Guiding Compiler Optimizations for Neutral Atom Quantum Computers Through Visualizations

Yannick Stade*, and Robert Wille*†
*Chair for Design Automation, Technical University of Munich, Munich, Germany
†MQSC, Garching near Munich, Germany
yannick.stade@tum.de, robert.wille@tum.de
www.cda.cit.tum.de/research/quantum

***Abstract*—The scale of Neutral Atom (NA) quantum computers requires automated compilation tools. Designing the required heuristic methods demands a deep understanding of complex hardware trade-offs, for which visualizations can provide crucial insights. This work introduces *NAViz*, the first publicly available app to visualize quantum computations on NA devices in real-time. A case study demonstrates how *NAViz* was instrumental in identifying and resolving inefficiencies in an existing compilation strategy, leading to a new, more performant one. The tool is available as part of the *Munich Quantum Toolkit* (MQT) at https://github.com/munich-quantum-toolkit/naviz.**

***Index Terms*—quantum computing, optimizing compilers, rydberg atoms, ultracold atoms, visualization, routing**

I. Introduction & Background

Neutral Atom (NA) quantum computers offer scalability and flexible connectivity by rearranging qubits with optical tweezers [1]. This makes *placement* (assigning atoms to sites) and *routing* (moving atoms) crucial compilation tasks [2]–[11].

The scale of current devices requires heuristic methods [8], [12]. Designing these heuristics requires deep insights, for which visualizations have proven successful in other domains [13], [14].

This work, for the first time, showcases how visualizations help to better understand and improve compilation strategies for NA devices. To this end, we introduce *NAViz*, a visualization app designed to analyze various compilation strategies and provide intuition on their effects. We then apply NAViz to a routing problem where an intuitively superior strategy paradoxically results in worse performance. NAViz reveals the cause of this inefficiency, which ultimately led to the development of a novel, better-performing strategy.

II. Visualization App

An NA quantum computation consists of a sequence of quantum gates interleaved with atom rearrangements. Minimizing rearrangement time is critical [8], [15]. Visualizing atom movements can reveal inefficiencies in routing strategies. To this end, we developed *NAViz*, a visualization app for quantum computations on NA devices, implemented in Rust to facilitate real-time rendering. NAViz takes the quantum computation and device architecture as input, with customizable appearance settings. It provides an interactive real-time visualization with adjustable speed, a timeline for detailed analysis, and an option to export animations as video files.

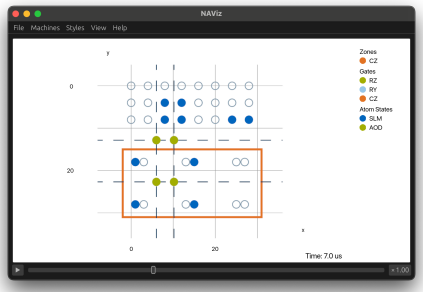

Figure 1. Screenshot of the visualization app NAViz while animating a quantum computation on an NA device.

Example 1. *Figure 1 shows a screenshot of NAViz while animating a quantum computation. In the background, gray circles depict the potential static locations for atoms on the device. The atoms themselves are shown as colored circles: blue for static atoms and green for those currently being rearranged. The timeline slider at the bottom allows the user to navigate through the time.*

III. Compilation Aspect: Relaxed Routing

The rearrangement of atoms on NA devices is facilitated by static *Spatial Light Modulator* (SLM) traps and two orthogonal *Acusto-Optic Deflectors* (AODs). These create an adjustable 2D grid of traps, where each column and row can be controlled independently [16]. To rearrange an atom, its corresponding column and row are activated picking it up, shifted to the new position, and then deactivated to drop it off. While multiple atoms can be rearranged in parallel, columns (rows) are activated/deactivated as a whole, creating complex *rearrangement constraints* that can lead to serializing rearrangements [17].

Furthermore, since during a rearrangement, columns (rows) cannot cross, the relative order of the atoms in a rearrangement cannot change. More precisely, any two atoms a_1 and a_2 can only be rearranged in parallel if they satisfy the conditions:

$$x_{a_1}^{(\text{old})} = x_{a_2}^{(\text{old})} \iff x_{a_1}^{(\text{new})} = x_{a_2}^{(\text{new})} \tag{1}$$
$$y_{a_1}^{(\text{old})} = y_{a_2}^{(\text{old})} \iff y_{a_1}^{(\text{new})} = y_{a_2}^{(\text{new})} \tag{2}$$
$$x_{a_1}^{(\text{old})} < x_{a_2}^{(\text{old})} \iff x_{a_1}^{(\text{new})} < x_{a_2}^{(\text{new})} \tag{3}$$
$$y_{a_1}^{(\text{old})} < y_{a_2}^{(\text{old})} \iff y_{a_1}^{(\text{new})} < y_{a_2}^{(\text{new})} \tag{4}$$

However, these *strict* conditions can be slightly *relaxed* by exploiting the fact that, when the first atom is picked up, it can already be moved past other atoms.

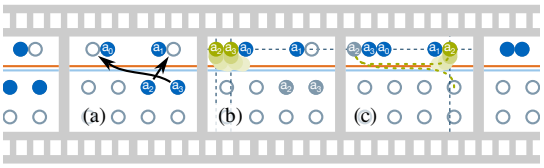

Figure 2. The black arrows in Fig. 2a indicate the desired rearrangement of two atoms. Figure 2b depicts the first movement and Fig. 2c the second. The dashed green line in Fig. 2c illustrates the distance traveled by atom a_2.

Table I. RESULTS OF DIFFERENT ROUTING STRATEGIES

Benchmark	Num. Qubits	**Strict** Rearr. T. [ms]	**Relaxed** Rearr. Time [ms]	**Hybrid** Rearr. Time [ms]
graphstate	100	22.4	↗ 26.2	↘ 22.1
	200	53.0	↗ 67.5	↘ 51.8
	500	133.8	↗ 198.5	↘ 132.8
	1000	320.5	↗ 517.4	↗ 320.9
	2000	802.7	↗ 1322.9	↘ 802.1
	5000	2588.8	↗ 4147.7	↘ 2582.3
qft	500	2324.5	↗ 2354.5	↘ 2287.1
	1000	4937.9	↗ 4986.7	↘ 4855.8
vqe two local	50	631.9	↗ 670.4	↘ 624.7
	100	2423.1	↗ 2688.8	↘ 2394.6
	200	8477.0	↗ 9757.0	↘ 8349.7
wstate	500	410.8	↘ 406.3	↘ 410.5
	1000	777.7	↘ 772.6	↘ 777.1
∅		1838.8	↗ 2147.4	↘ 1816.3

This flexibility allows dropping conditions Eqs. (3) and (4). However, as shown in Table I, this *relaxed* strategy paradoxically increased rearrangement times compared to the *strict* one.

IV. LEARNING FROM THE VISUALIZATION

To shed light onto these counter-intuitive results, we employed NAViz to visualize various rearrangements produced by the relaxed strategy. This analysis revealed inefficient detours.

Example 2. *Figure 2 depicts a scenario of relaxed routing discovered through NAViz. Here, both atoms are picked up simultaneously. Atom a_3 is moved to its target position. Meanwhile atom a_2 must (1) stay in the same row (cf. Eq. (2)), and (2) remain left of atom a_3 (cf. Eq. (3))—effectively causing a long detour for atom a_2 (dashed green line in Fig. 2c).*

Thanks to NAViz, a better *hybrid* strategy combining the strict and relaxed approaches could be derived to mitigate those inefficient detours. The strategy first computes a strict routing and then greedily merges rearrangement steps if doing so reduces the total overhead under relaxed conditions. Evaluations on a set of benchmarks [18] demonstrate that this hybrid strategy generally outperforms both others, as summarized in Table I.

V. CONCLUSIONS

We introduced *NAViz*, an open-source visualization tool for NA quantum computers. A case study demonstrated its utility in identifying compiler inefficiencies and developing a superior, hybrid routing strategy. The tool is available in open-source as part of the MQT under https://github.com/munich-quantum-toolkit/qmap/.

Acknowledgments

During the preparation of the code and manuscript, the authors used GitHub Copilot, powered by Claude's Sonnet 4.6 to improve code, spelling, grammar, clarity, and readability. Afterward, the authors reviewed and edited the content as needed. The authors take full responsibility for the final content.

The project leading to this publication has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 101001318), the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation, 563436708), and the BMFTR under grant number 01MQ25001I (FullStaQD).

REFERENCES

[1] Dolev Bluvstein et al. "Architectural mechanisms of a universal fault-tolerant quantum computer." arXiv: 2506.20661, pre-published.

[2] Tirthak Patel, Daniel Silver, and Devesh Tiwari, "GRAPHINE: Enhanced Neutral Atom Quantum Computing using Application-Specific Rydberg Atom Arrangement," in *Int'l Conf. for High Performance Computing, Networking, Storage and Analysis*, ACM, 2023. DOI: 10.1145/3581784.3607032

[3] Ludwig Schmid, Sunghye Park, and Robert Wille, "Hybrid Circuit Mapping: Leveraging the Full Spectrum of Computational Capabilities of Neutral Atom Quantum Computers," in *Design Automation Conf.*, ACM, 2024. DOI: 10.1145/3649329.3655959

[4] Daniel Silver, Tirthak Patel, and Devesh Tiwari. "Qompose: A Technique to Select Optimal Algorithm- Specific Layout for Neutral Atom Quantum Architectures." arXiv: 2409.19820.

[5] Hanrui Wang et al., "Q-Pilot: Field Programmable Qubit Array Compilation with Flying Ancillas," in *Design Automation Conf.*, ACM, 2024. DOI: 10.1145/3649329.3658470

[6] Jason Ludmir and Tirthak Patel, "PARALLAX: A Compiler for Neutral Atom Quantum Computers under Hardware Constraints," in *Int'l Conf. for High Performance Computing, Networking, Storage and Analysis*, IEEE, 2024. DOI: 10.1109/SC41406.2024.00079

[7] Daniel Bochen Tan, Wan-Hsuan Lin, and Jason Cong, "Compilation for Dynamically Field-Programmable Qubit Arrays with Efficient and Provably Near-Optimal Scheduling," in *Asia and South Pacific Design Automation Conf.*, ACM, 2025. DOI: 10.1145/3658617.3697778

[8] Yannick Stade, Wan-Hsuan Lin, Jason Cong, and Robert Wille, "Routing-aware placement for zoned neutral atom-based quantum computing," 2025. arXiv: 2505.22715.

[9] Nathan Constantinides et al. "Optimal Routing Protocols for Reconfigurable Atom Arrays." arXiv: 2411.05061.

[10] Daniel Bochen Tan, Dolev Bluvstein, Mikhail D. Lukin, and Jason Cong, "Compiling Quantum Circuits for Dynamically Field-Programmable Neutral Atoms Array Processors," *Quantum*, 2024. DOI: 10.22331/q-2024-03-14-1281

[11] Yannick Stade, Ludwig Schmid, Lukas Burgholzer, and Robert Wille, "Optimal State Preparation for Logical Arrays on Zoned Neutral Atom Quantum Computers," in *Design, Automation and Test in Europe*, 2024. DOI: 10.23919/DATE64628.2025.10993241

[12] Wan-Hsuan Lin, Daniel Bochen Tan, and Jason Cong, "Reuse-Aware Compilation for Zoned Quantum Architectures Based on Neutral Atoms," in *IEEE Int'l Symp. on High-Performance Computer Architecture*, 2025. arXiv: 2411.11784.

[13] Robert Wille, Jannis Stoppe, Eleonora Schönborn, Kamalika Datta, and Rolf Drechsler, "RevVis: Visualization of Structures and Properties in Reversible Circuits," in *Int'l Conf. of Reversible Computation*, red. by David Hutchison et al., Springer International Publishing, 2014. DOI: 10.1007/978-3-319-08494-7_9

[14] Malte Hansen, Jens Bamberg, Noe Baumann, and Wilhelm Hasselbring, "Semantic Zoom and Mini-Maps for Software Cities," in *IEEE Working Conference on Software Visualization (VISSOFT)*, IEEE, 2025. DOI: 10.1109/VISSOFT67405.2025.00013

[15] Ludwig Schmid et al., "Computational capabilities and compiler development for neutral atom quantum processors—connecting tool developers and hardware experts," *Quantum Science and Technology*, 2024. DOI: 10.1088/2058-9565/ad33ac

[16] Dolev Bluvstein et al., "A quantum processor based on coherent transport of entangled atom arrays," *Nature*, 2022. DOI: 10.1038/s41586-022-04592-6

[17] Yannick Stade, Ludwig Schmid, Lukas Burgholzer, and Robert Wille, "An Abstract Model and Efficient Routing for Logical Entangling Gates on Zoned Neutral Atom Architectures," in *Int'l Conf. on Quantum Computing and Engineering*, IEEE, 2024. DOI: 10.1109/QCE60285.2024.00098

[18] Nils Quetschlich, Lukas Burgholzer, and Robert Wille, "MQT Bench: Benchmarking Software and Design Automation Tools for Quantum Computing," *Quantum*, 2023, MQT Bench is available at https://www.cda.cit.tum.de/mqtbench/. DOI: 10.22331/q-2023-07-20-1062