

QuantEx: A Tensor Network Simulator for Quantum Circuits



Brennan^{*1}, Momme Allalen^{†2}, David Brayford^{†3} and Luigi Iapichino^{†4}
¹john.brennan@ichec.ie, ²allalen@lrz.de, ³brayford@lrz.de, ⁴iapichino@lrz.de



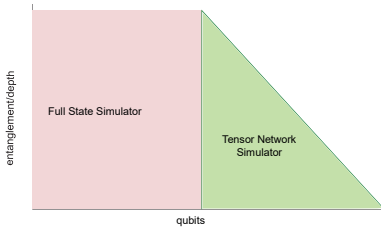
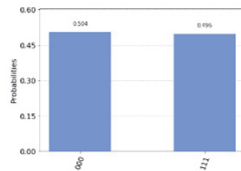
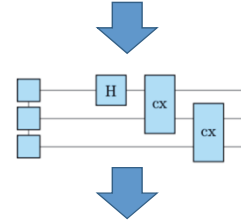
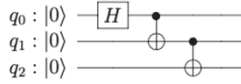
^{*}Irish Centre for High End Computing, NUI Galway
[†]Leibniz Supercomputing Centre



Tensor network methods are incredibly effective for simulating quantum circuits. This is due to their ability to efficiently represent and manipulate the wave-functions of large interacting quantum systems. In the QuantEx project, we have developed a framework for performing quantum circuit simulation using tensor network methods on large clusters using accelerators.

At a high level simulation using tensor network methods involves:

1. Converting the quantum circuit description to a tensor network description
2. Contracting the tensor network to obtain the probability amplitudes

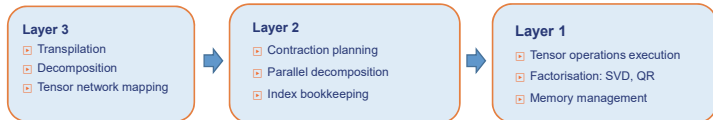


Project Objectives and features

The objectives of the project was to develop tools for quantum circuit simulation which are capable of:

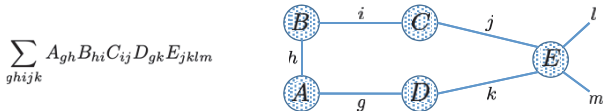
- Being easily adopted and used in user workflows
- Integrating and building upon existing tools and libraries
- Scaling to take advantage of emerging exa-scale compute clusters
- Be extended as new techniques, methods and hardware emerge

The project architecture has been designed to meet these objectives and features a design with components split into three distinct layers. This enables separation of concerns between the different components of the project. JuliaLang has been chosen as the main language for its performance and capabilities as a glue language.



Tensor Network Introduction

A Tensor Network is a tensor expression which can be expressed mathematically or graphically



- The cost of contracting a tensor network depends on the order in which tensors are contracted
- For a tensor network with N nodes, there are $N!(N-1)!/2^{N-1}$ possible contraction paths
- Finding an optimal contraction plan is believed to be NP-hard
- Developing methods for finding good contraction plans is an active area of research.

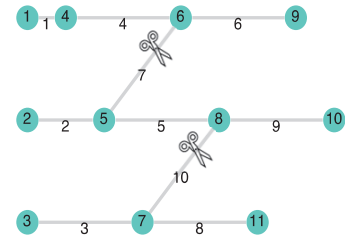
It is however possible to find “good” contraction plans in a fixed amount of time. In this project we are currently using the QuickBB algorithm to find the plan to use and are looking into more advanced approaches. The interfaces mean that these can be switched in easily.



Parallel Decomposition

As circuits grow, the memory and compute required for individual tensor contraction operations grows and can easily exceed the memory available on a single node/GPU. To overcome this limitation and to spread the computation across the available compute resources the contraction is divided into a set of smaller contractions using tensor slicing or entanglement deferral methods.

- Overcome tensor bubbling
- Load balancing
- Slicing: large network -> many smaller networks
- Choice of partitioning should be:
 - Tailored to specific hardware and architecture
 - Requires multi-optimisation approach



Profiling and Optimisation

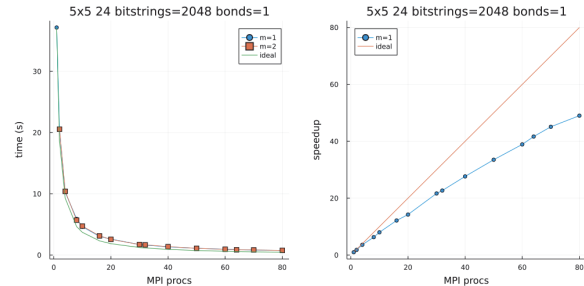
To guide optimization efforts, LIKWID was used to provide performance information on a range of platforms which to date includes:

- Intel Skylake
- AMD Rome
- A64FX

This will be extended to cover NVIDIA and AMD GPUs.

Results and Benchmarks

The Figure on the right shows preliminary MPI scaling results for simulation of a 5x5 random quantum circuit on 4 nodes with increasing numbers of processes: Benchmarking and optimization activities are ongoing.



Acknowledgements and Publications

This work was financially supported by the PRACE project funded in part by the EU's Horizon 2020 Research and Innovation programme (2014-2020) under grant agreement 823767.

Publications:

- J. Brennan et al., (2022). QXTools: A Julia framework for distributed quantum circuit simulation. *Journal of Open Source Software*, 7(70), 3711, doi:10.21105/joss.03711
- J. Brennan et al., "Tensor Network Circuit Simulation at Exascale," 2021 IEEE/ACM Second International Workshop on Quantum Computing Software (QCS), 2021, pp. 20-26, doi: 10.1109/QCS54837.2021.00006.
- D. Brayford et al., "Deploying Containerized QuanEX Quantum Simulation Software on HPC Systems," 2021 3rd International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIE-HPC), 2021, pp. 1-9, doi: 10.1109/CANOPIEHPC54579.2021.00005.

References

1. B. Villalonga et al., "Establishing the Quantum Supremacy Frontier with a 281 Pflöp/Simulation", *Quantum Sci. and Tech.*, 5 034003, (2020)
2. J. Biamonte and V. Bergholm, "Tensor networks in a nutshell," (2017), arXiv:1708.00006.
3. E. S. Friedet al., "qTorch: The quantum tensor contraction handler", *PLoS ONE*, vol. 13, no. 12, (2018), doi: 10.1371/journal.pone.0208510.
4. J. Gray and S. Kourtis, "Hyper-optimized tensor network contraction", (2020), arXiv:2002.01935.
5. E. Solomonik et al., "A massively parallel tensor contraction framework for coupled-cluster computations", *Journal of Parallel and Distributed Computing*, vol. 74, no. 12, pp. 3176–3190, Dec. (2014), doi:10.1016/j.jpdc.2014.06.002.
6. Y. Zhou and E. M. Stoudenmire and X. Waintal, "What limits the simulation of quantum computers?," (2020), arXiv:2002.07730.
7. Y. Pang, et al., "Efficient 2D Tensor Network Simulation of Quantum Systems", (2020), arXiv:2006.15234.
8. NVIDIA Corporation, "cuTENSOR: A high-performance cuda library for tensor primitives", <https://docs.nvidia.com/cuda/cutensor/>, (2019). Accessed: 2019-02-10.
9. R. N. C. Pfeifer et al., "NCON: A tensor network contractor for MATLAB", (2015), arXiv:1402.0939.
10. Likwid performance profiling <https://github.com/RRZE-HPC/likwid>
11. J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, Julia: A Fresh Approach to Numerical Computing, *SIAM Review* 2017 59:1, 65-98