LLNL-TR-819320

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

A technical report on the progress with Rigetti platform while working on Grover's search project

V. I. Geyko

February 9, 2021

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Using Grover's search algorithm to characterize the Rigetti Quantum Computing Platform

V. I. Geyko
Lawrence Livermore National Laboratory, Livermore, California, 94550, USA[*]

(Dated: February 9, 2021)

Performance of the Rigetti quantum computing platform was tested during the period of time from 12/16/2019 to 05/18/2020. The Grover's search (GS) algorithm was used as a testing tool, in particular, 3- and 4-level versions of the algorithm. As a result, a number of hardware issues were revealed, so the algorithm was split to smaller blocks and individual gates, and all of them were tested separately. The fidelity decay was found to be due to both decoherent processes and coherent errors of the native hardware gates. These errors were estimated from RX gate benchmarks and were in a form of extra rotation of the quantum state. As a consequence of that, performance of the algorithm was shown to be strongly dependent on the native gate decomposition of the program. Suggestions and possible improvements of the future Rigetti runs are made based on the obtained observations.

I. INTRODUCTION

Rigetti Computing[1] is a private company that provides quantum computing services for customers and collaborators. Their hardware is based on superconducting qubit technology, with a standard gate set and control options. There are two fundamental gates for a single qubit rotation (RX, RZ) and one two-qubit gate that performs multi-qubit entanglement (CZ). While Rigetti offers a variety of different lattices, which contain up to 28 qubits, the main focus of this research was on two-qubit machines, in particular Aspen-4-2Q and Aspen-7-2Q. Multi-qubit lattices have not yet been tested due to the increase in complexity of the required benchmarks and unexpected behavior of some 2-qubit systems. The initial motivation of the Grover's search (GS) project was to test the performance of the 3 and 4 level quantum systems. However, it was quickly revealed that a careful analysis of different components of the algorithm is required for better understanding of appeared issues. As a result, such aspects as abnormal hardware behavior, individual native gate benchmarking, and gate decomposition are addressed in the present report.

The paper is organized as follows. A brief overview on the native gates used by Rigetti is given in Sec. (II). A description of the Grover's search algorithm and its implementation is provided in Sec. (III). A list of revealed hardware issues is given in Sec. (IV). Native gate benchmarks are performed in Sec. (V). In Sec. (VI), the results are summarized and the ideas on future work and improvements are submitted.

II. NATIVE GATE OVERVIEW

Any program that is run on the Rigetti platform is always decomposed into a sequence of native gates [2, 3]. There are parametric single-qubit gates RX and RZ, and a non-parametric CZ gate for two given qubits. Since a 2-qubit system is considered in this report only, the CZ gate is applied to the second qubit and controlled by the first one, and it is denoted as

$$\mathrm{CZ}\ 1\ 0 = \mathrm{CZ}(1,0) \tag{1}$$

Here, a standard numeration is used in where 0 denotes the first qubit and 1 denotes the second qubit. The matrix for the CZ gate reads as

$$\mathrm{CZ}\ 1\ 0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{2}$$

The CZ gate is a *hardware gate*, thus, it is implemented on the hardware level and actual control signals are sent to the system in order to manipulate the qubits. It is also known as the weakest link in a quantum circuit, as the fidelity of this gate is worse than to single qubit gates.

The RZ is a parametric single qubit gate, which is denoted as

$$\mathrm{RZ}(\phi)\ q, \tag{3}$$

where q is the qubit the gate is applied to, and ϕ is the rotation phase in the range $[-4\pi; +4\pi]$. The gate performs a rotation along $\hat{z}$-direction, and its matrix for a single qubit reads as

$$\mathrm{RZ}(2\phi) = \begin{bmatrix} \exp(-i\phi) & 0 \\ 0 & \exp(+i\phi) \end{bmatrix} \tag{4}$$

For two qubits, the matrices are obtained via tensor product with the identity matrix, namely

$$\mathrm{RZ}(2\phi)\ 0 = \mathrm{RZ}(2\phi, 0) \otimes \mathrm{I}(1) \tag{5}$$
$$= \begin{bmatrix} \exp(-i\phi) & 0 & 0 & 0 \\ 0 & \exp(-i\phi) & 0 & 0 \\ 0 & 0 & \exp(+i\phi) & 0 \\ 0 & 0 & 0 & \exp(+i\phi) \end{bmatrix}$$

[*] Electronic address: geyko1@llnl.gov

for the RZ gate applied to 0-qubit, and

$$\mathrm{RZ}(2\phi)\ 1 = \mathrm{I}(0) \otimes \mathrm{RZ}(2\phi, 1) \tag{6}$$
$$= \begin{bmatrix} \exp(-i\phi) & 0 & 0 & 0 \\ 0 & \exp(+i\phi) & 0 & 0 \\ 0 & 0 & \exp(-i\phi) & 0 \\ 0 & 0 & 0 & \exp(+i\phi) \end{bmatrix}$$

for the RZ gate applied to 1-qubit. Notice that the matrices in Eqs. (5,6) are certainly different.

Since the RZ gate does not change the population of the qubit states and only rotates the relative phase between the ground and excited states, it is implemented on the *software* level, hence there is no fidelity penalty of applying this gate as many times as required, and the effect is taken into account when the other gates are applied later in the program.

The RX gate is another single qubit gate and, while it can be called with a parameter ϕ, it is actually implemented for $\phi = k\pi/2$, where k is an integer. The single qubit gate matrix representation reads as

$$\mathrm{RX}(2\phi) = \begin{bmatrix} \cos(\phi) & -i\sin(\phi) \\ -i\sin(\phi) & \cos(\phi) \end{bmatrix} \tag{7}$$

and two-qubit gates are obtained via tensor product with I(1) for the RX gate applied to 0-qubit

$$\mathrm{RX}(2\phi)\ 0 = \mathrm{RX}(2\phi, 0) \otimes \mathrm{I}(1) \tag{8}$$
$$= \begin{bmatrix} \cos(\phi) & 0 & -i\sin(\phi) & 0 \\ 0 & \cos(\phi) & 0 & -i\sin(\phi) \\ -i\sin(\phi) & 0 & \cos(\phi) & 0 \\ 0 & -i\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}$$

and with I(0) for the other case

$$\mathrm{RX}(2\phi)\ 1 = \mathrm{I}(0) \otimes \mathrm{RX}(2\phi, 1) \tag{9}$$
$$= \begin{bmatrix} \cos(\phi) & -i\sin(\phi) & 0 & 0 \\ -i\sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & \cos(\phi) & -i\sin(\phi) \\ 0 & 0 & -i\sin(\phi) & \cos(\phi) \end{bmatrix}$$

The RX gate is a *hardware* gate and unlike the RZ gate that just rotates the relative phase between the states, this one changes the population of the qubit states. As a result, this gate has finite accuracy and its fidelity is an interesting subject for future research.

III. GROVER'S SEARCH ALGORITHM

A. Theoretical background

Initially, the motivation for using the Rigetti platform was to compare its performance to LLNL QuDIT and possibly with IBM-Q. The main tool was using a 3-level Grover's search algorithm, as LLNL QuDIT was limited to 3 stable levels only at the time the project was proposed. Besides, a 3-level problem is more complicated and challenging than a standard 4-level (or 2-qubit) classical GS algorithm. However, the performance of the 3-level GS algorithm on the Rigetti platform was considerably below expected, so the research drifted to somewhat more simple, yet more fundamental tests, such as benchmarking the individual program constituents and subsequently to gate set tomography in order to reveal the reasons behind the observed discrepancies. A brief recap of GS algorithm is required here, as it provides the main blocks that are to be tested.

Consider an N-level quantum system, where $N = 3$ or $N = 4$ depending on the needs and benchmarks performed. A quantum state is denoted as $|\psi\rangle$, and the basis decomposition reads as

$$|\psi\rangle = \sum_{k=0}^{N-1} \alpha_k |k\rangle, \tag{10}$$

where $|k\rangle$ is the basis vector, such that the only k-th state is occupied, with the standard normalization condition applied

$$\sum_{k=0}^{N-1} |\alpha_k^2| = 1. \tag{11}$$

A 3-level system can be realized on a 2-qubit platform by simply using three states only ($|00\rangle$, $|01\rangle$, $|10\rangle$). In terms of matrix representation, a 3 by 3 block matrix is embedded in the 4 by 4 matrix of a 2-qubit system. Ideally, the unused states should remain unpopulated (which is not the case and the goal of the paper is to report the observed discrepancy) and therefore are kept intact by the gates, so the state vector takes the same form as in Eq. (10), or in other words, $\alpha_k = 0$, for $k \geq N$. In principle, the choice of the unused states is arbitrary, as any two basis states can be swapped while the corresponding matrix rows and columns are swapped as well. In this paper, however, the described structure is fixed, for several reasons. First, for the relative simplicity, when all unused states are gathered together instead of being spread all over the matrix. Second, the excited states of qubits are known to be less robust than ground states, so by choosing lowest states for the operation the fidelity of the simulations is improved.

Grover's search quantum circuit consists of the 3 main elements: Superposition gate S, Oracle U_ω, and Grover's diffusion operator U_s. The superposition gate is needed to spread the state vector "evenly" among all the states the search algorithm is operating on, assuming that the initial state is the ground state $|0\rangle$. The superposition state $|s\rangle$ is given by

$$|s\rangle = \frac{1}{N} \sum_{k=0}^{N-1} |k\rangle. \tag{12}$$

For the 4-level GS algorithm, the superposition gate is implemented as a 2-qubit Hadamard gate

$$S_4 = H_2 = H_0 \otimes H_1, \tag{13}$$

with the matrix

$$S_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{14}$$

For the 3-level system, the S-gate the discrete Fourier transformation[4] is used with the matrix

$$S_3 = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & e^{i\delta} & e^{-i\delta} & 0 \\ 1 & e^{-i\delta} & e^{i\delta} & 0 \\ 0 & 0 & 0 & \sqrt{3} \end{bmatrix} \tag{15}$$

where $\delta = 2\pi/3$, and i is the imaginary unit. The matrix in Eq. (15) is unitary, as any two vectors formed by matrix columns are orthogonal and of unit norm.

The Oracle gate, by the design of the algorithm, marks the state that is being searched for; hence it is assumed to be unknown in advance. The structure of the gate, however, is well-defined, and in this particular realization it is a phase flip of the ω state, namely $U_\omega|\psi\rangle = |\psi\rangle$ for $\langle\psi|\omega\rangle = 0$, and $U_\omega|\psi\rangle = -|\psi\rangle$ for $\langle\psi|\omega\rangle = 1$. In other words, the operator reads as

$$U_\omega = I - 2|\omega\rangle\langle\omega|, \tag{16}$$

and the matrix representation is trivial: it is a unit matrix with -1 instead of 1 at the position (ω, ω). Once the structure is fixed, it is only the state number ω that is unknown and has to be guessed by the algorithm.

The Grover's diffusion operator reflects the state vector $|\psi\rangle$ through the superposition vector $|s\rangle$

$$U_s = 2|s\rangle\langle s| - I. \tag{17}$$

The matrix representation is rather dense, it takes the following form for the 3-level system

$$U_{s3} = \frac{1}{3}\begin{bmatrix} -1 & 2 & 2 & 0 \\ 2 & -1 & 2 & 0 \\ 2 & 2 & -1 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix} \tag{18}$$

and for the 4-level system

$$U_{s4} = \frac{1}{2}\begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \tag{19}$$

The Grover's search circuit is demonstrated in Fig. (1). The initial state $|0\rangle$ is changed to $|s\rangle$ by applying the S gate, then the block of U_ω and U_s is applied several times. At the end of the algorithm, the state is measured. The algorithm was shown to have a quadratic speed-up compared to a classical linear search algorithm for large N[5]. The goal of the present research is, however, to repeat the Grover's amplification block as many times as possible to study the fidelity decay.

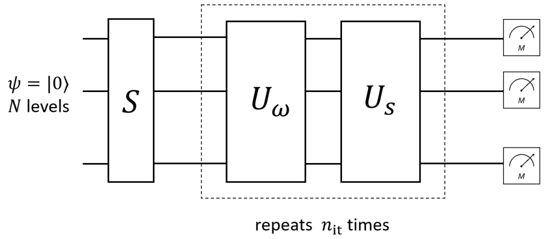

FIG. 1: Grover's search circuit. The superposition gate S transforms the ground state $\psi = |0\rangle$ to the superposition state $|s\rangle$. The block of two gates (Oracle U_ω and Grover's diffusion U_s) is then repeated $n_{\rm it}$ times. Finally, the measurements of the quantum state are performed.

Notice that Grover's algorithm operates on the state vector in the $(|\omega\rangle, |s\rangle)$ plane only. Thus, it is convenient is to carry out the analysis in the new basis $(|\omega\rangle, |s\rangle)$, so the state vector reads as $|\psi\rangle = \alpha|s\rangle + \beta|\omega\rangle$. The application of one Grover's iteration U_ω, U_s is equivalent to the application of the matrix D to $|\psi\rangle$ in the new basis $|\bar{\psi}\rangle = D|\psi\rangle$, i.e.

$$\begin{pmatrix} \bar{\alpha} \\ \bar{\beta} \end{pmatrix} = D\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{bmatrix} \frac{N-4}{N} & -\frac{2}{\sqrt{N}} \\ \frac{2}{\sqrt{N}} & 1 \end{bmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{20}$$

After a number of Grover's iterations $n_{\rm it}$, the population of the desired state p_ω is found via measurements as

$$\begin{aligned} p_\omega = |\langle\omega|\psi\rangle|^2 &= (\alpha\langle\omega|s\rangle + \beta\langle\omega|\omega\rangle)^2 \\ &= \frac{\alpha^2}{N} + \beta^2 + 2\frac{\alpha\beta}{\sqrt{N}}. \end{aligned} \tag{21}$$

Initially, the system is in the ground state, and after the application of the superposition gate it goes to the $|s\rangle$ state, so $\alpha = 1$, and $\beta = 0$. Skipping some straightforward algebra, one can show that after $n_{\rm it}$ applications of the D matrix the probability of the desired state reads as

$$p_\omega = \sin^2(n_{\rm it}\theta + \kappa), \tag{22}$$

where

$$\begin{aligned} \sin(\theta) &= \frac{2\sqrt{N-1}}{N}, \\ \sin(\kappa) &= \frac{1}{\sqrt{N}}. \end{aligned} \tag{23}$$

As pointed out, two cases are of particular interest: $N = 3$, and $N = 4$. For the latter, $\theta = \pi/3$, and $\kappa = \pi/6$, so that the probability p_ω is exactly equal to unity for every third Grover's step $n_{\rm it} = 1, 4, 7, 10....$ It can be used as an ideal tool to diagnose the fidelity of quantum gates, as the difference between the unity and measured probability shows the rate of the decay processes in the

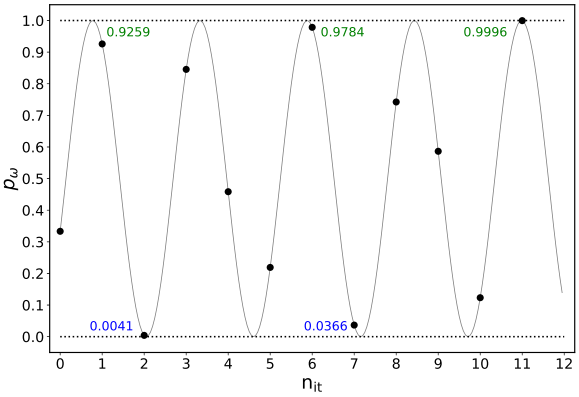

FIG. 2: The analytical probability p_ω to find the system in the desired state ω after $n_{\rm it}$ iterations of the Grover's search algorithm. Blue points: p_ω is close to zero, green points p_ω is close to unity.

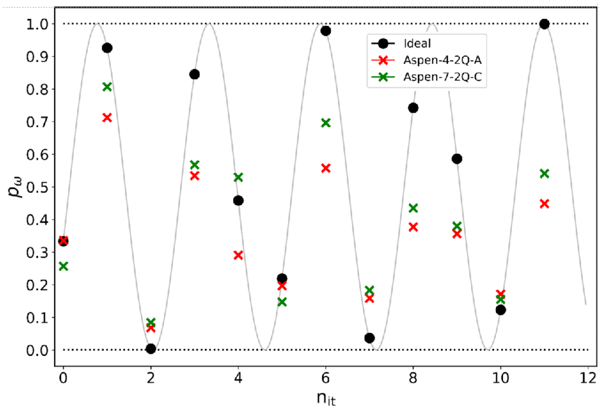

FIG. 3: Probability to find the system in the $|2\rangle$ state as a function of Grover's iterations. Results from two different Rigetti QPUs are compared to analytical theory. Every point is obtained as an average over 10000 shots.

system. For the former, $N = 3$, the probability is a periodic function of $n_{\rm it}$, with an irrational period, therefore only certain points are of interest, shown in Fig. (2). For example, points where the probability is close to unity, as these points serve the purpose of a circuit fidelity benchmark. Normally, other points are not of interest for the Grover's search algorithm, however in this particular context, points such that the probability is close to zero can be used as well. The reason is that the total number of states is small, so if $p_\omega \approx 0$, it automatically implies that the other states have population of 1/2, which is clearly distinguishable from 0, and therefore such points can be used for a study of the fidelity decay. This argument is not valid when N is large, as the populations of other states are $1/(N-1)$, which is close to zero for $N \gg 1$.

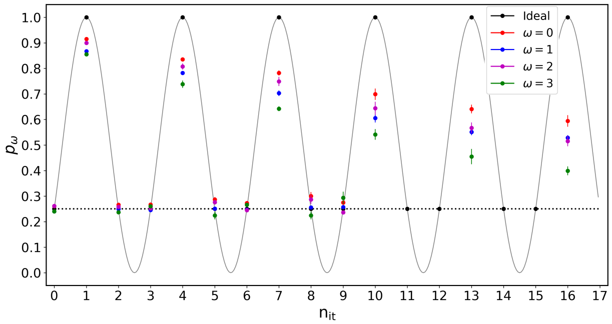

FIG. 4: Population of desired state measured as a function of $n_{\rm it}$ for the 4-level GS program run on Aspen-7-2Q-C lattice. Every point is obtained as an average over 10000 shots. The 4 colors correspond to 4 different possible oracles $\omega = 0, 1, 2, 3$.

B. Practical realization of the GS algorithm on the Rigetti platform

All the gates in the GS circuit are decomposed into a sequence of native Rigetti gates[6]. Depending on the number of the levels N, the decomposition of S and U_s gates is very different. For example, the S_3 gate has approximately 35 native gates in its decomposition with 3 CZ gates, while S_4 has only 6 native single qubit gates, as the 2-qubit Hadamard gate does not require any qubit entanglement. The similar story is for the U_s gate: U_{s3} has 26 native gates with one CZ among them, while U_{s4} has only 9 gates with one CZ. The more gates involved in the decomposition, the worse the fidelity is, and the following tests confirm that statement.

All the Rigetti circuits are built using PRAGMA PRESERVE_BLOCK[7] compiler macro in order to ensure that the total number of the native gates in every component of the GS algorithm is fixed and to disable compiler optimization. Every measurement is obtained as an average over 10K shots, so that the statistical error (standard deviation of the binomial distribution, see Appendix(A) is negligible compared to the deviation due to the hardware.

Fig. (3) illustrates the result of a 3-level GS program run on the Aspen-4-2Q and Aspen-7-2Q QPUs. The Oracle state is $\omega = 2$, however the results are not very different for the other options ($\omega = 0$ and $\omega = 1$). In all cases the fidelity drops very quickly as $n_{\rm it}$ increases, and the system seems to be decoherent after just several iteration of the Grover's diffusion operator. Moreover, even the superposition gate S_3 has non negligible infidelity in the case of Aspen-7. There are still points that match closely, for example, $n_{\rm it} = 5$ or $n_{\rm it} = 10$, which presumably occurred accidentally, taking into account how poor it is at some other points ($n_{\rm it} = 6$ or $n_{\rm it} = 11$).

The 4-level GS algorithm was also tested on the Rigetti Aspen-7-2Q machine. The results are shown in Fig. (4). While it looks somewhat better than in the 3-level GS case, the fidelity still drops rapidly and after $n_{\rm it} = 15$

the system becomes almost decoherent. Notice also the difference between the performance of the program with the oracles based on different desired states. The ground state based oracle (red color in Fig. (4)) performs considerably better than the one based on the $|11\rangle$ state, which was not apparent in the 3-level case. The reason for that might be the increased fidelity decay of the 3-level system GS.

C. Coherent errors

To further investigate the fidelity decay of the 4-level GS algorithm, every individual block was tested separately. The most interesting results were obtained when the Grover's diffusion gate U_{s4} was considered. Applied to the ground state even amount of times, ideally it should not change it at all, because

$$\begin{aligned}(2|s\rangle\langle s| - I)^2|0\rangle &= (2|s\rangle\langle s| - I)(2|s\rangle\langle s|0\rangle - |0\rangle) \qquad (24)\\ &= 4\langle s|0\rangle\langle s|s\rangle|s\rangle - 4\langle s|0\rangle|s\rangle + |0\rangle = |0\rangle.\end{aligned}$$

The results obtained from the Aspen-7-2Q QPU are drastically different and are shown in Fig. (5). The U_{s4} gate is applied to the ground state n_{it} times, and each point is obtained via averaging over 10K shots. The most prominent feature of the observed behavior is that decaying oscillations are present, therefore there are some combined coherent and incoherent errors that affect the fidelity of the measurements. Coherent errors imply that the U_{s4} gate does not perform the unitary transformation correctly, in other words, some extra parasitic rotation is added to the designed gate. Unlike decoherent errors that cannot be mitigated by any means except for improving the hardware, coherent errors can be in principle corrected by extra over/under rotation, however, their source, magnitude and direction vector have to be identified first. This is the main motivation for the gate set tomography of the native Rigetti gates.

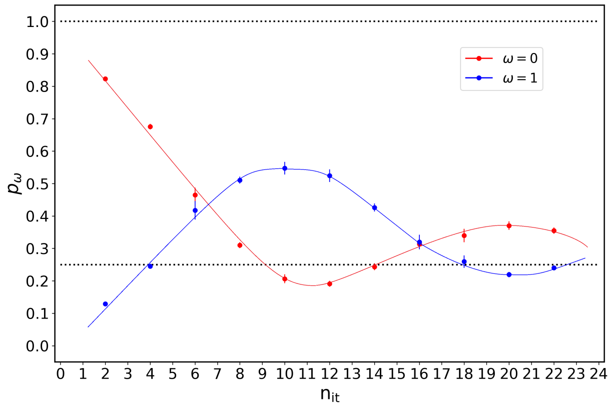

FIG. 5: Population of $\omega = 0$ and $\omega = 1$ states as a function of n_{it} for the $U_{s4}^{n_{\text{it}}}$ program run on Aspen-7-2Q-C lattice. Ideally, the population of $\omega = 0$ state should be unity, and the population of $\omega = 1$ state should be zero.

IV. HARDWARE ISSUES

In this section, we address a number of issues that arise when a quantum circuit is run on actual hardware. Most of the issues described in this section have an unpredictable nature, thus are hard to control and mitigate. Yet, having an idea of what the symptoms are might be useful when production runs are performed. The names of the issues are made up based on the observed behavior and referred to "spikes", "drifts", etc.

The main tool to diagnose these issues is measuring the qubit states at a certain point of the circuit and comparing the results to the analytic expectations. In particular, a program P that comprises of the S_3 superposition gate and the measurements of both qubits is used. Every set of the 4 measurement points (p_k, $k = 0, 1, 2, 3$) is obtained via averaging over 10K shots in order to eliminate statistical errors (see Appendix(A)). Ideally, one can expect to measure $p_0 = p_1 = p_2 = 1/3$ and $p_3 = 0$ at all times with the statistical spread of $\sigma \approx 0.0047$. As different features of the hardware are tested, the program P might be repeated many times and typically be run for a period of time of up to 30 minutes repeatedly. Observations of the data lead to the issues reported below.

A. Retune

Sometimes, even an empty program can have very poor readout properties. For example, the ground state probability $p_0 = 0.86$, which is quite far from unity. Since the program is empty and no active reset [8] has been performed, the issue is due to the state preparation and measurements (SPAM). Having at most 86% fidelity for any measurement is undesired, especially if the program of interest has many gates and therefore is suffering from even more fidelity decay. According to the response of the Rigetti team, the reason for this problem *is caused by readout characterization drift caused by changes in the ambient temperature and device operating point inside the fridge. This is specific to the readout circuit, and not necessarily related to gate fidelity or other measures of overall performance. Aspen-7 has been noted by the team as suffering from readout classification error, more than Aspen-4.* This is part of the natural design iteration process the team was going through and expected improvements in the future. This problem is solved by a *retune* of the qubits, so the remedy is quite simple, namely to perform all the production runs immediately after a retune. Retunes are typically performed several times per day. Running the program after a retune improves the readout fidelity to 94%, which is far from perfect but close to the parameters reported on the Rigetti lattices page[1] fRO = 93.55% (fRO = 93.55%).

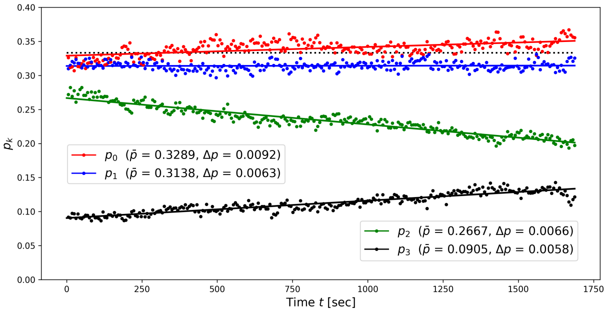

FIG. 6: Measured populations of four states (p_0, p_1, p_2, p_3) as functions of time. The program is the S_3 gate, run on Aspen-4 lattice. This type of a behavior is called "drifts".

B. Drifts

Drifts represent the processes of slow, yet considerable, change of the readout fidelity. An example of such a drift is shown in Fig. (6). The program P was run on the Aspen-4 machine for about 25 minutes, and the time evolution of p_k for all k was collected. A strong drift of the population values was observed instead of the theoretical constant values 1/3 for $k < 3$ and 0 for $k = 3$. While p_0 and p_1 are more or less close to the estimated values, p_2 and p_3 drift uncontrollably. This behavior can be explained as a failure of the second qubit, because the ground and the first states are given by $|00\rangle$ and $|01\rangle$, and the other two states by $|10\rangle$ and $|11\rangle$ where the second qubit is in excited state.

Fortunately, strongly drifting behavior is relatively rare and can be detected by repeating the same program several times in a reasonable period of time, like 15-30 minutes. Some means of control are needed in order to guarantee the validity of the results.

C. Spikes

Spikes are a different type of irregular behavior of the measured population numbers. Similarly to the case of drifts, the program P was run for 30 minutes and the data was collected. Unlike drifts, the population values were stationary except for a short period of time approximately from 400 to 500 seconds. During this time, all 4 populations behaved very chaotically and irregularly with no apparent trend, and the system returned to its quiet state just after 500 seconds. We could not determine what caused such behavior, however it was very rarely observed, so it is less of a concern.

Spikes are easy to detect even while a production run. In order to do so, one needs to repeat the same program several times (5-10 repetitions should be enough) and check the standard deviation of this small sample. If any spikes occur, the standard deviation will immediately increase and be detected.

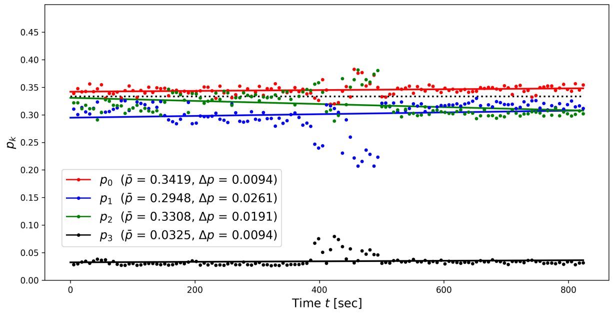

FIG. 7: Measured populations of four states (p_0, p_1, p_2, p_3) as functions of time. The program is the S_3 gate, run on Aspen-4 lattice. This type of a behavior is called "spikes".

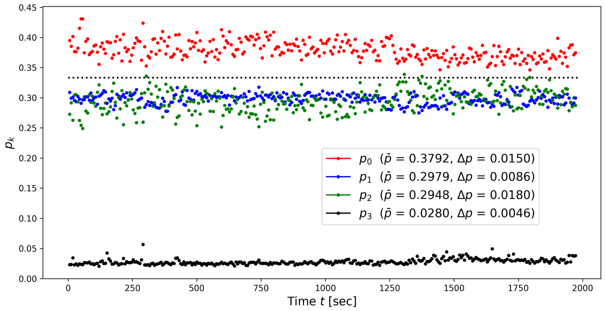

FIG. 8: Measured populations of four states (p_0, p_1, p_2, p_3) as functions of time. The program is the S_3 gate, run on Aspen-4 lattice. The standard deviation is about 3 times greater than the one estimated from the binomial distribution.

D. Anomalous noise

The other interesting feature are periods of *anomalous noise*. When the same program is run many times, one can assume the results are governed by the binomial distribution. If the properties of the system are unchanged, the standard deviation depends on the number of measurements N_m only (see Appendix(A)) and scales as inverse of the square root of N_m. However, this is not what was observed in some runs of Aspen-7 machine. Fig. (8) illustrates this phenomenon. Again, the program P was run many times, no drifts or spikes were detected, yet the spread of the data is considerably greater than $\sigma = 0.0047$.

Notice that such behavior was quite common at the early stages of the project, namely in January-February 2020, and became very rare later on (April-May). While the cause was not revealed, it appears as if it was some kind of permanent low amplitude spikes issue. While there is no means of control and mitigation of such a thing, it can be easily detected by performing the same trick as with spikes, namely, sampling every data point over 5-10 shots and checking if the standard deviation is still in the limits of the binomial distribution.

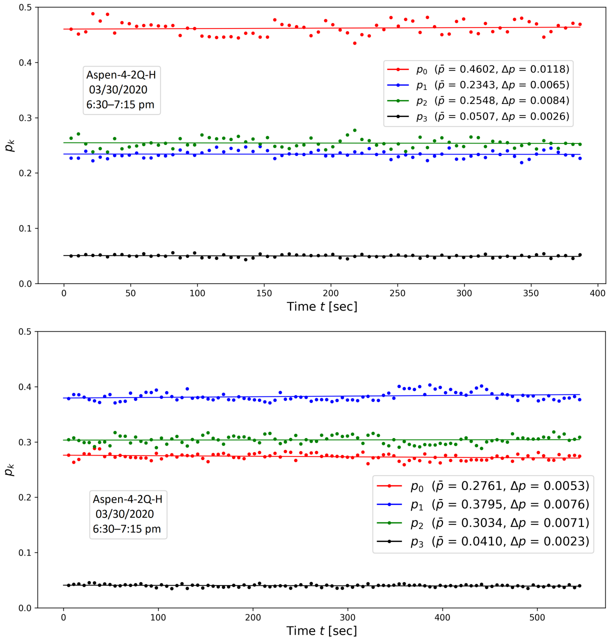

FIG. 9: Measured populations of four states (p_0, p_1, p_2, p_3) as functions of time. The program is the S_3 gate, run on Aspen-4 lattice. The only difference is in the native gate decomposition. Top: decomposition 2, bottom: decomposition 1, corresponding to Table (I).

E. Native gate decomposition dependence

Notice that the mean values of p_0, p_1, and p_2 in Fig. (8) are considerably different, while all of them were expected to be 1/3. It might initially seem that the system is not properly calibrated, so the ground state has higher probability, and, indeed, $p_0 > p_{1,2}$. This was not the case, however. The main reason was the difference of the native gate decomposition. Rigetti uses a compilation algorithm that has a random component, therefore the gate decomposition can vary from one instance to another. For example, Fig. (9) demonstrates how two different native gate decompositions can lead to completely different performance of the program P. The examples of such decompositions are shown in Table (I). They have the same number of CZ gates but different numbers of single qubit gates. Ignoring the global phase, the resulting unitary matrices are identical up to machine precision, therefore the conclusion can be drawn that some coherent errors are responsible for such pronounced differences between the two decompositions when they are run on the QPU.

Decomposition 1	Decomposition 2
RZ(1.901868944617673) 1	RZ(-1.2397237089721242) 1
RX(pi/2) 1	RX(pi/2) 1
RZ(0.84460947696346) 1	RZ(2.2969831766263367) 1
RX(-pi/2) 1	RX(-pi/2) 1
RZ(1.0931380177326422) 1	RZ(2.048454635857148) 1
RX(pi/2) 2	RZ(pi) 2
RZ(1.2199169159226386) 2	RX(pi/2) 2
RX(-pi/2) 2	RZ(1.9216757376671576) 2
RZ(-0.7853981633974487) 2	RX(-pi/2) 2
CZ 2 1	RZ(-2.356194490192344) 2
RZ(-1.5707963267948941) 1	CZ 2 1
RX(pi/2) 1	RZ(-pi/2) 1
RZ(3*pi/4) 1	RX(pi/2) 1
RX(-pi/2) 1	RZ(3*pi/4) 1
RZ(-pi/2) 2	RX(-pi/2) 1
RX(-pi/2) 2	RZ(-pi/2) 2
CZ 2 1	RX(-pi/2) 2
RX(pi/2) 1	CZ 2 1
RZ(-1.6940492545467922) 1	RX(pi/2) 1
RX(-pi/2) 1	RZ(-1.6940492545467922) 1
RZ(1.6940492545467922) 2	RX(-pi/2) 1
RX(pi/2) 2	RZ(1.6940492545467922) 2
CZ 2 1	RX(pi/2) 2
RZ(1.093138017732643) 1	CZ 2 1
RX(pi/2) 1	RZ(2.048454635857151) 1
RZ(0.8446094769634596) 1	RX(pi/2) 1
RX(-pi/2) 1	RZ(2.296983176626333) 1
RZ(-1.2397237089721216) 1	RX(-pi/2) 1
RZ(-1.0931380177326417) 2	RZ(1.9018689446176724) 1
RX(pi/2) 2	RZ(2.048454635857145) 2
RZ(0.84460947696346) 2	RX(pi/2) 2
RX(-pi/2) 2	RZ(0.8446094769634598) 2
RZ(1.2397237089721203) 2	RX(-pi/2) 2
	RZ(1.2397237089721203) 2

TABLE I: Two different decompositions of the same gate S_3. While, theoretically, these two decompositions are nearly indistinguishable, they differ significantly in practice when the list of gates is run on Rigetti QPUs.

V. NATIVE GATE BENCHMARKS

A. RZ gate benchmark

Aspen-7-2Q-C lattice was used for the benchmarks of the RZ gate. There were two tests: one immediately after a retune, and the other approximately 1.5 hours later. The program consists of the RZ gate with $\phi = \pi/4$ applied to either the 0 or 1 qubit and the measurement of all qubit states. The gate was applied $n_{\rm it}$ times. Fig. (10) shows that the result does not depend on $n_{\rm it}$, as it was expected since RZ is a software gate. It does not depend on the qubit the gate was applied to for the same reasons. Basically, this program is just equivalent to the measurement of the ground state.

The result, however strongly depends on the retune time, which is demonstrated in Fig. (11). The same loss of fidelity (from 0.94 to 0.86, as reported before) was observed. Furthermore, analyzing the results in Fig. (11),

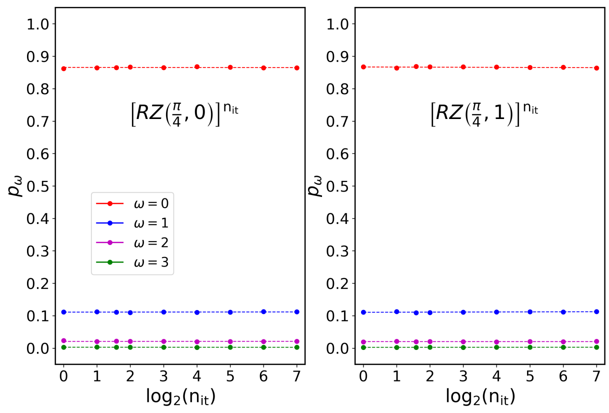

FIG. 10: Measured populations of all the four states after the RZ($\pi/4$) gate was applied n_{it} times to 0 qubit (left) or 1 qubit (right). The program was run approximately 1.5 hours after retune.

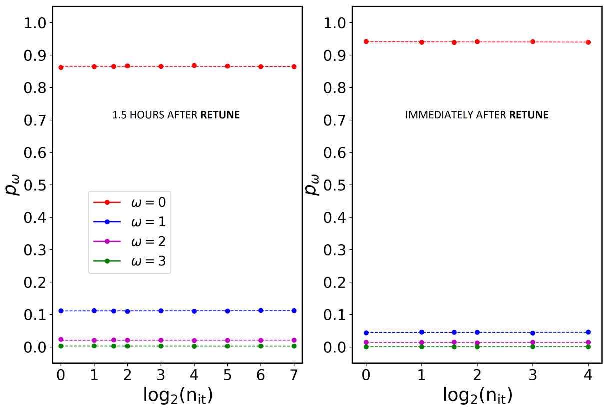

FIG. 11: Measured populations of all the four states after the RZ gate applied n_{it} times. Left: 1.5 hours after retune, right: immediately after retune.

one can notice that only one qubit is actually having performance issues and needs some retuning. Indeed, only the $\omega = 0$ and $\omega = 1$ states are affected (red and blue lines) and they correspond to $|00\rangle$ and $|01\rangle$. The population of the other two states ($|10\rangle$ and $|11\rangle$) remains nearly zero, so the second qubit does not require a retuning.

B. RX gate benchmark

The same Aspen-7 machine and the same approach was used to benchmark the RX gate. It was applied to either of the qubits with the parameter $\pi/4$. As was mentioned in Sec. (II), the only native RX gate is RX($k\pi/2$), so, in fact, some decomposition was benchmarked instead of the native RX gate [9]. These tests reveal some coherent errors in the nature of the RX gate, which was one of the main motivations for this benchmark.

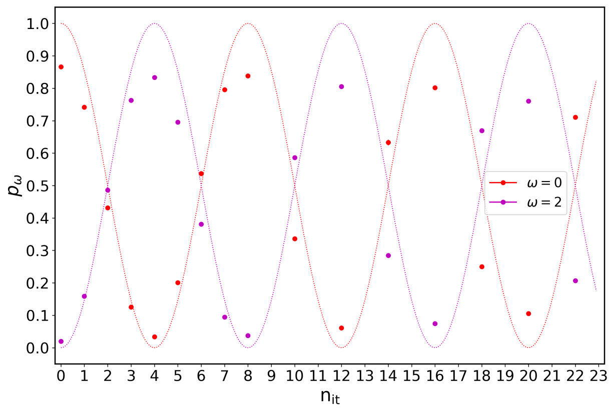

FIG. 12: Measured populations of the $\omega = 0$ and $\omega = 2$ states, as the RX($\pi/4, 0$) gate is applied n_{it} times. Dashed: ideal theoretical population oscillations, solid points - experimental data.

When the RX gate is applied to the 0 qubit, the population densities of $\omega = 0$ and $\omega = 2$ states will oscillate periodically

$$p_0 = \cos^2(n_{it}\phi/2), \qquad (25)$$
$$p_2 = \sin^2(n_{it}\phi/2),$$

where $\phi = \pi/4$ in our setup. Neglecting the effects related to the state measurement errors (94% at best), one can observe in Fig. (12) how this analytical prediction is violated and the experimental results get out of phase as n_{it} grows up. In order to find this phase mismatch, we assumed that every time the RX gate is applied, a phase shift $\Delta\phi$ appears. This phase shift can be found by performing least squares fit of the experimental data using the form

$$p_0 = A_0 \cos^2(n_{it}[\phi/2 + \Delta\phi]), \qquad (26)$$
$$p_2 = A_2 \sin^2(n_{it}[\phi/2 + \Delta\phi]),$$

where $A_{0,2}$ are the amplitudes to compensate readout fidelity losses. Fig. (13) demonstrates a good fit between the experimental data and Eq. (26). As the result, $\Delta\phi$ was found for both qubits the RX gate was applied to:

$$\Delta\phi_0 = 0.0145, \qquad (27)$$
$$\Delta\phi_1 = 0.009.$$

It would be interesting to see the decomposition of RX($\pi/4$), as it might contain several RZ and RX gates and it is not yet clear which of them are responsible for coherent errors. More benchmarks of RX($\pi/2$) are required to fully understand the phenomenon.

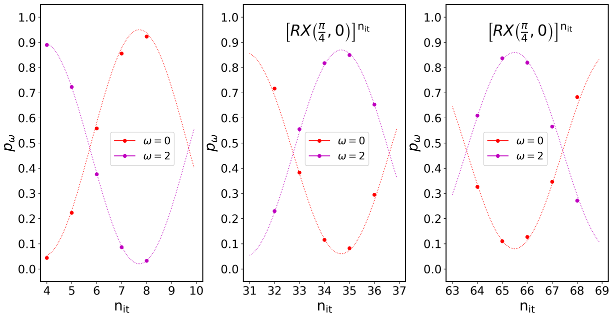

FIG. 13: Measured populations of $\omega = 0$ and $\omega = 2$ states at three different parts of n_{it} scale. Dashed line: least square fit of the data with shifted analytical expression.

C. CZ gate benchmark

Two different approaches were used to benchmark the CZ gate. Since the gate only changes the sign of the $\omega = 3$ state, applying the gate numerous times to the ground state might not be very informative. Therefore the system was prepared in the $\omega = 1$ or $\omega = 2$ states by applying RX$(\pi, 0)$ or RX$(\pi, 1)$ gates respectively, and then the CZ gate was applied n_{it} times. In Fig. (14), both scenarios are shown. The main difference between them is that one set of measurements was performed immediately after a retune ($\omega = 2$ case) and the other was done some time after ($\omega = 1$ case). This circumstance significantly changed the readout fidelity of the prepared states (approximately from 0.9 to 0.8) and the state collapse time from 128 to 32 CZ gate time intervals. Here, the state collapse time is defined as the time it takes for the population of the ground state to reach the population of the excited state, measured in CZ gate time intervals. This approach can be used to estimate T_1 time of the qubits. If the time of the CZ gate is about 200 ns [10], then application of 100 gates takes about 20 μs. By that time the excited state loses half of it population, so that $T_1 \approx 20$ μs. According to Rigetti report, the T_1 time of Aspen-7 was about 60 μs that is on the same order of the estimated value but greater by a factor of 3. The nature of this discrepancy remains unknown, therefore more benchmarks are needed to reveal it.

The second set of benchmarks is the application of either $U_0 = \text{RX}(\pi, 1) \cdot \text{RX}(\pi, 0) \cdot \text{CZ}(0, 1)$ or a pure CZ gate. The U_0 gate only changes the phase of the $\omega = 0$ state and the CZ gate changes the phase of the $\omega = 3$ state. Since the U_0 gate has two RX gates inside, which are far from ideal, one can expect considerably different performance for these two cases. However, the difference was nearly indistinguishable, as shown in Fig. (15). The result of this benchmark is not understood for now and needs further investigation.

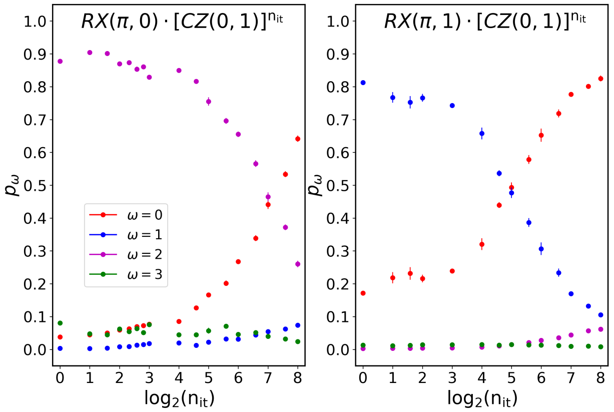

FIG. 14: Measured populations of all the states as a function of n_{it} of the CZ gate applied. Left picture: the $\omega = 2$ state is excited, measurements are taken immediately after a retune, right: the $\omega = 1$ state is excited, measurements are taken some time after a retune.

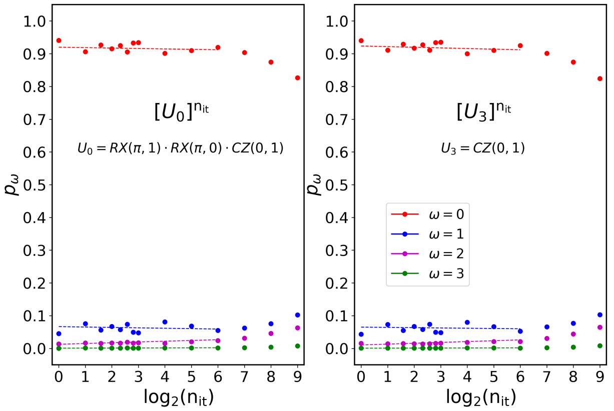

FIG. 15: All four states populations are measured as functions of n_{it} of either U_0 or CZ gates applied.

VI. DISCUSSION AND CONCLUSION

Running Grover's search (both 3-level and 4-level versions) can be an excellent illustration of Rigetti capabilities, weak points and areas of possible improvements. As demonstrated in the present report, a lot of efforts should be made in order to improve the results of the GS runs on Aspen-4 and Aspen-7 QPUs.

- All production runs should be performed immediately after a retune, as the readout fidelity can be improved from 0.86 to 0.94. While it remain unclear how a retune affects performance of the gates, the measurements of different programs are noticeably affected.

- A fixed native gate decomposition should be used for all runs of the program. One very important mistake that was made when the initial GS program was run on Aspen-4/7 QPUs, was that the native gate decomposition was not fixed for all iterations n_{it}. As a result, it was in fact a different program that ran each time, and since the hardware coherent errors of the native decomposition can be significant, the results were noticeably affected in an uncontrollable way. The solution of this problem is relatively simple. Once a decomposition is obtained, it should be fixed for all iterations n_{it}. Moreover, the decomposition can be tested individually before the production run of the GS program, and only the best decompositions can be used.

- By default, the accuracy of the native gate decomposition of a desired unitary transformation is as small as machine precision. That leads to a large number of native gates required to represent one unitary transformation (up to 35 gates for the S_3 gate). Since every gate brings more errors causing overall fidelity decay, it might be beneficial to reduce the accuracy of the decomposition and decrease the total number of gates. However, the dependence of the decomposition accuracy on the number of gates and even such a possibility remain unknown, therefore more detailed research is required.

- Many hardware issues can be mitigated by performing a small batching of the data. Such an approach can, for example, distinguish *spikes* and *anomalous noise.* The cost is that the runtime is increased by a factor 5-10. However it is always better to run a longer program than to run a short program many times and collect untrustworthy results.

- Ideally, native gates can be corrected in order to eliminate *coherent errors.* Decoherent errors, such as T_1 or T_2 processes (decoherence, i.e. a collapse of the wavefunction to the ground state, and dephasing respectively) cannot be cured, yet coherent errors, when the gate does not perform the desired unitary transformation, can be fixed by applying extra corrections, therefore gate set tomography (GST) for all the native gates should be performed.

Everything except for the last item can be done in a reasonable amount of time, as the required experience and expertise with Rigetti platform have already been obtained. The pyquil scripts for most of the tasks have been written. The last item, however, might be a more complicated task, as it requires many different tests. In fact, the gate correction can be realized with the RZ gate only. Indeed, the other two, RX and CZ, are non-parametric gates, so even if the full GST of them is performed and corrective unitary operators are found, the only way these corrections can be implemented is via a combination of these gates and a number of RZ gates with the correct parameters. We might expect that the total number of gates to correct coherent errors might increase considerably, so that more decoherent errors will be introduced. In this case a precise balance of terms might be needed, and, what is most important, this balance might also depend on the problem of interest and on when the optimization is performed.

VII. ACKNOWLEDGMENTS

This work was performed under the auspices of U.S. Department of Energy (DOE) by Lawrence Livermore National Laboratory (LLNL) under Contract No. DE-AC52-07NA27344 and was supported by the DOE Office of Fusion Energy Sciences “Quantum Leap for Fusion Energy Sciences” project FWP-SCW1680 and LLNL Laboratory Directed Research and Development (LDRD) project 19-FS-072.

APPENDIX A: BINOMIAL DISTRIBUTION

Consider a random variable X that can take values either 1 or 0 with probability p and $(1-p)$ respectively, where $0 < p < 1$. The variable is measured independently n times, and the normalized total number of success is denoted as $Y = \sum_i^n X_i/n$. The random variable Y is then has binomial distribution with the mean value p and the variance $p(1-p)/n$. In the limit of large n, when p is not *too close* to either 0 or 1 (pn or $(1-p)n$ grows unboundedly as n grows), the distribution may be approximated by the normal distribution with the same mean value, $\mu = p$, and the variance $\sigma^2 = p(1-p)/n$.

For typical values of p and n used in this work ($p \approx 2/3$, $n = 10^4$), one can estimate the standard deviation to be $\sigma \approx 0.0047$. This estimate provides a bound on the statistical spread of the data collected via measurements. This estimate might not be correct for the cases of p too close to 0 or 1, for example, $p = 0.9997$ for at $n_{\text{it}} = 11$ iteration of the 3-level GS algorithm, but for the majority of the measurements it is still a very good estimate.

[1] Rigetti Computing URL: https://www.rigetti.com.
[2] S. A. Caldwell *et al.*, Phys. Rev. Applied **10**, 034050 (2018).
[3] Rigetti native gates can be found at URL: https://pyquil-docs.rigetti.com/en/stable/apidocs/gates.html.
[4] H. O. Kunz, IEEE Trans. Comps. **C-28**, 267 (1979).
[5] L. K. Grover, “A fast quantum mechanical algorithm for database search,” (1996), arXiv:9605043 [quant-ph] .
[6] R. S. Smith, M. J. Curtis, and W. J. Zeng, “A prac-

tical quantum instruction set architecture," (2017), arXiv:1608.03355 [quant-ph] .

[7] PRAGMA PRESERVE_BLOCK URL: https://pyquil-docs.rigetti.com/en/1.9/compiler.html.

[8] Active reset is the command that resets the system to the ground state manually, instead of waiting long enough until the system collapses naturally. Active reset is usually performed for the sake of time conservation, yet this process can affect the fidelity of state preparation, making it not pure ground but somewhat close.

[9] The decomposition of the RX($k\pi/4$) gate is unknown but can be recovered, once the access to Rigetti platform is restored.

[10] S. S. Hong, A. T. Papageorge, P. Sivarajah, G. Crossman, N. Didier, A. M. Polloreno, E. A. Sete, S. W. Turkowski, M. P. da Silva, and B. R. Johnson, Physical Review A **101** (2020), 10.1103/physreva.101.012302.