters* 119, 15 (2017), 150503.

[10] David Kielpinski, Chris Monroe, and David J Wineland. 2002. Architecture for a large-scale ion-trap quantum computer. *Nature* 417, 6890 (2002), 709.

[11] Tzvetan S Metodi, Arvin I Faruque, and Frederic T Chong. 2011. Quantum computing for computer architects. *Synthesis Lectures on Computer Architecture* 6, 1 (2011), 1–203.

[12] Tzvetan S Metodi, Darshan D Thaker, Andrew W Cross, Frederic T Chong, and Isaac L Chuang. 2005. A quantum logic array microarchitecture: Scalable quantum data movement and computation. In *38th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'05)*. IEEE, 12–pp.

[13] Michael A Nielsen and Isaac Chuang. 2002. Quantum computation and quantum information. (2002).

[14] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L OâĂŹbrien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nature communications* 5 (2014), 4213.

[15] Peter W Shor. 1999. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review* 41, 2 (1999), 303–332.

[16] K Wright, KM Beck, S Debnath, JM Amini, Y Nam, N Grzesiak, J-S Chen, NC Pisenti, M Chmielewski, C Collins, et al. 2019. Benchmarking an 11-qubit quantum computer. *arXiv preprint arXiv:1903.08181* (2019).