Simon's algorithm in the NISQ cloud

Reece Robertson, Emery Doucet, Ernest Spicer, and Sebastian Deffner

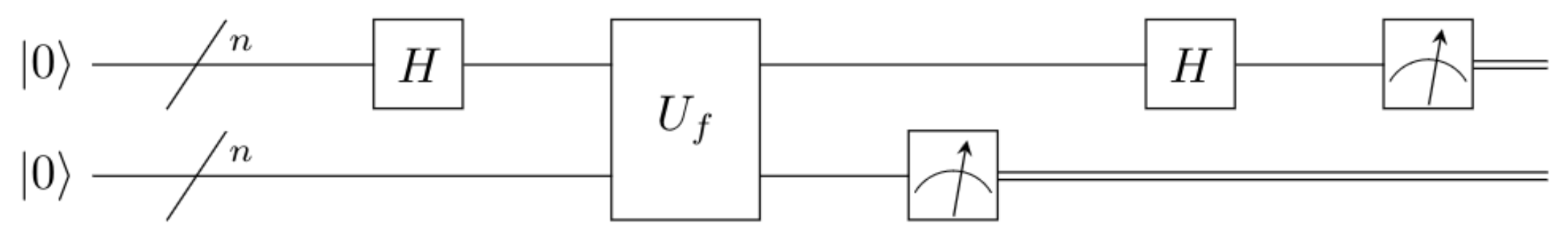
Department of Physics, University of Maryland Baltimore County, Baltimore, Maryland 21250, USA

Abstract

Simon's algorithm demonstrates genuine quantum advantage, however, it assumes access to noise-free qubits. We use Simon's algorithm to benchmark the error rates of devices currently available in the "quantum cloud." Our main result is a comparison between the platforms made available by IBM and IonQ. This highlights the importance of understanding device architectures when transpiling quantum algorithms onto hardware. For instance, we show that two-qubit operations between distant superconducting qubits should be avoided.

Background

Simon's problem takes as input an oracle function U_f which operates on bitstrings of size n. The function is either one-to-one, or two-to-one with a fixed secret string s that relates each pair of inputs. The objective is to classify the provided function into one of the two classes. Simon invented a quantum algorithm which completes this task in polynomial time, while a classical computer must run a probabilistic calculation with an exponential expected runtime. [1]

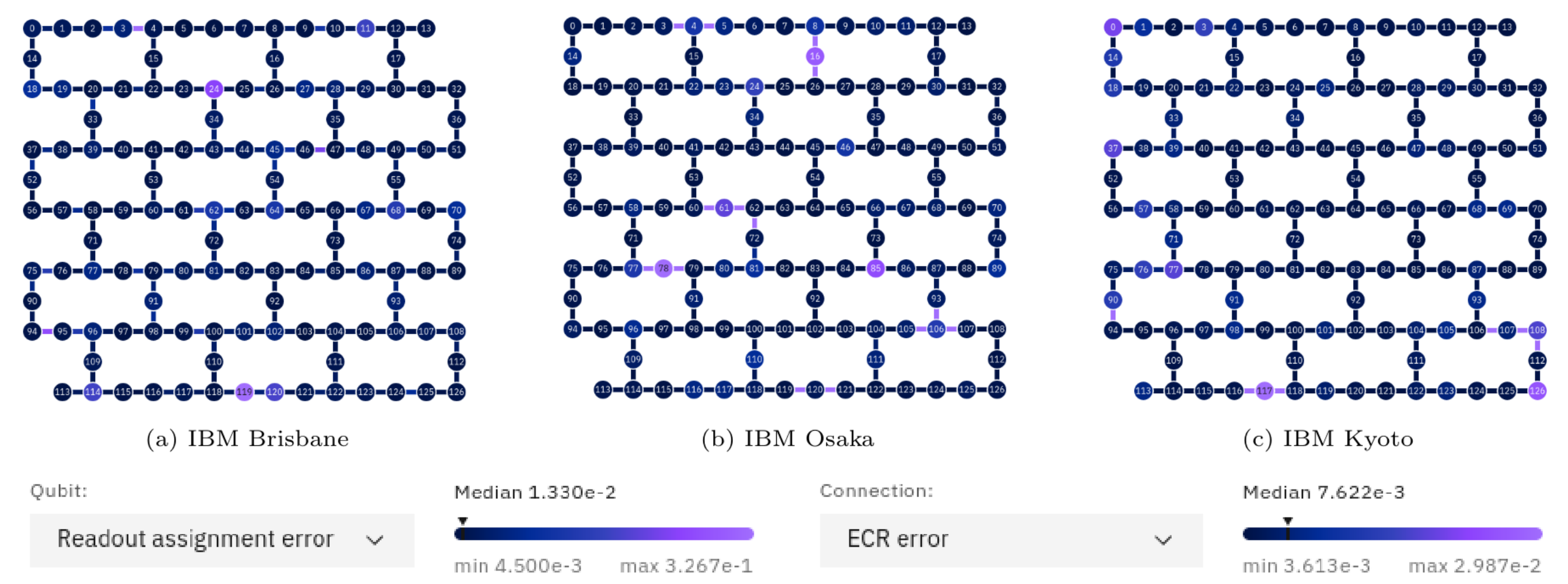


Above is the quantum circuit diagram for Simon's algorithm. Note the use of two quantum registers of size n, both initialized to the state $|0\rangle$, as well as the oracle U_f. The algorithm uses two Hadamard transformations and U_f to create a superposition over all the size n bitstrings that are orthogonal to the secret string s. An iteration of the algorithm is successful if the final measurement result is indeed orthogonal to s (this is always true in the noise free case). The entire algorithm is successful if a complete set of $n-1$ linearly independent bitstrings are measured and the resulting system is solved classically in polynomial-time.

The Quantum Cloud

Several large corporations as well as smaller start-up companies have made their NISQ devices available for cloud access. This offers exciting prospects for fundamental as well as applied research, as now with comparatively little effort "quantum experiments" can be performed. However, it has also already been demonstrated that the reported *quantum volume*[a] often overestimates what is experimentally accessible. This raises the somewhat natural question, how do algorithms designed for the very purpose of exhibiting quantum advantage perform on available NISQ devices? For our study, we had access to IBM's superconducting platform and IonQ's trapped ion devices.

Superconducting qubits – IBM: Quantum processors are made up superconducting transmon qubits are located in dilution refrigerators at the IBM Research headquarters at the Thomas J. Watson Research Center. For our work, we used the 127-qubit Brisbane, Osaka, and Kyoto superconducting quantum processors, for which noisy simulators are also provided. The device topologies for each quantum processing unit are depicted below (note that all are instances of the IBM Eagle chip design).

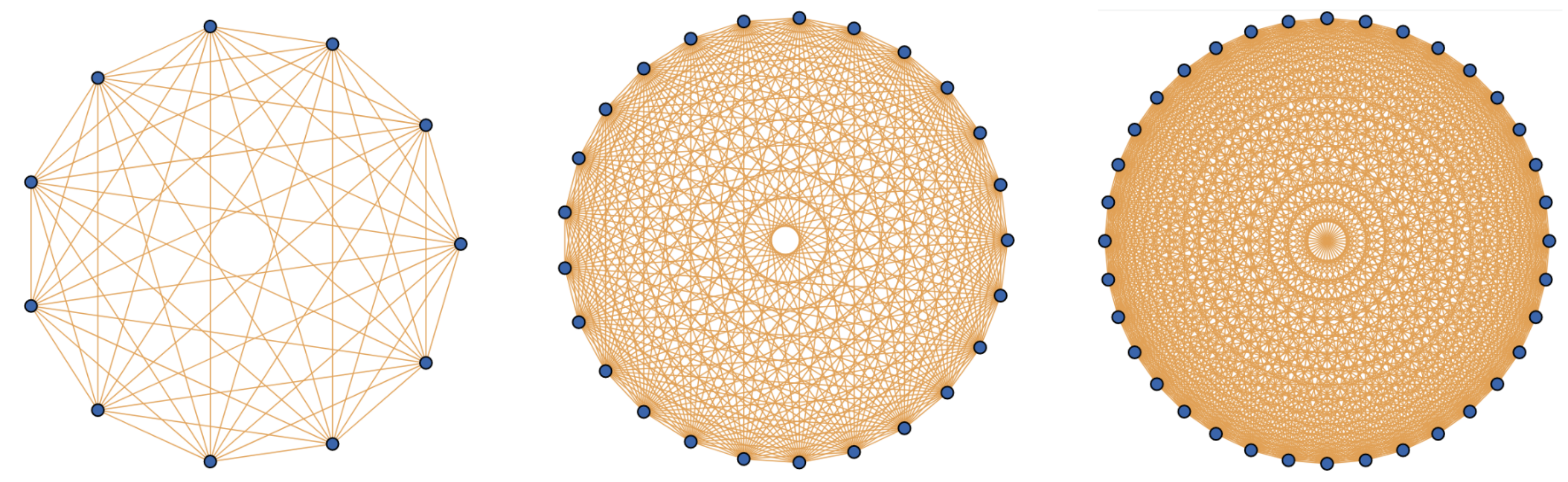


(a) IBM Brisbane (b) IBM Osaka (c) IBM Kyoto

Qubit: Readout assignment error — Median 1.330e-2, min 4.500e-3, max 3.267e-1

Connection: ECR error — Median 7.622e-3, min 3.613e-3, max 2.987e-2

Ion traps – IonQ: A fundamentally different type of system are the trapped ion devices of IonQ, which are physically located here in College Park, Maryland. Typically, trapped ion NISQ devices exhibit several advantages, such as accuracy, predictability, and coherence time. However, such systems have the disadvantage that the number of qubits is much smaller. For instance, the largest IonQ device (Forte) has only 36 qubits. Some of the limitations arising from the small qubit number is made up by the fact that trapped ion computers possess full connectivity.



(a) IonQ Harmony (b) IonQ Aria 1 (c) IonQ Forte 1

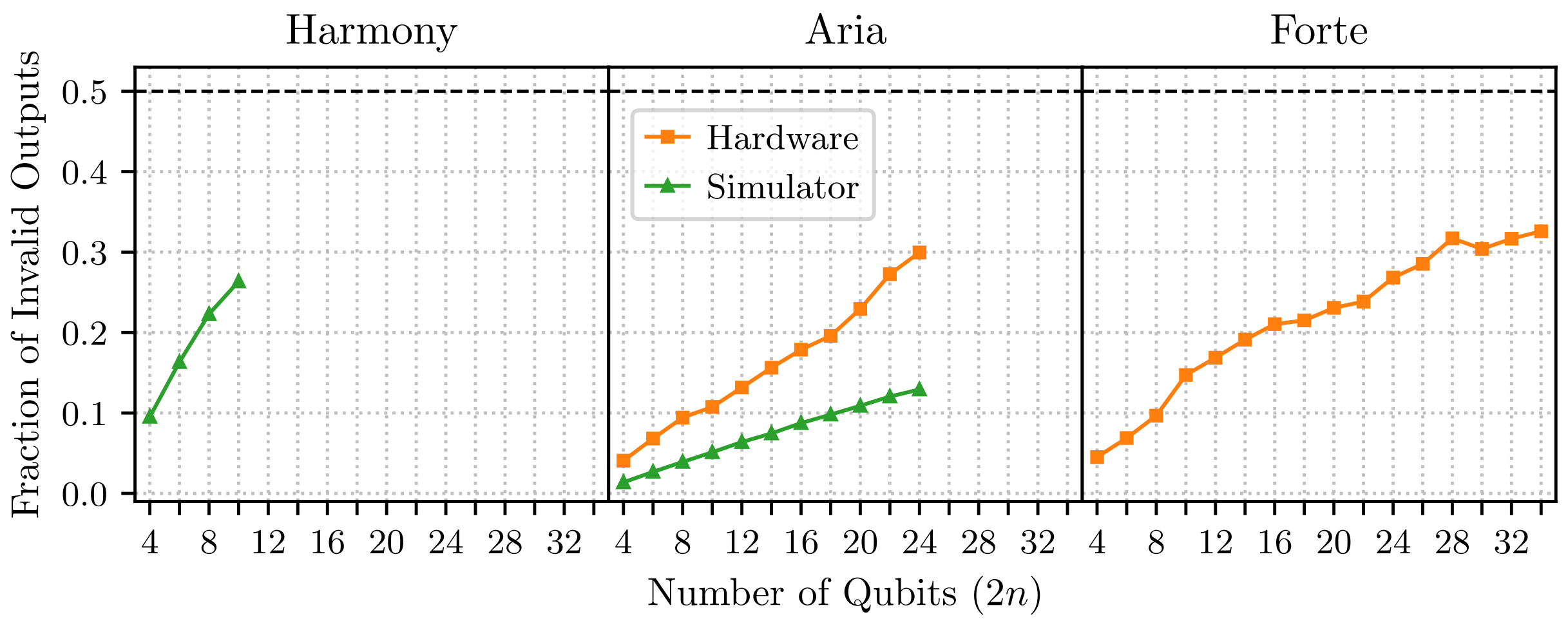
[a] The *quantum volume* is defined using the number of qubits and the number of operations that can be effectively implemented on the device.

Methods & Results

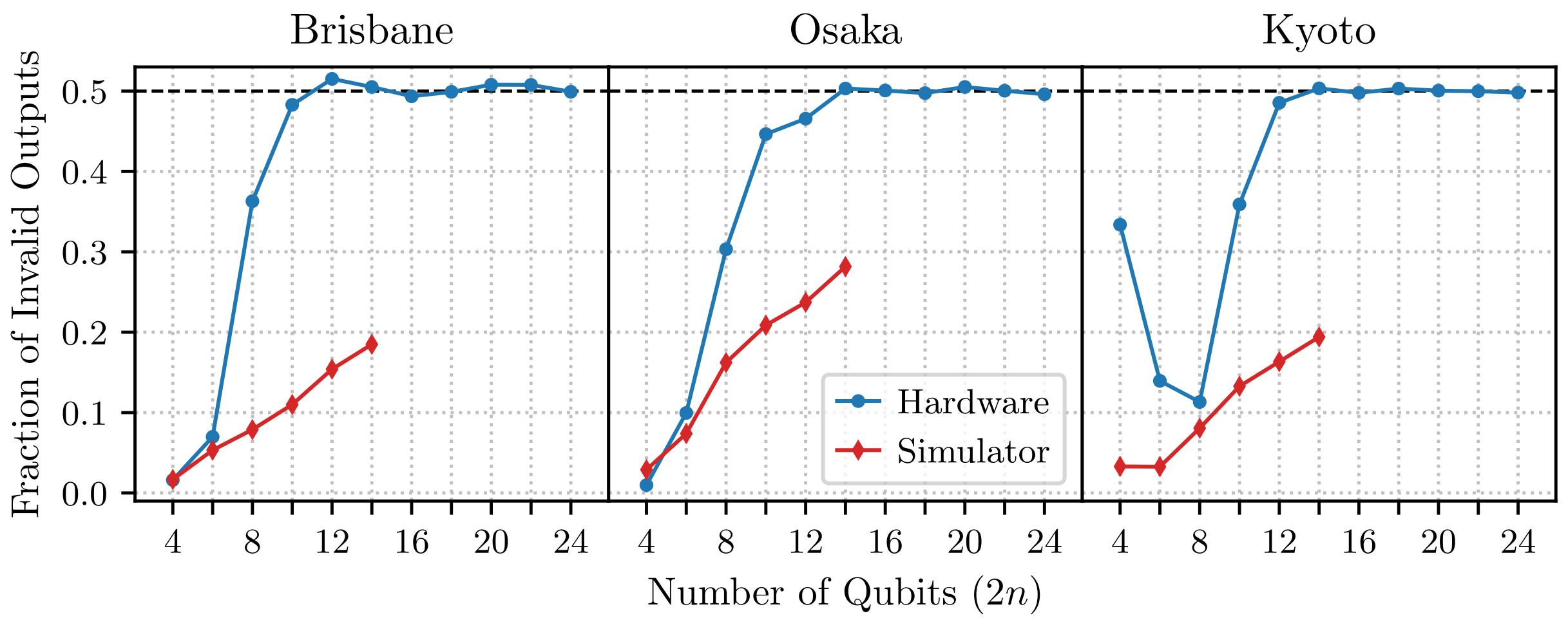
We ran Simon's algorithm on the IBM superconducting and IonQ trapped ion quantum devices listed in "The Quantum Cloud" section. We also ran Simon's algorithm on each devices' corresponding simulator. Our process was as follows:

1. We initialized a variable n to iterate over the values 2—12.
2. For each n, we defined a two-to-one function using the string of n 1s as the secret string s. This represents the most complex oracle in terms of the number of two-qubit gates required to implement it.
3. We prepared an implementation of Simon's algorithm to classify this function.
4. We repeated the implementation 8192 times.
5. We counted the number of times the implementation produced a final result which was not orthogonal to s, indicating a failure of the algorithm (the "algorithmic error").
6. We averaged the algorithmic error of each device over many repetitions of steps 1-5, and plotted the results.

Our findings are presented in the figures below. The first plot shows the algorithmic error on each IonQ device and simulator scales linearly. This indicates two things: first, that the IonQ devices scale consistently throughout the "intermediate scale," and second, IonQ simulators agree with the general behavior of the real devices (up to a constant factor). However, with an observed 30% algorithmic error for $n = 12$, the IonQ devices are not performing particularly well—we expect this amount of error to nullify quantum advantage.



The second plot shows that IBM Brisbane experiences a nonlinear jump in algorithmic error at $n = 4$, and by the moderately sized $n > 8$, the algorithm fails entirely for this oracle. More curious is the output of the noisy simulators. The IBM AER local simulators all predict a roughly linear scaling in algorithmic error as problem size increases, similar to what was observed on IonQ but in sharp contrast to the IBM results.
A plausible hypothesis is that this increase in error on superconducting hardware coincides with the addition of a number of swap gates into the algorithm due to the limited chip layout. Interestingly, however, the simulated algorithm also has swap gates of the same pattern, but the jump in error is not observed. Thus, the dramatic jump in algorithmic error observed on the hardware may be due to correlated errors that accompany the addition of the swap operations which is not captured in the noise model of the simulators.



In short, we find that the algorithmic error of Simon's algorithm on NISQ devices scales at least linearly, and all devices show irrecoverable error for intermediate-scale problems ($n = 12$).

Physical Parameters of the NISQ Devices

Parameter	Brisbane	Osaka	Kyoto	Forte	Aria 1	Harmony
Manufacturer	IBM	IBM	IBM	IonQ	IonQ	IonQ
T1 Time	213.12 μs	297.17 μs	215.43 μs	100 s	100 s	10000 s
T2 Time	145.97 μs	127.23 μs	109.44 μs	1 s	1 s	0.2 s
2-Qubit Gate Speed	660 ns	660 ns	660 ns	970 μs	600 μs	200 μs
1-Qubit Gate Error (%)	0.03	0.03	0.03	0.09	0.06	0.67
2-Qubit Gate Error (%)	0.74	0.93	0.92	0.74	8.57	3.07
Average Readout Error (%)	1.32	2.18	1.48	0.5	0.52	0.42
Topology	Eagle r3	Eagle r3	Eagle r3	all-to-all	all-to-all	all-to-all

Performance metrics for IBM and IonQ devices. Note that for one and two qubit gate errors and times, IBM reports median values while IonQ reports average values. The native gate set supported by the IBM devices includes: ECR, ID, RZ, SX, and X gates; while the native gate set for the IonQ devices consists of: MS, GPI, and GPI2 gates (IonQ Forte also implements ZZ gates natively).

References

[1] D. R. Simon, On the power of quantum computation, SIAM Journal on Computing 26 (5) (1997) 1474–1483. arXiv: https://doi.org/10.1137/S0097539796298637, doi:10.1137/S0097539796298637.