Energy Efficient Quantum Computing Simulations

Jakub Adamski [1]

Supervisors: Dr. Oliver Thomson Brown [1] Dr. Raul Garcia-Patron Sanchez [2]

[1]EPCC, University of Edinburgh [2]School of Informatics, University of Edinburgh

Introduction

Classical simulations of *quantum circuits* are essential for the understanding and development of *quantum computing*. However, it is a task that **scales exponentially with the problem size**, both in *time and memory*, which makes it very energy hungry. The goal of this poster is to outline different HPC methods for performing the simulations and investigate **how to make them more efficient** in resources and energy.

Methodology

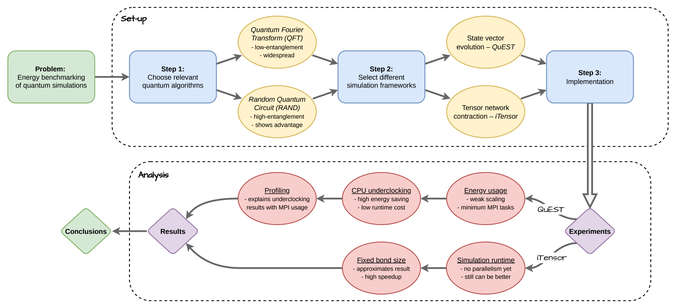

Figure 1. Flowchart outlining the process of developing of this poster.

Quantum circuits

Two circuits were selected for benchmarking, to cover a range of properties. A unique characteristic of quantum mechanics is **entanglement**, which defines how correlated different measurements are. Quantum circuits can modify the entanglement to various extents.

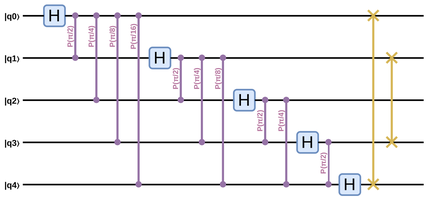

Figure 2.
Quantum Fourier Transform (QFT) – a widespread quantum algorithm. It is easy to verify, as inputting a $|0\rangle$ state should output an equal superposition of all states. This circuit can modify the entanglement only to a limited degree. It consists of $\mathcal{O}(n)$ non-diagonal *Hadamard gates* and $\mathcal{O}(n^2)$ diagonal controlled θ *phase shift gates*.

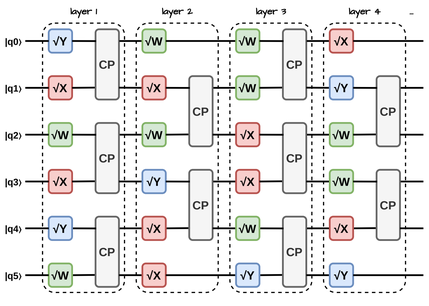

Figure 3.
Random circuit (RAND) – a simplified version of the circuit used in Google's quantum advantage claim. It is built of multiple layers that create entanglement. First, each qubit is acted on by a random gate from the set $[\sqrt{X}, \sqrt{Y}, \sqrt{W}]$; the neighbouring qubits are then entangled with controlled $\frac{\pi}{2}$ *phase shift gates*. At least n layers are necessary to significantly entangle n qubits.

The circuit has $\mathcal{O}(n^2)$ random non-diagonal single-qubit gates, and likewise, $\mathcal{O}(n^2)$ diagonal two-qubit CP gates.

The circuits above are made of quantum gates that can be described with the following matrices. In general, **diagonal gates** require less communication to simulate, since they *act locally*.

$$\boldsymbol{H} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \boldsymbol{\sqrt{X}} = \frac{1}{2}\begin{pmatrix}1+i & 1-i\\ 1-i & 1+i\end{pmatrix}, \boldsymbol{\sqrt{Y}} = \frac{1}{2}\begin{pmatrix}1+i & -1-i\\ 1+i & 1+i\end{pmatrix}, \boldsymbol{\sqrt{W}} = \frac{1}{2}\begin{pmatrix}1+i & \sqrt{2}\\ -\sqrt{2}i & 1+i\end{pmatrix},$$

$$\boldsymbol{CP(\theta)} = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&e^{i\theta}\end{pmatrix}, \quad \boldsymbol{CP} = \boldsymbol{CP}(\frac{\pi}{2}) = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&i\end{pmatrix}, \quad \boldsymbol{SWAP} = \begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}$$

Approach 1: state vector evolution – *QuEST*

State vector simulations require keeping track of the whole state, which takes up the size of $\mathcal{O}(2^n)$ for n qubits. It quickly becomes necessary to distribute the task to multiple nodes (on ARCHER2 at 34 qubits), as can be seen in table 1 below. This induces *increased communication*.

Number of qubits	32	32	33	33	34	35	36
Number of nodes	1	2	1	2	4	8	16
Peak total memory	64 GB	128 GB	128 GB	256 GB	512 GB	1 TB	2 TB
Peak per-node memory	64 GB	64 GB	128 GB	128 GB	128 GB	128 GB	128 GB
QFT min. runtime	125 s	80 s	251 s	169 s	193 s	231 s	263 s
RAND min. runtime	713 s	528 s	1524 s	724 s	1476 s	1943 s	2359 s

Table 1. Memory and time required to run quantum circuit simulations with QuEST framework.

With so much communication, nodes likely spend most time waiting on *send/receive*. Therefore, it may be possible to decrease the clock frequency without incurring much runtime cost. This can be achieved via a SLURM `--cpu-freq` input argument.

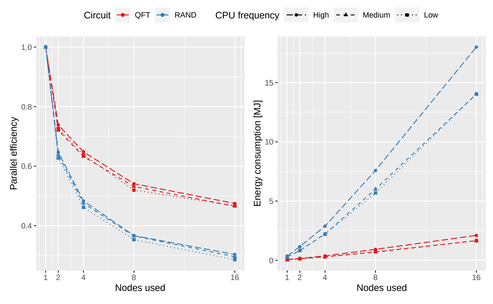

Figure 4.
Weak scaling – for a minimum number of ARCHER2 nodes required to fit the problem, starting from 1 node for 32 qubits and doubling for every new qubit. **Parallel efficiency** is poor, so this is clearly the most efficient approach. An exception is that 33 qubits can fit on 1 node (no MPI `sendrecv` buffer).

QFT performed better then RAND due to prevalent diagonal gates. As predicted, there is a significant *energy consumption gap* between high and medium CPU clock frequencies.

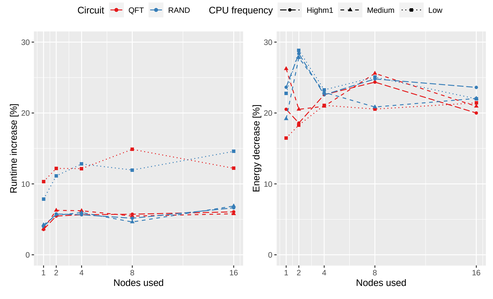

Figure 5.
Clock frequency impact – for the same scaling as on figure 4. The data is displayed with respect to high frequency. New highm1 frequency was added, which lies above medium.

It is clear that the *runtime penalty* of decreasing the frequency is *lower than the boost in energy efficiency* – medium setting being the most optimal. Therefore, it is a *viable strategy for greener simulations*, adding only a few seconds of runtime cost.

Simulation profiling with *Arm MAP*

Profiling can provide a clear picture of what is happening in the simulations.

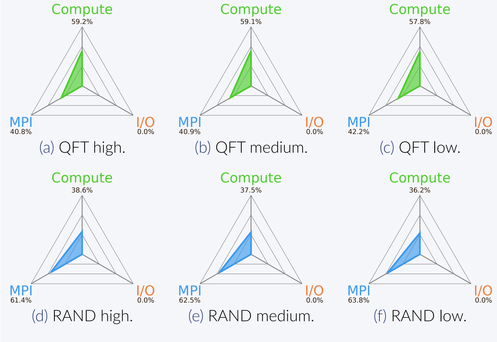

Figure 6.
Measured profiles – were collected with Arm MAP profiler on 16 nodes for 36 qubits, with varying frequencies. They show the fraction of the program spent on different *runtime stages*.

As predicted, there is a lot of *MPI calls*, especially for RAND, which is dominated by *non-local operators*. The communication fraction also seems to slightly increase when lowering the frequency, likely because MPI also requires some CPU time. The compute, on the other hand, doesn't increase, which means it is not as affected, despite lower CPU speed.

Approach 2: tensor network contraction – *iTensor*

Tensor networks allow *an efficient state vector representation* via a **Matrix Product State (MPS)**. Instead of storing 2^n amplitudes, each qubit is represented by a site, which is connected to others with *contractable bonds* of dimensions that correspond to the entanglement of the state:

- *low entanglement* states are compact and easy to evolve/contract
- *high entanglement* bond size explodes, but can be *trimmed*, which approximates the state

The platform used for the experiments is iTensor. It doesn't support any parallelism for general tensor networks, but even under those limitations it can be better than QuEST for some circuits.

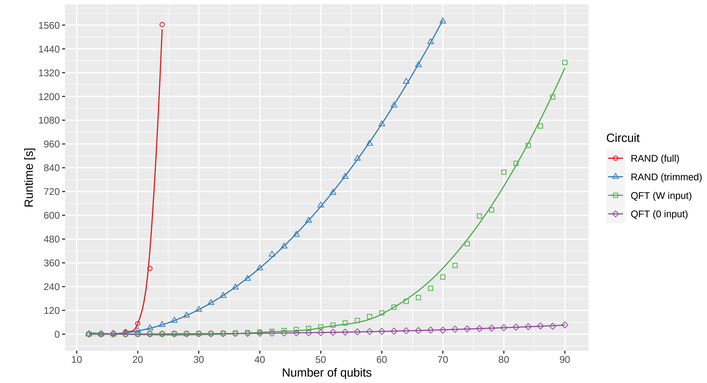

Figure 7. Runtimes of different algorithms in iTensor.

QFT is a circuit that doesn't induce much entanglement, so it can be simulated very fast. It was fed with a trivial $|00...0\rangle$ state, and a lightly entangled $|W\rangle$ state defined as:

$$|W\rangle = \frac{1}{\sqrt{n}}(|100...0\rangle + |010...0\rangle + ... + |000...1\rangle)$$

RAND is *highly entangling*, which makes it difficult to simulate the full state. Promising results were achieved by trimming the number of bond dimensions. On figure 7 *maximum bond dimension of 256* was used for the trimmed state. However, its accuracy quickly degrades for more qubits.

Conclusions

The state vector approach is *stable* regardless of the entanglement. Due to *high communication*, it is most economical when running on the **minimum number of nodes** to fit the problem, and reducing the **CPU clock frequency to medium**.

In contrast, the runtime, and thus energy consumption of tensor networks is very *dependent on the entanglement*. If it is low, **iTensor can drastically outperform QuEST** despite featuring *no parallelism*. However, for highly entangled states, the **full simulation is very slow** – but can be approximated by *trimming the bond dimensions*.

In the future, tensor network simulations should be *parallelised with OpenMP and MPI*. Then they could be directly compared against state vector methods to figure out whether and when they are advantageous, especially in case of an approximated state.

References and acknowledgements

[1] Frank Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, Oct 2019.

[2] Matthew Fishman, Steven R. White, and E. Miles Stoudenmire. The ITensor Software Library for Tensor Network Calculations. *SciPost Phys. Codebases*, page 4, 2022.

[3] Tyson Jones et al. Quest and high performance simulation of quantum computers. *Scientific Reports*, 9(1):10736, Jul 2019.

[4] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*, page 219. Cambridge University Press, USA, 10th edition, 2011.

[5] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011. January 2011 Special Issue.

This work used the ARCHER2 UK National Supercomputing Service (`https://www.archer2.ac.uk`).

This work was supported by the Engineering and Physical Sciences Research Council.