Quantum Volume

Daniel Moreno Manzano

October 27, 2018

1 Introduction

Given the different hardware implementations and technologies in Quantum Computation (super-conducting, 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 Quantum Volume definition

In this section, we define the Quantum Volume metric as well as the insights and ideas motivated by its capabilities

2.1 Literature review

Few studies have been published on the Quantum Volume topic [1, 2]. In this section, a brief introduction to the metric is offered, reviewing the important concepts and basis.

2.1.1 Hardware parameters

The Quantum Volume metric considers that a quantum computer's performance mainly depends on the next hardware specifications:

- N. The number of physical qubits

- Quantum chip topology. The connectivity between qubits

- Maximum number of sequential gates with correctable errors. The 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

2.1.2 Definitions and metrics

In order to understand the metric of Quantum Volume, some concepts need to be defined first. In this section we offer the inferred definitions from [1, 2] that are part of the rationale of Quantum Volume.

Model algorithm. In the literature, Bishop et al. use the term *model algorithm* in [1] to refer to a depth-one circuit, "constructed by random 2-qubit unitary matrix chosen uniformly over $SU(4)$ on a random pairing of the qubits". Or, what is the same, the *model algorithm* is a *circuit unit* defined by a combination of any single- or two-qubit gates. The mapping requirements of the device and the mapping quality is included as well. In the case that a two-qubit gate force any qubit to be routed, the gate additions will be included in the *model algorithm*. Therefore, we can say that the it is a device specific metric, as soon as it takes into account the constraints from it.
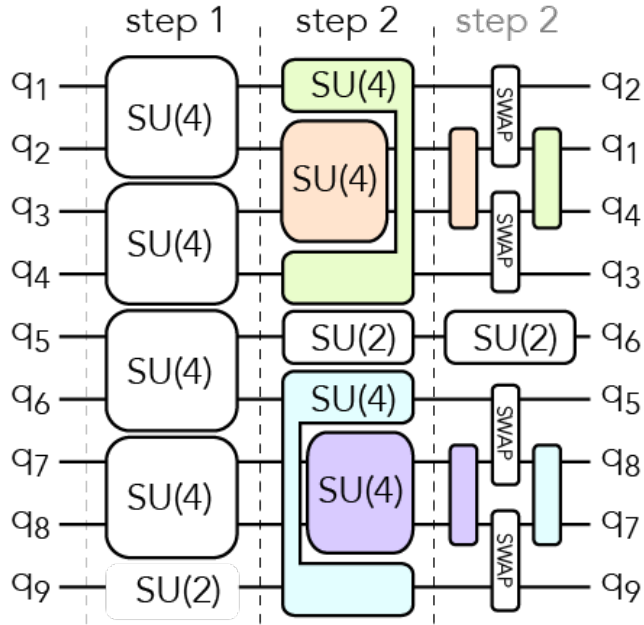

Figure 1: Model algorithm example from [2]. Each step represents a possible combination of gates considered as *model algorithm*. Notice that step 2 requires a mapping process that is shown afterwards.

Active qubits (n) Number of active qubits in a device of N qubits for a given algorithm.

Effective error rate ($\epsilon_{eff} \sim 1/(dn)$). It defines how well a device can implement arbitrary pairwise interactions between qubits. ϵ_{eff} is the error rate per *model algorithm*, an averaged error over many realizations of depth-one circuits conformed with random combinations of single-

and two-qubit gates. It encapsulates errors of both single- and two-qubit gates. And 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 $(d(N) \simeq \frac{1}{N\epsilon_{eff}})$ Maximum circuit depth for which the results, after running it on some device, are correctable and useful.

(General) 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.

2.2 Runnability

After understading the concept of Quantum Volume, we derived some insights and we had ideas motivated by the possibilities that the new metric offers. We define the **runnability** of a given quantum circuit on a device based on the separation of the concepts of *device* Quantum Volume (V_Q) and *algorithm* Quantum Volume (V_Q^a).

2.2.1 Quantum Volume of a device

Following [1, 2], we can expand the Quantum Volume general equation ($\tilde{V}_Q$) with the other definitions in the previous section and maximize for the biggest possible n in the device. Then, the maximum Quantum Volume that a device could run is defined by:

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

We define this Quantum Volume as the *device* Quantum Volume. In Fig. 2 and Fig. 3 a graph describing the Quantum Volume as a function of n and ϵ_{eff} is shown. For this example we are not considering $\epsilon_{eff}(n)$. Otherwise it would be a device specific graph and the purpose of this figure is tho show the general behaviour of V_Q. Note that the axis are in a logarithmic scale in order to show that V_Q grows exponentially as n increase and that ϵ_{eff} is abruptly detonating V_Q growth from values smaller than 10^{-3}. Therefore, we outline that the main limit for the V_Q is the ϵ_{eff}.
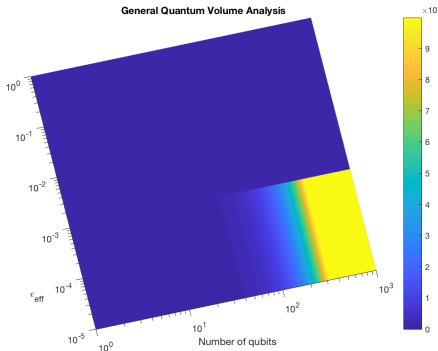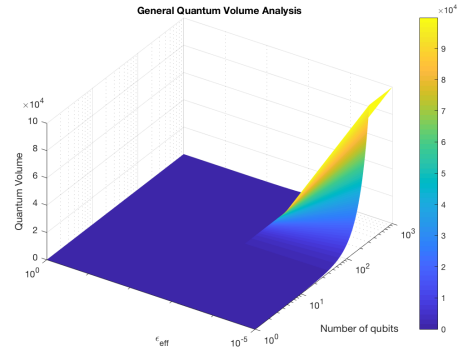

Figure 2:

Figure 3:

2.2.2 Quantum Volume of an algorithm

As with V_Q, we initially defined the algorithm's Quantum Volume from the general equation $\tilde{V}_Q$, although we will later adapt it.

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

Note that d is not $d(N)$ but the real depth of the given algorithm. At the same time, n is the number of qubits required by the algorithm itself. One can see how d and n are equally important in Fig. 6 and Fig. 7. The growth of both variables causes an equally exponential growth of V_Q^a.
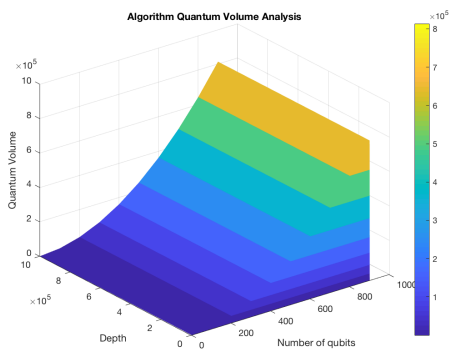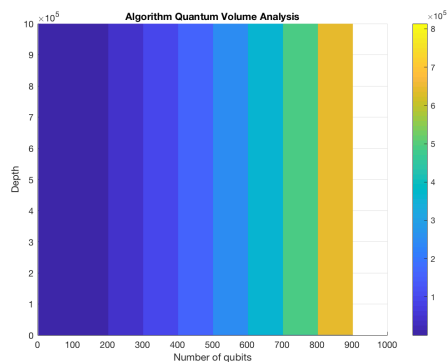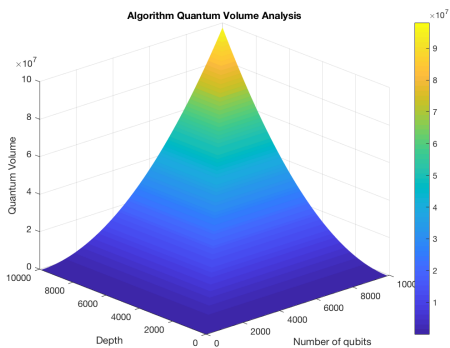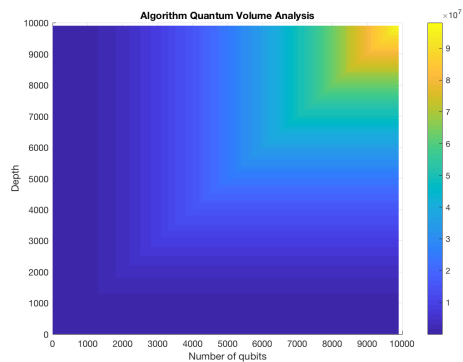

Figure 4:

Figure 5:

Figure 6:

Figure 7:

We aware that this approach has a limitation regarding the mapping of the quantum circuit. As explained before, V_Q is able to take into account the sophistication of the mapping procedure. It is inherited in the *model algorithm*. But, in this case, the V_Q^a of an algorithm before and after mapping will remain the same. After mapping an algorithm, the usual effect is an increase in the depth or the number of operations. Rare mapping methods consider the qubit addition in the technique. And, even considering it, n is not often growing too much in comparison with d. In the current NISQ era, the quantum circuits need much less qubits than depth. Therefore, most of the times, the minimum value between n and d will be n. As soon as V_Q^a is taking into account the minimum of them and the mapping procedure affects mostly to d we can conclude that this definition of V_Q^a is not considering the mapping in its results.

A simplified solution for this problem would be the V_Q^a definition as the multiplication of n and d. Unfortunately, this approach has several drawbacks as well. As Moll et al. point out [2], extreme cases of high n and low d – or the other way around – lead to inconsistencies of the multiplication metric. Considering that most of our work is not going to be in any of these extreme cases and that we can avoid those outliers, we consider the definition of the algorithm's Quantum Volume as:
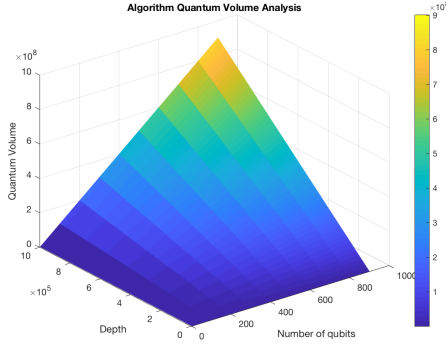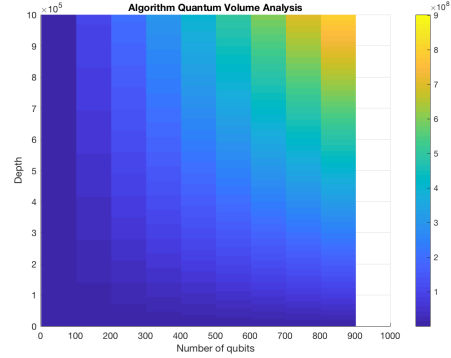
$$V_Q^a = n \times d$$

Figure 8:

Figure 9:

2.2.3   Runnability

Finally, once the Quantum Volume of device and algorithm are stated, we define runnability as the condition for which the V_Q should be bigger than V_Q^a. That is the condition that the computational power of the device should be bigger than the computational power required by the algorithm.

$$\text{Runnable if: } V_Q > V_Q^a$$

For instance, in order to understand this concept, one may imagine the process of checking, whether or not, some cube with a given volume – representing the algorithm – would fit in a box – the device –. If the "algorithm's box" volume is smaller than the volume of the device's box, the algorithm's box will fit inside.

Indeed, one acceptable criticism of this definition is that, as V_Q and V_Q^a are finally defined in the previous sections, it seems that it is not really fair to compare them. But, as soon as the general behaviour of the final and the initial V_Q^a is the same – one can see in the Fig. 7 and Fig. 9 – and that the final definition tends to be bigger than the initial one – so it is defining a more restrictive and exigent scenario – we believe that this definition of runnability mathematically correct and useful.

Therefore, we define runnability as the condition of:

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

5

3 Methodology

References

[1] Andrew Cross Jay M. Gambetta Lev S. Bishop, Sergey Bravyi and John Smolin. Quantum volume. 2017.

[2] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, and et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3):030503, Jun 2018.