Quantum Volume

Guillermo Mijares Vilariño

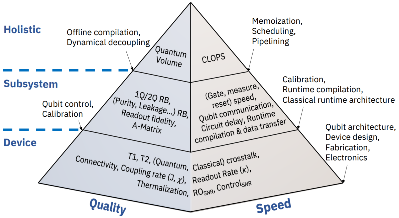
Universidad Nebrija

16 September 2022

Index

Benchmarking

There are many ways to evaluate the performance of a quantum computer depending on complexity and priority (speed or quality).



Depending on the complexity they can be classified into:

- Device: intrinsic characteristics of the quantum device.
- Subsystem: evaluate the performance of part of the device.
- Holistic: consider the device as a whole.

Holistic Benchmarks

Some of the main benchmarks of this kind are:

- Algorithmic qubits [IonQ, 2022]
- CLOPS (Circuit Layer Operations Per Second) [Wack et al. 2021]
- Quantum Volume [Cross et al. 2019]

Quantum Volume

Concept

Quantum Volume evaluates what's the biggest square random circuit that can be successfully run. If d is that size, the Quantum Volume is given by 2^d.

The Quantum Volume test can be divided into:

- Generate random square circuit.
- Simulate the circuits classically.
- Run the circuits on a real quantum device.
- Check whether the results are statistically close to those of the simulation.

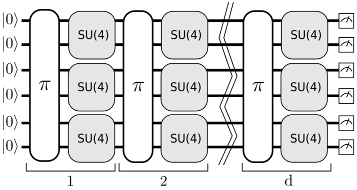
Generate random square circuits

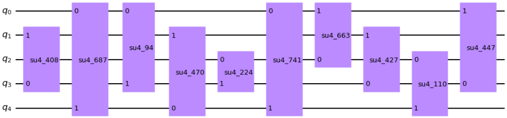
Square circuit

A square circuit of width and depth d is a circuit formed by d qubits and d layers.

Layer

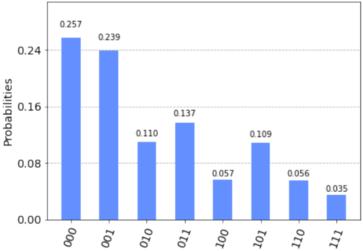
A random circuit layer consists of a random permutation π of all the qubits, followed by random 2-qubit gates over each pair of adjacent qubits.



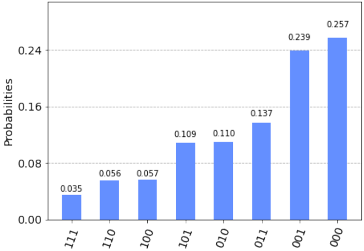


Simulate the circuits classically

We'll use a classical simulator to get the ideal output probabilities.



$\longrightarrow$



We look for the heavy outputs, that is, the outputs with higher probability than the median.

Run the circuits on a real quantum device

The objective is to get a heavy output probability (HOP) higher than $\frac{2}{3}$.

Intuition

If we run n_c circuits, n_s times each and we obtain n_h heavy outputs, it may seem natural to take

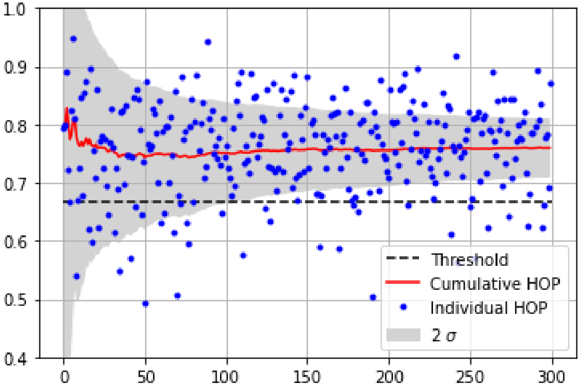
$$\text{HOP} = \frac{n_h}{n_c n_s} > \frac{2}{3}$$

Confidence

If we want to ensure a confidence of 0.975

$$\text{HOP} - 2\sigma = \frac{n_h - 2\sqrt{n_h(n_s - \frac{n_h}{n_c})}}{n_c n_s} > \frac{2}{3}.$$

Run the circuits on a real quantum device



Every trick is valid as long as an honest attempt at replicating the circuit is made, like using a circuit-to-circuit transpiler.

SABRE

SABRE (SWAP-based BidiREctional heuristic algorithm) can be used to add SWAPs or choose the best qubit layout if the device has limited connectivity.

Routing algorithm steps

- Order 2-qubit gates in layers and take the first (F)
- Choose a random qubit mapping π
- Look for gates from F that can be immediately executed, execute them and remove from F.
- Add to F any successor gates of the execute ones, if possible.
- If there are no gates that can be immediately executed, consider all SWAPs that act on the qubits of the gates of F.
- For each SWAP consider a temporal mapping π_{temp} and evaluate an objective function with it.
- Add the SWAP that minimizes the objective function.
- Repeat (except first 2 steps) until F is empty.

SABRE

Objective functions

$$H_{basic} = \sum_{gate \in F} D[\pi(gate.q_1), \pi(gate.q_2)]$$

$$H_{lookahead} = \frac{1}{|F|} \sum_{gate \in F} D[\pi(gate.q_1), \pi(gate.q_2)] + \frac{W}{|E|} \sum_{gate \in E} D[\pi(gate.q_1), \pi(gate.q_2)]$$

$$H_{decay} = \max(decay(SWAP.q_1), decay(SWAP.q_2)) H_{lookahead}$$

This routing algorithm can be used to get an initial layout by going forward and backwards through the circuit several times.

Noise adaptive layout

The noise adaptive layout algorithm [Murali et al. 2019] uses the device properties to para get a qubit layout.

Steps

- The virtual CNOTS are ordered by descending weight $\rightarrow$ ascending control qubit $\rightarrow$ ascending target qubit.
- If there is a gate with only one mapped qubit, choose it. Otherwise, pick the first from the list.
- If no qubit from the gate is mapped, pick the physical CNOT with the highest total reliability.
- If one of the qubits is mapped, pick the physical qubit that best fits the already mapped neighbors of the virtual qubit.

Studied Devices

Dispositivo	qubits	QV	obtained QV
ibm_cairo	27	64	16
ibmq_mumbai	27	128	8
ibmq_montreal	27	128	16
ibmq_jakarta	7	16	16
ibm_perth	7	32	8
ibm_lagos	7	32	8
ibmq_lima	5	8	8

Bigger devices vs smaller

Quantum Volume 8

Device	qubits	n_c	n_s	HOP	2σ
`ibmq_lima`	5	200	4000	**0.782**	0.058
`ibmq_jakarta`	7	200	4000	**0.781**	0.058
`ibm_perth`	7	300	5000	**0.768**	0.048
`ibm_lagos`	7	300	5000	**0.744**	0.050
`ibmq_cairo`	27	300	5000	**0.672**	0.054
`ibmq_mumbai`	27	300	5000	**0.669**	0.054
`ibmq_montreal`	27	300	5000	**0.720**	0.052

Smaller devices give better heavy output probability (HOP).

Bigger devices vs smaller

Quantum Volume 16

Device	qubits	n_c	n_s	HOP	2σ
`ibmq_jakarta`	7	1500	5000	**0.587**	0.025
`ibm_perth`	7	300	5000	**0.561**	0.057
`ibm_lagos`	7	300	5000	**0.595**	0.057
`ibm_cairo`	27	300	5000	**0.57**	0.057
`ibmq_mumbai`	27	300	5000	**0.548**	0.057
`ibmq_montreal`	27	300	5000	**0.621**	0.056

There are no clear differences between bigger and smaller devices.

Qubit Layout

Smaller devices

device	n_c	n_s	HOP	2σ	layout
ibmq_jakarta	2400	2000	**0.676**	0.019	
ibmq_jakarta	900	2000	**0.650**	0.032	**noise adaptive**
ibmq_jakarta	3600	2000	**0.687**	0.015	**SABRE**
ibm_perth	300	5000	**0.684**	0.054	
ibm_perth	300	5000	**0.632**	0.056	**noise adaptive**
ibm_perth	300	5000	**0.679**	0.054	**SABRE**
ibm_lagos	300	5000	**0.654**	0.054	
ibm_lagos	300	5000	**0.632**	0.056	**noise adaptive**
ibm_lagos	300	5000	**0.646**	0.055	**SABRE**

SABRE gives better results

Qubit Layout

Bigger devices

Device	n_c	n_s	HOP	2σ	layout
ibmq_mumbai	300	5000	**0.609**	0.056	
ibmq_mumbai	300	5000	**0.683**	0.054	**noise adaptive**
ibmq_mumbai	300	5000	**0.623**	0.056	**SABRE**
ibmq_montreal	300	5000	**0.654**	0.055	
ibmq_montreal	300	5000	**0.714**	0.052	**noise adaptive**
ibmq_montreal	300	5000	**0.645**	0.055	**SABRE**

Noise adaptive gives better results.

Optimization level

Quantum Volume 8

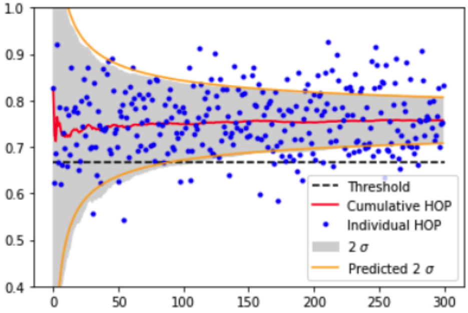
Device	qubits	HOP increase
ibmq_montreal	27	0.048
ibmq_mumbai	27	0.097
ibm_cairo	27	0.044
ibm_perth	7	0.016
ibm_lagos	7	0.014

Quantum Volume 16

Device	qubits	HOP increase
ibmq_montreal	27	0.033
ibm_lagos	7	0.059
ibmq_mumbai	27	0.061
ibmq_jakarta	7	0.089
ibm_cairo	27	0.082
ibm_perth	7	0.123

Number of circuits

Heavy output probability (HOP) becomes approximately constant



$$2\sigma = \frac{2\sqrt{n_h\left(n_s - \frac{n_h}{n_s}\right)}}{n_c n_s} = 2\sqrt{\frac{\frac{n_h}{n_s n_c}\left(1 - \frac{n_h}{n_s n_c}\right)}{n_c}} = 2\sqrt{\frac{\mathrm{HOP}(1 - \mathrm{HOP})}{n_c}}$$

Conclusiones

- Obtained Quantum Volume is far from the one reported by IBM for bigger devices $\rightarrow$ need for more techniques.
- For smaller devices the result is closer to IBM's $\rightarrow$ Fewer variables or less studied?
- Without optimization, smaller devices perform better at Quantum Volume 8 test while there are no clear difference at Quantum Volume 16 test.
- SABRE works better than noise adaptive for smaller devices. For bigger ones noise adaptive is better.
- The effect of increasing optimization level is more noticeable for bigger devices for Quantum Volume 8 test. For Quantum Volume 16 the result varies from device to device.
- Number of circuits doesn't have a big impact on HOP but 2σ is inversely proportional to its square root.

Bibliography

IonQ (2022)
Algorithmic Qubits: A Better Single-Number Metric
URL: https://ionq.com/posts/february-23-2022-algorithmic-qubits

Wack, A., Paik, H., Javadi-Abhari, A., Jurcevic, P., Faro, I., Gambetta, J. M. & Johnson, B. R. (2021)
'Quality, speed, and scale: three key attributes to measure the performance of near-term quantum computers'
URL: https://arxiv.org/abs/2110.14108 83

Cross, A. W., Bishop, L. S., Sheldon, S., Nation, P. D. & Gambetta, J. M. (2019)
'Validating quantum computers using randomized model circuits'
Physical Review A 100(3). URL: https://doi.org/10.1103%2Fphysreva.100.032328

Li, G., Ding, Y. & Xie, Y. (2018)
'Tackling the qubit mapping problem for nisq-era quantum devices'
URL: https://arxiv.org/abs/1809.02573

Murali, P., Baker, J. M., Abhari, A. J., Chong, F. T. & Martonosi, M. (2019)
'Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers'
URL: https://arxiv.org/abs/1901.11054