Title

Introduction to the Special Issue on Software Tools for Quantum Computing: Part 2

Journal

ACM Transactions on Quantum Computing, 4(1)

ISSN

2643-6809

Authors

Alexeev, Yuri
McCaskey, Alex
de Jong, Wibe

Publication Date

2023-03-31

DOI

10.1145/3574160

Special Issue on Software Tools for Quantum Computing Part 2 – Introduction

Yuri Alexeev (yuri@anl.gov)
Computational Science Division, Argonne National Laboratory,
Alex McCaskey (amccaskey@nvidia.com),
NVIDIA,
Wibe de Jong (wadejong@lbl.gov)
Computational Science Department Applied Computing for Scientific Discovery, Lawrence Berkeley National Laboratory

Quantum computing is emerging as a remarkable technology that offers the possibility of achieving major scientific breakthroughs in many areas. By leveraging the unique features of quantum mechanics, quantum computers may be instrumental in advancing many areas, including science, energy, defense, medicine, and finance. This includes solving complex problems whose solution lies well beyond the capacity of contemporary and even future supercomputers that are based on conventional computing technologies. As a foundation for future generations of computing and information processing, quantum computing represents an exciting area for developing new ideas in computer science and computational engineering.

Interacting with the emerging capabilities of quantum computers, including noisy-intermediate scale quantum devices, for both basic and applied research will require an end-to-end software stack, not unlike the one we rely on in classical computing. This quantum software stack plays an important role in the quantum computing ecosystem, providing quantum practitioners with the essential tools to take advantage of the quantum revolution. Critical components of a quantum software stack include programming models and languages, compilers, verification and debugging tools, and hardware control capabilities. While advances are being made by the community, we are still far off from providing quantum practitioners with a cohesive software toolchain.

Over the last few years, there has been considerable effort to develop software tools that make quantum computing technology more accessible to the broader community. Many of those developed by industry, national laboratories, and academia are being made available as open-source software tools. Programming languages are being developed that make it easier for domain scientists to translate their science onto quantum computers. Similar to classical computing, compilers have been developed with the aim of minimizing resource needs with respect to the number of quantum processing units (qubits, qutrits, etc.) and the number of quantum operations that need to be performed. To aid in the development and testing of new algorithms, scalable numerical simulators and resource profilers have been developed, which form a critical component of the quantum computing software ecosystem. Only recently, approaches and tools have been developed for verifying, validating, and debugging quantum computer programs and quantum computer hardware. Finally, operating on quantum computers requires a quantum control software toolset that is likely to be hardware-technology specific. Continued research and development of a broad and open-source collection of software tools and techniques will be critical to enabling the broad adoption of quantum computing in research and industry.

The purpose of this special issue is to present recent research and development accomplishments resulting in the implementation and availability of new quantum computing software tools that will make quantum computing more practical and accessible. We hope that this special issue provides a springboard for new ideas and development across the quantum computing software stack.

The papers published in this special issue are an outflow of the First International Workshop on Quantum Computing Software [1] held in conjunction with The International Conference for High-Performance Computing, Networking, Storage, and Analysis in 2020 (SC20). Workshop contributors and the broader community of quantum software developers were invited to contribute to this special issue. The issue is split into two parts.

In this second part of the special issue, five papers are presented which make important progress toward the development of a quantum software ecosystem.

- “A backend-agnostic, quantum-classical framework for simulations of chemistry in C++” by D, Claudino, A. J. McCaskey, and D. I. Lyakh
  In this work, XACC system-level quantum computing framework is presented as a platform for prototyping, developing, and deploying quantum-classical software that specifically targets chemistry applications. This paper is a review of the fundamental design features in XACC, with special attention to its extensibility and modularity for key quantum programming workflow interfaces, and it is an overview of the interfaces most

relevant to simulations of chemistry. The provided APIs enable the community to extend, modify and devise new algorithms and applications in the realm of chemistry.

- “Automatic Qubit Characterization and Gate Optimization with QubiC” by Y. Xu, G. Huang, J. Balewski, R. K. Naik, A. Morvan, B. Mitchell, K. Nowrouzi, D. I. Santiago, and I. Siddiqi
  In this work, the authors presented a concise and automatic calibration protocol to characterize qubits and optimize gates using QubiC, which is an open-source FPGA-based control and measurement system for superconducting quantum information processors. They proposed a multi-dimensional loss-based optimization of single-qubit gates and a full XY-plane measurement method for the two-qubit CNOT gate calibration. It was demonstrated that the QubiC automatic calibration protocols are capable of delivering high-fidelity gates for transmon quantum devices operating at the Advanced Quantum Testbed at Lawrence Berkeley National Laboratory.
- “qprof: a gprof-inspired quantum profiler” by A. Suau, G. Staffelbach, and A. Todri-Sanial
  The authors introduced qprof, which is a new and extensible quantum program profiler able to generate profiling reports of various quantum circuits. The paper describes the internal structure and working of qprof and provides three practical examples of practical quantum circuits with increasing complexity. This tool will allow researchers to visualize their quantum implementation in a different way and reliably localize the bottlenecks for efficient code optimization.
- “LEAP: Scaling Numerical Optimization Based Synthesis Using an Incremental Approach” by E. Smith, M. G. Davis, J. Larson, E. Younis, C. Iancu, and W. Lavrijsen
  In this work, the authors try to address the scalability issues of circuit synthesis techniques that combine numerical optimization with search over circuit structures due to exponential search spaces and complex objective functions. The proposed LEAP algorithm improves scaling across these dimensions using iterative circuit synthesis, incremental re-optimization, dimensionality reduction, and improved numerical optimization. LEAP draws on the design of the optimal synthesis algorithm QSearch by extending it with an incremental approach to determine constant prefix solutions for a circuit. By narrowing the search space, as an example, LEAP improves scalability from four to six-qubit circuits while compiling circuits much faster than other compilers.
- “Tensor Network Quantum Virtual Machine for Simulating Quantum Circuits at Exascale” by T. Nguyen, D.I. Lyakh, E. Dumitrescu, D. Clark, J. Larkin, and A. McCaskey
  This paper presents a modernized version of the software package Tensor Network

Quantum Virtual Machine (TNQVM), which serves as a quantum circuit simulation backend in the eXtreme-scale ACCelerator (XACC) framework. The new version is based on the general purpose, scalable tensor network processing library, ExaTN, and provides multiple configurable quantum circuit simulators enabling either exact quantum circuit simulation via the full tensor network contraction or approximate quantum state representations using suitable tensor factorizations. Stochastic noise modeling from real quantum processors is incorporated into the simulations by modeling quantum channels with Kraus tensors. By combining XACC and ExaTN, it produces an end-to-end virtual quantum development environment that can scale from laptops to future exascale platforms.

Finally, we would like to thank ACM Transactions on Quantum Computing for the invitation to serve as guest editors for this special issue. We would also like to thank all the contributing authors and reviewers who found time and put effort into this special issue in spite of the pandemic.

References:

[1] https://sc20.supercomputing.org/proceedings/workshops/workshop_pages/wksp109.html