Quantum Volume

Daniel Moreno Manzano

October 24, 2018

1 Introduction

Given the different hardware implementations and technologies in Quantum Computation (superconducting, ion-trap, spin qubits, ...), it is often difficult to benchmark the usefulness or power of quantum systems. A **hardware-independent measure** is required to depict whether a device is able to run a quantum circuit or not. Here is where the Quantum Volume metric appears on the scene.

The aim of Quantum Volume is to quantify the computational power of quantum devices. Consequently we will use it as a metric to measure the runnability of the quantum algorithms and the quantum devices – ***"can this algorithm be run in a given device?"***. While the device is the basis of the Quantum Volume metric, we fix our attention on the circuit. Our purpose is to assert how the mapping procedure affects the runnability of a given circuit and to study how the Quantum Volume is related to the probability of success.

2 Literature review. A Quantum Volume definition

2.1 Hardware parameters

To define a suitable metric, we first note that a quantum computer's performance depends on:

N number of physical qubits

Quantum chip topology connectivity between qubits

Maximum number of sequencial gates with correctable errors number of gates that can be applied before errors or decoherence mask the result

Gate set Available hardware gate set

Maximum number of parallel operations Number of operations that can be run in parallel

2.2 Definitions and metrics

Model algorithm performing a depth-1 circuit, constructed by random 2-qubit unitaries chosen uniformly over SU (4) on a random pairing of the qubits.

Effective error rate ϵ_{eff} how well a device can implement arbitrary pairwise interactions between qubits. Error rate per two-qubit gate averaged over many realizations of such depth-one circuits. Therefore, it depends on the gate overhead required when all-to-all connectivity, full parallelism and a suitable gate set is not available. It encapsulates errors of both single- and two-qubit gates.

The equivalent per-gate error rate that would lead to the same overall error rate. It depends not only on the gate error rates and connectivity, but also on the sophistication of the scheduling algorithm responsible for mapping the model algorithm to the hardware.

Achievable circuit depth $d(N) \simeq \frac{1}{N\epsilon_{eff}}$ *Note that the possibility that several consecutive errors could act as the correction of that error is not take into account.*

n Number of active qubits.

Quantum Volume $\tilde{V_Q} = min(N, d(N))^2$ quantifies the space-time volume occupied by a model circuit with random two-qubit gates that can be reliably executed on a given device.

$$V_Q = \max_{n \leq N} \min \left[n, \frac{1}{n\epsilon_{eff}(n)} \right]^2$$

(u_j, v_j) qubit pairs that interact

D_0 distance between u and v for a given connectivity graph

$\sum_j D(u_j, v_j)$ Total distance

$\xi_{u,v}$ random variable $N(0, 1/n)$

D Distance Function

$$D(u, v) = (1 + |\xi_{u,v}|) D_0(u, v)^2$$

r lowest computed depth

2.3 Heuristic routing algorithm

Since the algorithm is randomized, we repeat $D(u, v) = (1 + |\xi_{u,v}|) D_0(u, v)^2$ and choose the lowest computed depth, r. Averaging r over many instances of the model algorithm gives the effective error rate as

$$\epsilon_{eff} = \epsilon(\bar{r} + 1)$$

where we assume that all SWAP gates and the needed $SU(4)$ interactions all can be done with constant error ϵ.

Note that the algorithm is random.

3 Methods

3.1 Runnability

"Can this device run a given algorithm?"

$$V_Q > V_Q^a$$

It can be understood as if a cube fits in another cube.

3.1.1 Quantum Volume of a device

Maximum Quantum Volume that a device could run

$$V_Q = \max_{n \leq N} \min \left[n, \frac{1}{n\epsilon_{eff}(n)} \right]^2$$

3.1.2 Quantum Volume of an algorithm

$$V_Q^a = \min [n, d]^2$$

3.1.3 Problem

It could be the case that one quantum circuit using more qubits than the ones available in a device has lower Quantum Volume than the one in the device. This would mean that theoretically the algorithm could be run in the quantum system, when it actually couldn't.

3.2 Depict $\epsilon_{eff}(n)$

How to depict a function of ϵ_{eff} based on experiments/simulations?

3.2.1 Bounds

With no intelligent compiler/mapping:

$$\epsilon_{eff} > \epsilon$$

3.2.2 Averaging ϵ_{eff}

With several random circuits of just 1 cycle, check their fidelity and average. That would be the $\bar{\epsilon}_{eff}$.

3.2.3 Finding the real $\epsilon_{eff}(n)$

Is not this thing kind of the error model?

3.3 Near future

~~Quantum Volume assumes that a square circuit ($d = \frac{1}{N\epsilon_{eff}} = N$) is the maximum a quantum device could get in term of errors.~~ *Maybe is not that but the initial maximum depth calculation formula that leads you to this result* Following that reasoning, with current devices of $\epsilon_{eff} > 10^{-3}$, the maximum N will be

$$N = \sqrt{\frac{1}{\epsilon_{eff}}} = 31.623$$

4 Probability of success relation with Quantum Volume

How Quantum Volume is related with Probability of success?

How to calculate ϵ_{eff} with the methods of Probability of success?