QURE: Qubit Re-allocation in Noisy Intermediate-Scale Quantum Computers

Abdullah Ash- Saki
Pennsylvania State University
University Park, PA
ash.saki@psu.edu

Mahabubul Alam
Pennsylvania State University
University Park, PA
mxa890@psu.edu

Swaroop Ghosh
Pennsylvania State University
University Park, PA
szg212@psu.edu

ABSTRACT

Concerted efforts by the academia and the industries e.g., IBM, Google and Intel have brought us to the era of Noisy Intermediate-Scale Quantum (NISQ) computers. Qubits, the basic elements of quantum computer, have been proven extremely susceptible to different noises. Recent experiments have exhibited spatial variations among the qubits in NISQ hardware. Therefore, conventional mapping of qubit done without quality awareness results in significant loss of fidelity for a given workload. In this paper, we have analyzed the effects of various noise sources on the overall fidelity of the given workload for a real NISQ hardware. We have also presented novel optimization technique namely, Qubit Re-allocation (QURE) to maximize the sequence fidelity of a given workload. QURE is scalable and can be applied to future large scale quantum computers. QURE can improve the fidelity of a quantum workload up to 1.54X (1.39X on average) in simulation and up to 1.7X in real device compared to variation oblivious qubit allocation without incurring any physical overhead.

CCS CONCEPTS

• **Hardware** → *Quantum error correction and fault tolerance*;

KEYWORDS

Quantum Computing, Fidelity, Decoherence, Noise, Gate-error.

ACM Reference Format:

Abdullah Ash- Saki, Mahabubul Alam, and Swaroop Ghosh. 2019. QURE: Qubit Re-allocation in Noisy Intermediate-Scale Quantum Computers. In *The 56th Annual Design Automation Conference 2019 (DAC '19)*, June 2–6, 2019, Las Vegas, NV, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3316781.3317888>

1 INTRODUCTION

Following Feynman's idea of using quantum computer based on the intricate principles of quantum mechanics, the scientific community has embarked on a quest to tap into the unprecedented potential of quantum computing. The concerted effort by industry and academia has produced commercial quantum computers and algorithms which, at least in principle, offers speed-up over classical counterparts. The basic element of a quantum computer is a qubit that has to satisfy DiVincenzo criteria [5]. Since a number of quantum systems satisfy them, several competing technologies exist such as superconducting [17], ion-trap [1], solid-state [15] qubits, to name a few.

In spite of all the promises and potentials, quantum computers are still in nascent stage. This is due to fragile and noise/error-prone qubits (we will use noise and error interchangeably). This class of quantum computers is called Noisy Intermediate-Scale Quantum (NISQ) computers [16]. Qubits interact with the environment and hence the qubit states deteriorate and get corrupted over time. This phenomenon is known as *decoherence* [11] and is usually characterized by the relaxation time T_1 and the dephasing time T_2 . Moreover, the gate operations on qubits are erroneous e.g., the microwave

pulses that implement a certain gate may cause *leakage* into non-computational states limiting the precision of the gate [17].

There may be several different orthogonal approaches to solve the noise issue. First, the obvious one, is to build a better qubit. There has been a thousand-fold improvement in decoherence time in last two decades [19]. However, there are still some grounds to break. The second approach is Quantum Error Correction (QEC) (e.g., Shor code, Steane code, Surface code) However, QECs have substantially large qubit overhead (e.g., 18X extra qubit in surface code [7]). Existing universal quantum computers have limited number of qubits (e.g., Google's Bristlecone has 71 qubits [10]) which makes QEC expensive. Another approach is probabilistic which is followed in IBMQX [4]. In this approach, the same quantum workload is executed on the target hardware many times and the outcome that appears in the highest number of trials is the output.

Existing quantum computers exhibit qubit-to-qubit variation in terms of error-rates. Therefore, blind allocation of gates to the qubits may not be optimal for fidelity (or, probability of correct output). In this paper, we present "QURE: Qubit RE-allocation" a asymmetric scheduling of quantum gates where high-quality qubits are prioritized and utilized more often than poor quality qubits leading to a more reliable performance and better fidelity. As a proof-of-concept, we executed a quantum circuit (Figure 1(b)) in IBMQX4 5 qubit quantum computer [4] (Figure 1(a)). In test case - I, the circuit ((Figure 1(b))) is implemented on qubit pair (2, 1) with better performance specs than qubit pair (3, 2) (test case - II). The distributions in Figure 1(d) and Figure 1(e) corroborates to our intuition as the operations on better qubit pair resulted in much higher correct outcomes (868 correct outputs among 1024 trials, 83.8%) than operations on lesser qubit pair (only, 489 correct outputs among 1024 trials, 47.8%). For the same quantum workload for IBMQX4 (Figure 1(a)), multiple qubit assignment choices exist (Figure 1(c)). We executed the test circuit on the other qubit pair of IBMQX4 and the observation, in general, remains valid with Q(2, 1) resulting in the best result.

Although the above intuition seems straight-forward, there are several hurdles to overcome. For example, various sources of error and noise in a real quantum computer exist as mentioned earlier. To make any gate scheduling decision, the relative impact of each type of noise source has to be understood so that one type of error can be prioritized over the other. However, figuring out this relative impact may be challenging by executing programs on an actual quantum computer since all noise components exist simultaneously. A simulation framework, therefore, may be useful to gauge the system performance with respect to an isolated noise component. Moreover, gate assignments cannot be done randomly. The architecture of the real device, connectivity between qubits, allowed native gate-sets etc. has to be taken into account to make a holistic decision for an, eventually, improved performance.

To this end, we make the following contributions in this paper. We (a) develop a simulation framework with real-device performance parameters to emulate circuit performance with isolated noise sources, (b) propose heuristic to assign gate operations to better qubits more often to improve overall fidelity compared to variation agnostic qubit assignment, and (c) verify QURE through simulations and running on real quantum computer.

The paper is organized as follows: In Section 2, we delineate modeling of the noisy quantum system and the simulation flow. In Section 3, we discuss contemporary strategies to allocating gate operations to qubits. In Sections 4 and 5, we present QURE methodology and results, respectively. Conclusions are drawn in Section 6.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DAC '19, June 2–6, 2019, Las Vegas, NV, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6725-7/19/06...\$15.00

<https://doi.org/10.1145/3316781.3317888>

Figure 1: Distribution of observed output for a quantum operation on IBMQX4 (Tenerife - 5 qubit quantum computer). Same sets of gate operations when applied on different qubit pair resulted in different probabilities of correct result due to qubit quality variation. (a) Topology and coupling graph of IBMQX4; (b) test circuit (workload). Qubit Q_A is expected to result in a $|0\rangle$ while Q_B is expected to give out $|1\rangle$; (c) all possible mapping of the selected workload; (d) applying the gates on qubit pair (2, 1) resulted in highest % of expected result. X - axis shows the measured qubit states ($Q_0Q_1Q_2Q_3Q_4$) in Z-basis (gate sequence is repeated 1024 times on each qubit pair to obtain the distribution); (e) transferring the operation to different qubit pair (3, 2) results in substantially degraded performance; (f) the table lists % of correct observations for several qubit pairs.

2 MODELING AND SIMULATION SETUP

In this section, we describe the evaluation platform used to test a quantum circuit performance under noise.

2.1 State Vector, Density Matrix, and Fidelity

Qubit is analogous to classical bits 0 and 1 with a key difference: qubit can be in a superposition state i.e., a combination of 0 and 1 at the same time. Qubit state is expressed with a *ket* ($| \cdot \rangle$) notation which is represented by a column matrix known as *state vector*. A qubit state $|\psi\rangle$ is described as $|\psi\rangle = a|0\rangle + b|1\rangle$. Here, $|0\rangle$ and $|1\rangle$ are known as computational basis states and a and b are complex numbers such that $|a|^2 + |b|^2 = 1$.

An alternate approach of representing qubit state is the *density matrix* (ρ) formalism which is expressed as $\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|$ where p_i is the probability of pure state $|\psi_i\rangle$. Density matrix representation is advantageous as it can express mixed states. In a noisy quantum computer, qubits may end up in a mixed state. Therefore, it is customary to use density matrix formalism when noise is considered.

Fidelity is a quantity to measure the *closeness* of two quantum states. Fidelity between two states, ρ and σ is defined, mathematically, as $F = \text{trace}(\sqrt{\rho^{1/2} \sigma \rho^{1/2}})$ [14]. Fidelity is 1 for same states. However, if one state deviates from the other, fidelity will decrease. Higher fidelity means higher probability of getting correct output.

2.2 Modeling of Noisy Quantum System

2.2.1 Gate Error. The gate errors are mimicked with a depolarizing noise channel that consists of all Pauli errors: bit-flip error (Pauli X), phase-flip error (Pauli Z) and bit-phase-flip error (Pauli Y). Under depolarizing noise, the qubit retains its state with a probability of $(1 - p)$ (p = probability of error) and undergoes X, Y and Z errors with a probability of $(p/3)$ each. Rigetti reports average gate fidelity (F) of its QPU to characterize the gate error. The reported average gate fidelity is translated to probability of error used in the Kraus operators with relation $F(p) = 1 - p/2$ [9] for 1-qubit gate error and $F(p) = 1 - 3p/4$ for two-qubit gate error.

2.2.2 T1 Relaxation. T1 relaxation is simulated with amplitude damping channel. Note that the T1 relaxation affects only the state $|1\rangle$ (i.e., $|1\rangle \rightarrow |0\rangle$) leaving state $|0\rangle$ invariant. Reported T1 times are converted to probability (of no error) using the formula $p = \exp(-t/T_1)$ where t is the time of operation that depends on the gate-time of a particular quantum computing hardware. For example, for Rigetti 8Q-Agave QPU 1-qubit gate-time is about 50ns [17].

2.2.3 T2 Dephasing. T2 dephasing is simulated with phase damping channel. Phase damping is a quantum-mechanical phenomenon and therefore, difficult to comprehend intuitively. Mathematically, the off-diagonal elements of a density matrix representing the qubit state decay to 0 due to T2 dephasing or phase damping. For example, Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$ is an entangled state. If the qubits in Bell state undergo dephasing then eventually the off-diagonal terms in

Kraus Operators

Figure 2: Kraus operators for different noise channels and extending Kraus operators for multi-qubit case.

the density matrix become zero and entangled qubits end up in a mixed state. Entanglement is believed to be one of the key properties that fuel quantum computers' computing ability and dephasing is detrimental to that. Reported T2 times are converted to probability (of no error) using the formula $p = \exp(-t/T_2)$.

2.2.4 Operator-sum Representation. To simulate the effect of noise on the quantum computation, we adopt the operator-sum representation using the appropriate Kraus operators [12]. In this representation, a quantum operation $\mathcal{E}()$, that maps the input state ρ_{in} to output state ρ_{out} such that $\rho_{in} \mapsto \mathcal{E}(\rho_{in}) = \rho_{out} = \sum_k E_k \rho_{in} E_k^\dagger$. E_k is called the operation element. By choosing appropriate operation elements, the operator-sum representation can be used to compute the output after applying a gate on a qubit.

Likewise, if appropriate Kraus operators are chosen as operation elements, the operator-sum representation can be used to simulate the effect of different errors on a qubit state. In this paper, we emulate the noisy quantum-processing-units' behavior with gate-error, T1 relaxation and T2 dephasing with suitable Kraus operators. The Kraus operators are listed in Figure 2.

2.2.5 Simulation Flow. Figure 3(a) shows the schematic of the Python-based simulation platform we developed. We use modules from QuTip [8] to execute matrix operations pertinent to quantum computation. To simulate the behavior of a real quantum device, the simulator takes: (i) the input states of the qubits in density matrix format, (ii) 1-qubit gate error probabilities, (iii) 2-qubit gate error probabilities for each allowed qubit pair, (iv) T1 relaxation times, (v) T2 dephasing times, (vi) 1-qubit and 2-qubit gate times and, (vii) a quantum program compiled with native gate-sets of a specific hardware. We primarily used Rigetti quantum processing units reported specifications [2]. However, it can simulate another quantum

Figure 3: (a) Diagram of the simulation platform and the program flow; and (b) 8Q and modeled 9Q-square architecture.

Figure 4: Comparison between fidelities from the simulation model and IBMQX4 real device.

processor (e.g., IBMQX4) if aforementioned (ii) - (vii) items are fed for that particular quantum processor.

We used Rigetti QuilC compiler to compile a circuit to native gate-sets. Since QuilC lacked compilation capability for its 8Q Agave architecture (Figure 3(b) (top)) at the time of writing this paper, we used the 9Q-square architecture (Figure 3(b) (bottom)) that resembles 8Q-Agave. It has one extra qubit in the middle (qubit - 4). The performance parameters of the identical qubits of the 9Q structure are kept the same as the 8Q structure (e.g., qubit-5 of 8Q structure is identical to qubit-6 of 9Q structure). For the additional qubit in the center (Qubit - 4) following assumptions are made: 1-qubit gate error, T1 time and T2 time are averages of other 8 qubits. The 2-qubit gate-error and gate-time between Q4 and one of the nearest neighbors is taken to be the average of 2-qubit gate errors and gate-times of the nearest neighbor respectively. For example, 2-qubit gate error between qubit pair (4,1) is the average of Q(1,0) and Q(2,1) 2-qubit gate errors.

The simulator reads the quantum program and executes each gate instruction. First, an ideal gate is applied to the qubit states followed by the errors in the following sequence: gate error, T1 error and T2 error. The simulator runs two copies of operations on the qubit states simultaneously, one with errors and another without errors. In the end, two sets of output density matrices are fed into a fidelity calculator which generates the fidelity of the program on the NISQ computer. Higher fidelity corresponds to better performance.

Table 1: Performance specification of modeled QPU. F1Q and F2Q are 1-qubit gate and 2-qubit gate average gate fidelity.

Qubit	F1Q	T1(μ s)	T2(μ s)	Pair	F2Q	Pair	F2Q
0	0.956	10.7	10.6	0-1	0.91	4-7	0.77
1	0.951	10.0	9.2	0-3	0.93	5-8	0.82
2	0.982	15.5	12.5	1-2	0.92	6-7	0.67
3	0.970	14.2	18.5	1-4	0.915		
4	0.961	14.6	26.2	2-5	0.91		
5	0.962	14.7	12.8	3-4	0.93		
6	0.969	14.2	12.9	3-6	0.93		
7	0.932	13.2	17.7	4-5	0.865		
8	0.969	14.7	12.8	7-8	0.87

Figure 5: (a) Fidelity distribution of QFT-3 benchmark under different error sources in isolation. The distribution shows that 2-qubit gate error is the dominant error source; (b) mean fidelity and gate counts for several benchmarks showing that 2-qubit gate error causes maximum loss in fidelity.

To validate the model, we simulated the same circuit as in Figure 1(b) using the model with IBMQX4 [4] specs and program compiled in IBM native gate-set. The comparison between model data and real-device data is shown in Figure 4 (IBMQX4 fidelity = correct trials/total trials). The model slightly underestimates maximum fidelity and overestimates minimum fidelity with an average error of about $\approx 7\%$. However, the model data follows the real-device trend and shows maximum and minimum fidelities at exactly same qubits as in the real device.

2.3 Relative Impact of Each Noise Source

Understanding the relative effect of each noise source is paramount to make re-allocation decision described in Section 4. We devise a simulation using the proposed platform. 9Q-square architecture is used with an average of the values listed in Table 1. Only one type of error is considered at a time by zeroing other error probabilities. The error probability for the error under consideration is taken from a Monte-Carlo generator with mean equal to the average value of the respective error probability and 10% of the mean as the standard deviation. The fidelity with respective noise distribution is generated for a particular benchmark as the workload. Figure 5(a) shows the fidelity distribution for QFT - 3 benchmark. It is apparent that 2-qubit gate error causes the highest degradation in fidelity followed by 1-qubit gate error, T2 dephasing, and T1 relaxation. We simulated four other benchmarks, namely Hamming - 3, QFT - 5, Graycode - 6, and QFT - 7, each with different number of qubits and gates, to develop an insight and plotted the mean fidelity in Figure 5(b) (top) along with the number of gates (Figure 5(b) (bottom)). With increasing gate numbers, fidelity decreases since more gates increase error. In general, 2-qubit gate error contributes to the most loss in fidelity for all the benchmarks. However, in deeper benchmarks with more gates (e.g., QFT - 5 and QFT - 7) decoherence start to show more impact. This is because decoherence depends on time (and on qubit state) and deeper benchmarks take longer time of operation.

The insights from this analysis are: (i) reducing depth when possible will result in better decoherence related performance, and (ii) selecting qubits with better average gate fidelities (or, lower probability of error as calculated using expressions in Section 2.2.1) will reduce 1-qubit and 2-qubit gate-errors and thus, will improve fidelity of the circuit. The first one is discussed in Section 3.3 with contemporary methods from the literature. Later ones are discussed in Section 4 with the proposed QURE technique.

3 IMPLEMENTATION OF QUANTUM CIRCUITS ON HARDWARE

3.1 Hardware Agnostic Quantum Circuits

A quantum circuit is first written with a high-level description with more human-friendly gate sets (any unitary matrix can be considered to be a quantum gate). At this point, the hardware topology (connectivity/coupling, and available gate-set) of the target quantum computer is not considered. Such quantum circuit description is referred to as Hardware Agnostic Quantum Circuits (HAQC).

Figure 6: DOQA and VQA approaches for qubit allocation and associated sequence fidelities.

3.2 Circuit Compilation with Native Gate-Set

Every quantum computer supports a limited set of quantum operations (called native gate-set) which can be used to realize any arbitrary quantum gate described by unitary matrices. For instance, Rigetti 8Q-Agave native gate-set includes three unitary single qubit operation (RX , RZ , and I) and a single two-qubit operation (CZ). Therefore, the high-level $HAQC$ is first compiled to the native gate-set of a target hardware architecture considering the 2-qubit gate operation constraints as, for example in Rigetti devices, 2-qubit gates are permitted only between nearest neighbor qubits. The single qubit gates can be applied to any qubit of the computing hardware.

3.3 Qubit Allocation/Mapping

In this step, the logical qubits in the compiled circuit description are mapped to the physical hardware. A perfect initial mapping to satisfy all two-qubit gate dependencies is unlikely for most of the circuits. Hence, the initial mapping has to be modified during execution. SWAP based approach is considered to satisfy these dependencies [13, 18]. When two qubits are not neighbors on the quantum device based on the current mapping, they are moved together using SWAPs until a direct coupling exists between them, and then the two-qubit gate is applied. Previous efforts have been concentrated on minimizing the depth of the circuit during the mapping/qubit allocation procedure to minimize the effect of decoherence noise [13, 18, 22] which we term as depth optimal qubit allocation (DOQA). However, these solutions do not consider the variability among the fidelity of the individual gate operations.

During qubit allocation, multiple options may exist to satisfy a 2-qubit gate dependency. A recent work [20] proposes Variation-aware Qubit Allocation (VQA) policy based on the two-qubit coupling strengths (two-qubit gate fidelity) between the qubit pair in the options. It is concluded that choosing a higher depth (longer path) execution over a shorter depth one may result in higher success probability if the higher depth involves better 2-qubit gate fidelities than the shorter path. Such an approach may indeed increase the overall sequence fidelity. However, [20] did not consider decoherence error. Choosing a longer path among several possible alternatives, eventually means that the circuit operation time will increase. During this additional time, all the qubits will decohere (depending on the state). Therefore, a 2-qubit coupling-strength variation-aware policy may minimize coupling strength induced errors in exchange for added decoherence induced error. Thus, the overall fidelity may decrease instead of increasing. Next we present this conflicting scenario with a test-case and then, advocate for a depth-optimal allocation.

DOQA vs VQA: A 3-qubit hypothetical quantum computer coupling graph is shown in Figure 6(a) with the performance metrics shown in 6(b). The values are chosen based on the similar performance metrics of IBMQX4 quantum computer [4]. Two-qubit CNOT gates are allowed in both directions between (Q0,Q1), and (Q1,Q2) for the quantum computer (QC), and input states are taken to be superposition state ($(|0\rangle + |1\rangle)/\sqrt{2}$). A hypothetical two-qubit workload (CNOT Q0 Q2) is also shown in Figure 6(a). As there is no direct coupling available between Q0 and Q2, a qubit allocation algorithm compiles the workload based on the available gate operations. Two possible circuit compilation for the given workload is shown in Figure 6(c). The VQA proposed in [20], would choose C1 as it ensures the maximum sequence fidelity among the two circuits by only considering the two-qubit errors. However, a comprehensive simulation considering both the gate errors and decoherence induced errors using our simulation framework shows that the output

sequence fidelity will be greater for the shallow depth circuit (C2). The difference between the fidelities with/without decoherence is not significant for the simple test circuit. However, decoherence will affect a deeper circuits more in a realistic and larger workload. Therefore, we opt for the depth-optimal-qubit-allocation to apply QURE strategy (described next).

4 QUBIT RE-ALLOCATION (QURE)

By choosing a depth-optimal implementation, the decoherence induced error is addressed and optimized. Next, the QURE technique optimizes gate-errors. The circuit description after resolving all the multi-qubit gate dependencies (which we call the initial mapping - IM) can be directly implemented in the target quantum computing platform. The IM itself is a sub-graph to the target hardware coupling graph. Multiple sub-graphs exist in the target hardware coupling graph which is isomorphic to the IM . The circuit can be mapped to any of those sub-graphs without inserting any additional SWAP gates. For instance, the workload shown in Figure 1(b) can be mapped into 6 different ways in IBMQX4 [4] as shown in Figure 1(c). It is apparent from Figure 1(f) that a global maxima exists for a given circuit description among all possible mappings. Hence, there is scope to maximize the sequence fidelity of a compiled, depth-optimal, and swap inserted netlist by re-allocating the qubits to a different isomorphic sub-graph using QURE. Two different QURE approaches are discussed in the following Sections.

4.1 QURE: Sub-graph Search

This QURE approach requires solving two problems - (i) finding all the isomorphic sub-graphs (ISG) to the IM , and (ii) finding the sub-graph that offer maximum (or greater than an acceptable minimum) output fidelity for a given workload. This approach can be applied to any type of workload and we term it as *appNISQ_Universal*.

4.1.1 Finding Isomorphic Sub-graphs. A quantum computer can be represented as an undirected coupling graph where each node represents a qubit and an edge between two qubits indicates that multi-qubit gate operation is possible between these two qubits in any direction (2-qubit gates are allowed in both direction between two coupled qubits for newer quantum computing devices such as 20-Qubit chips from IBM [20]). There exists many sub-graphs in the coupling graph which are isomorphic to IM . However, finding all the ISGs is not straight-forward. This sub-graph isomorphism problem has been extensively studied in the past and is believed to be NP-complete [6].

The quantum hardware developed so far shows some regularities in their coupling graph. They generally follow a grid-like architecture [4] (similar to the 9Q-Square, Figure 3(b)) where the internal nodes are equivalent to each other in terms of the number of neighbouring qubits they are coupled to. This particular observation can reduce the time and space complexity of finding a reduced set of ISGs of an IM in a coupling graph. We describe the steps with an example:

Step-1: After getting the IM from a DOQA algorithm as shown in Figure 7(b) [22], we extract a rectangular grid architecture from the coupling graph which is enough to implement the given workload. We call this V-Grid. We rotate V-Grid by $\pi/2$ and call this H-Grid. If the extracted grid is a square, then V-Grid and H-Grid denote the same grid architecture. Inside each V-Grid and H-Grid, the given workload can be mapped in at least 4 different ways as shown in Figure 7(d). These are basically the horizontal and vertical mirrored re-assignment of the qubits from IM within the V-Grid/H-Grid.

Step-2: We determine the uniquely fitted V-Grids/H-Grids within the QC. Let the number of qubits in the horizontal and vertical

Figure 7: Finding ISG's for a given workload and target quantum hardware.

Figure 8: Normalized sequence fidelities for selected benchmarks for all possible mappings on target 9Q-Square hardware based on comprehensive simulation ((a)-(e)) and approximation ((f)-(j)). Each bar represent a qubit set e.g., first bar in (a) and (f) both represent qubit set {0, 1, 2}. Scaled (1/10x) error probabilities are used in both comprehensive simulation and approximation.

direction are QH , and QV for the QC , VQH and VQV for the V -Grid, HQH and HQV for the H -Grid. Then the number of uniquely fitted V -Grid (NVG) and H -Grid (NHG) within the QC will be $(QH-VQH+1)(QV-VQV+1)$ and $(QH-HQH+1)(QV-HQV+1)$ respectively.

Step-3: We compute all possible mapping coordinates for the qubits within each unique V -Grid/ H -Grid. The total number of unique mapping ($NISG$) in the target hardware for the given workload will be at least $4^*(NVG+NHG)$ for $VQV, HQV, HQH, HQV > 1$ and $VQH \neq HQH$. For $VQH = HQH$ (square and similar V -Grid and H -Grid), and $VQV, HQV, HQH, HQV > 1$, $NISG$ will be at least $4^*(NVG)$. If any of the four variables VQV, HQV, HQH , and HQV is equal to 1, it indicates a completely linear nearest neighbor (LNN) synthesis and mapping of the workload. For such workloads, the number of possible mapping is at most $(^N C_W)$ for a fully connected architecture (i.e., coupling exists between any two qubits) where N is the number of physical qubits in the hardware and W is the required number of logical qubits. For such cases, a straightforward heuristic re-allocation technique can be applied which is discussed at a later stage.

We acknowledge that the true number of possible ISGs can be much larger than the value we might get through the above steps. An exact list can be determined by using a more sophisticated algorithm [3]. However, the above approach can be useful as a tool to generate a reduced search space for an optimization algorithm whose goal is to find a qubit placement that tries to find an approximate fidelity maxima of the given workload in a target hardware.

4.1.2 Finding Approximate Global Maxima. The brute force approach is to find out the sequence fidelity for each individual mapping from $NISG$ and choosing the mapping with highest fidelity. Even though such an approach is not scalable, this can be useful for the $NISQ$ devices. For instance, in a hypothetical 256-qubit quantum computer with 16×16 grid architecture similar to the 16-qubit architecture shown in Figure 7(b), there will be $4^*(16-2+1)(16-2+1)$ or 900 possible mapping for a four qubit workload that has been mapped into a 2×2 sub-grid of the 16×16 architecture using a $DOQA$ algorithm.

To pick the *best* ISG among all the possible combinations, we employ a simpler model based on gate-error rates and rules of probability. It is to be noted, the simpler (and faster) model is used in decision making for qubit re-allocation to best ISG and the model developed in Section 2.2, which is more close to real device characteristics, is used to validate the decision and the simpler model. In real practice, the decision from the simpler model can be directly implemented in a real device.

We have mentioned earlier that the fidelity of any given workload (with total operation time significantly less than the coherence time) is limited by the two-qubit and single-qubit gate operation fidelity. The success probability of a workload for a given mapping can be approximated using the equation $SP = \prod_{i=1}^N (1 - \eta_i)$.

Here, SP denotes probability of success, N = total gate number, and η_i is the error rate of the i^{th} gate operation which has been calculated from probability of error (p in Section 2.2.1) using relations $\eta = 3p/4$ (1-qubit gate) and $\eta = 15p/16$ (2-qubit gate) [9]. The sequence fidelities for different benchmarks considering the one-qubit and the two-qubit gate errors using our simulation framework is shown in Figure 8(a)-(e) for all possible mappings in the target 9Q-Square QC. The sequence fidelities based on the above equation is also shown in Figure 8(f)-(j) for the same set of mappings for individual workloads. The fidelity plots for each of the benchmarks for the aforementioned approaches (comprehensive simulation vs approximation) produces peaks and valleys at exactly same locations (i.e., for the same qubits set). Therefore, the approximate equation can be used as a simplistic formula during QURE procedure. For larger quantum systems, techniques e.g., simulated annealing can be used to find a sub-graph from $NISG$ to maximize output fidelity.

4.2 QURE: Greedy Approach

In this approach, the logical qubits are allocated to physical qubits in a successive fashion and for each allocation step, the best performing available physical qubit is allocated for the selected logical qubit. It is particularly suitable for workloads which only require linearly

Figure 9: Greedy allocation of given LNN workload on target hardware.

coupled qubits for depth optimal mapping. We term this approach as *appNISQ_LNN*. In the beginning, we rank all physical qubits in a target hardware based on the average two-qubit gate fidelities of individual qubits (due to their high impact). The average two-qubit gate fidelities involving the selected qubit and its coupled neighbors and taking the mean. We also rank the logical qubits in the workload in terms of total number of two-qubit gate operations.

Let the ranking of the qubits in the 9Q-Square QC as follows: {Q5 Q0 Q3 Q7 Q8 Q4 Q1 Q6 Q2} in descending order (i.e., Q5 is best qubit) as shown in Figure 9(a). A hypothetical 4-qubit LNN workload {q0 q1 q2 q3} has the following ranking {q2 q1 q3 q0} in descending order (i.e., q2 involves highest number of 2-qubit operations) (Figure 9(b)). During the allocation process, we need to maintain the lists of *Assigned_Physical_Qubits*, and *Allocated_Logical_Qubits* with two ancillary lists *unallocated (logical) neighbors* and *unallocated physical neighbors*. Initially, these lists are empty. At the start of each iteration, we find the unallocated neighbors of the *Allocated_Logical_Qubits*. If *Allocated_Logical_Qubits* is empty, we select all the logical qubits in the workload (q0, q1, q2, q3) as the unallocated neighbors. We pick the logical qubit from the list of unallocated neighbors with the highest rank. We find the logical neighbors of the selected qubit (for LNN it can be at most 2) and check whether they are already allocated or not. If one of them is allocated, we find its unallocated physical neighbors (UPN), if both are allocated, we set UPN to their common unallocated physical neighbors, otherwise, we set UPN to all the currently unallocated physical qubits. We pick the physical qubit from UPN with the highest rank and assign it to the selected logical qubit. Then we add the selected logical qubit to *Allocated_Logical_Qubits* and the physical qubit to *Assigned_Physical_Qubits* before moving on to the next iteration. We stop when the length of *Allocated_Logical_Qubits* is equal to the number of logical qubits. A step-by-step procedure is demonstrated in Figure 9(c). The final mapping is shown in Figure 9(d).

5 SIMULATION RESULTS

We have simulated benchmarks from [21] and [2] with the modeled 9Q-square architecture with specs listed in Table 1 using the simulator described in Section 2.2. For the LNN workloads, we have used the *appNISQ_LNN* re-allocation procedure. For non-LNN workloads, we have used *appNISQ_Universal* to find the best possible mapping in terms of sequence fidelity without adding SWAP or any other gates. The results (Figure 10) indicate that QURE can improve the sequence fidelity by as much as 1.54x compared to a baseline allocation. For benchmarks with more qubits (e.g., QFT7), the improvement is the lowest since most of the available qubits (7 out of 9) are utilized leaving little scope for optimization.

6 CONCLUSIONS

In this paper, we present QURE to intelligently schedule gate operation to qubits with lower error probabilities and thus, resulting in better fidelity of the output state. We perform a comprehensive simulation with a model noisy quantum system and also perform an experiment in IBM's real quantum device to show that QURE can improve correct output probability or fidelity by a significant margin without incurring any physical or circuit level overhead.

Figure 10: Fidelity improvement with QURE (normalized with respect to fidelity without QURE). Simulation results are from the simulator described in Section 2.2.

Acknowledgement: This work is supported by SRC (2847.001), NSF (CNS- 1722557, CCF-1718474, CNS-1814710, DGE-1723687 and DGE-1821766) and DARPA Young Faculty Award (D15AP00089).

REFERENCES

- [1] J. I. Cirac and P. Zoller. 1995. Quantum Computations with Cold Trapped Ions. *Phys. Rev. Lett.* 74 (May 1995), 4091–4094. Issue 20.
- [2] Rigetti Computing. [n. d.]. Rigetti Forest SDK 2.0. <https://www.rigetti.com/>
- [3] Don Coppersmith and Shmuel Winograd. 1990. Matrix multiplication via arithmetic progressions. *Journal of symbolic computation* 9, 3 (1990), 251–280.
- [4] International Business Machines Corporation. [n. d.]. IBM Q Experience. <https://www.research.ibm.com/ibm-q/technology/devices/>
- [5] David P. DiVincenzo. 2000. The Physical Implementation of Quantum Computation. *Fortschritte der Physik* 48, 9 (2000), 771–783.
- [6] David Eppstein. 2002. Subgraph isomorphism in planar graphs and related problems. In *Graph Algorithms And Applications I*. World Scientific, 283–309.
- [7] Clare Horsman et al. 2012. Surface code quantum computing by lattice surgery. *New Journal of Physics* 14, 12 (2012), 123011.
- [8] J.R. Johansson et al. 2013. QuTiP 2: A Python framework for the dynamics of open quantum systems. *Computer Physics Communications* 184, 4 (2013), 1234 – 1240.
- [9] Yuval R Sanders et al. 2016. Bounding quantum gate error rate based on reported average fidelity. *New Journal of Physics* 18, 1 (2016), 012002.
- [10] Google. 2018. Google Bristlecone. <https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html>
- [11] E. Joos and H. D. Zeh. 1985. The emergence of classical properties through interaction with the environment. *Physik B Condensed Matter* 59, 2 (01 Jun 1985).
- [12] K. Kraus. 1983. States, Effects, and Operations: Fundamental Notions of Quantum Theory. *Springer-Verlag Berlin Heidelberg* (1983).
- [13] Gushu Li, Yufei Ding, and Yuan Xie. 2018. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. *arXiv preprint arXiv:1809.02573* (2018).
- [14] Michael A Nielsen and Isaac Chuang. 2002. Quantum computation and quantum information.
- [15] Jarryd J. et al. Pla. 2012. A single-atom electron spin qubit in silicon. *Nature* 489 (19 Sep 2012), 541 EP –.
- [16] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *arXiv preprint arXiv:1801.00862* (2018).
- [17] Matthew Reagor et al. 2018. Demonstration of universal parametric entangling gates on a multi-qubit lattice. *Science Advances* 4, 2 (2018).
- [18] Marcos Yukio Siraichi et al. 2018. Qubit allocation. In *Proceedings of the 2018 International Symposium on Code Generation and Optimization*. ACM, 113–125.
- [19] Matthias Steffen. 2011. Superconducting Qubits Are Getting Serious. <https://physics.aps.org/articles/pdf/10.1103/Physics.4.103>
- [20] S S Tannu and Moinuddin K Qureshi. 2018. A Case for Variability-Aware Policies for NISQ-Era Quantum Computers. *arXiv preprint arXiv:1805.10224* (2018).
- [21] Robert Wille et al. [n. d.]. RevLib. <http://www.revlib.org/>
- [22] Alwin Zulehner, Alexandru Paler, and Robert Wille. 2018. An efficient methodology for mapping quantum circuits to the IBM QX architectures. *IEEE TCAD* (2018).