QDiff: Differential Testing of Quantum Software Stacks

Jiyuan Wang, Qian Zhang, Harry Xu, and Miryung Kim

University of California, Los Angeles

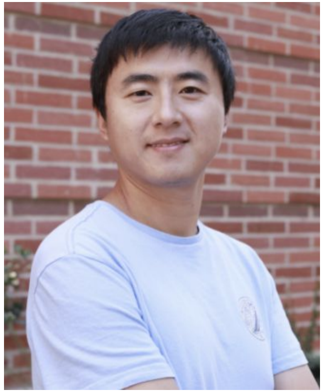
ASE ’21, November 17, 2021, Virtual

UCLA

TEAM MEMBERS

Jiyuan Wang

Qian Zhang



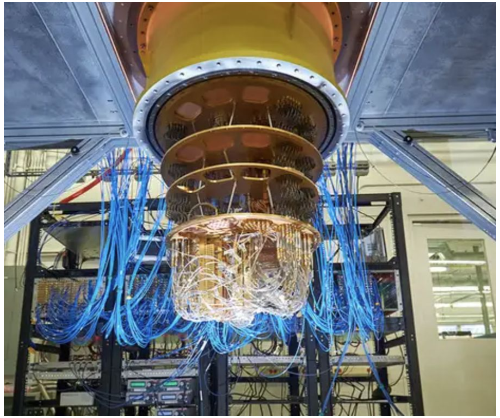
Harry Xu

Miryung Kim

People are using hardware accelerators

FPGA

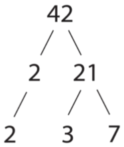
GPU



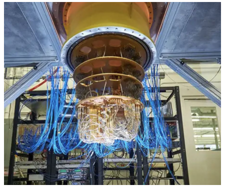
Quantum Computer

Quantum Computers (QCs) are promising hardware accelerators

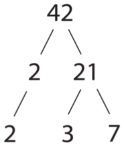

Classical: $O(2^n)$

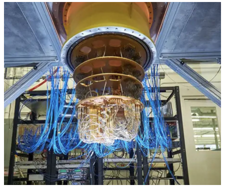


Classical: $O(e^{P(\log N)})$

Quantum: $O(2^{\sqrt{n}})$





Quantum: $O(P(\log N))$

Grover algorithm can speed up unstructured search with a quantum computer.

Shor algorithm can speed up integer factorization with a quantum computer.

Quantum platforms are becoming publicly available

Cirq

PYQUIL

Qiskit
Elements for building

Google

rigetti

IBM®

A study of real world errors in Qiskit, Cirq, and Pyquil

- We studied 76 bugs posted on **StackOverflow** and **Github** regarding Qiksit, Cirq, and Pyquil.
- Errors happen in three parts: language constructs, compilers, and simulators, which we call quantum software stacks (QSS).

Survey Statistics	
Keywords Searched	Quantum framework bugs, error, unexpected behavior
Posts Studied in total	76 posts
Common Fault Types	4

Error Types	Percentage	Example
Bugs in Language Constructs	**11.8%**	Qiskit crashes on cnot() gate
Bugs in Compiler Setting	**53.9%**	Qiskit has bugs at the compiler optimization level
Bugs in Different Simulators	**19.7%**	Pyquil crashes on PyQvm with controlled gates
Bugs in Installation / Interaction with Other Tools	**14.6%**	Qiskit installation fails with Python 3.9

Errors in QSS are hard to identify

Challenge 1: Inherent probabilistic nature of quantum computing makes the error detection hard.

Challenge 2: There are very few quantum programs out here.

Challenge 3: Wait time is long due to limited public access to expensive quantum hardware.

QDiff's solutions

Challenge 1: Inherent probabilistic nature of quantum computing makes the error detection hard.	Differential test with Kolmogorov-Smirnov (KS) test [1] based distribution comparison.
Challenge 2: There are very few quantum programs out here.	Generate more quantum programs with equivalent transformations.
Challenge 3: Wait time is long due to limited public access to expensive quantum hardware.	Avoid unnecessary invocations of a quantum simulator and hardware with static characteristics of quantum circuits.

1. Kolmogorov A. Sulla determinazione empirica di una legge di distribuzione. G. Inst. Ital. Attuari. 1933, **4**: 83.

QDiff approach overview

- QSS
- Seed Program
- Equivalent Programs
- Filtering
- Differential Execution
- Testing loop

Exploration of program variants, backends, and compiler settings

Filtering mechanism to avoid unnecessary hardware and simulator invocations

K-S statistics based comparison [1]

We found 4 software crashes in the **Cirq** and **Pyquil** simulations, and 2 possible root causes of 29 divergence cases on **IBM** quantum hardware.

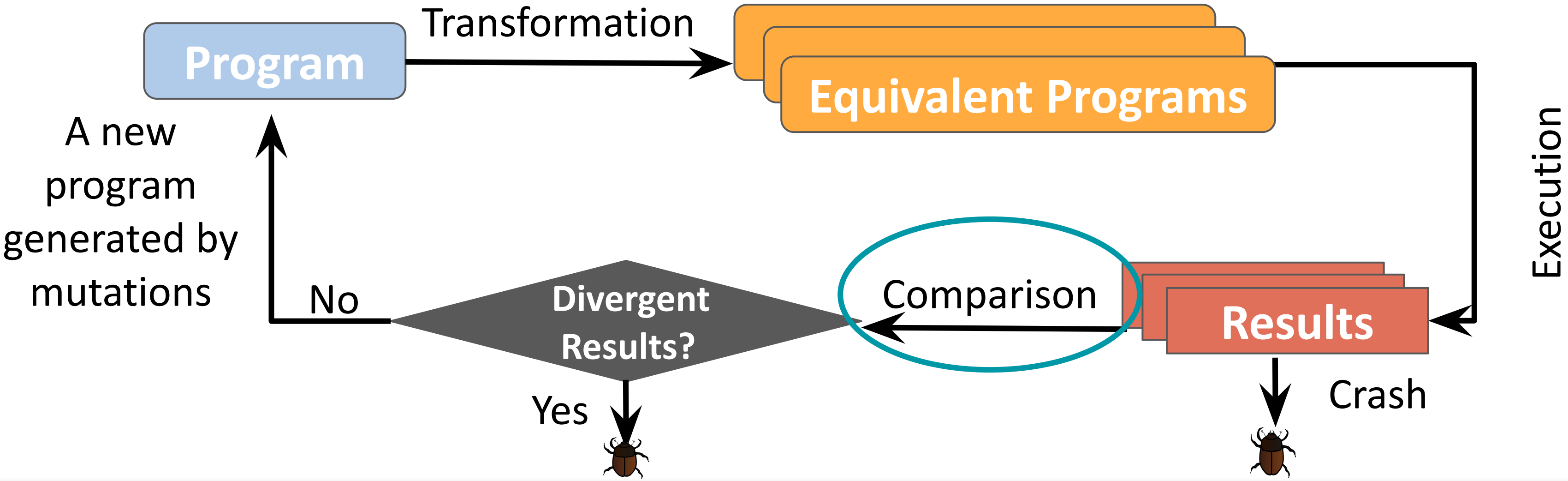
Approach 1: Equivalent program transformation

- QDiff transforms gates across their equivalent forms
  - We pick 7 commonly used rules in quantum compilation [2] and users can easily extend their own rules.
- QDiff also varies backend settings and compiler optimizations.

Rule ID	Original Gate	Equivalent Gate
G1	SWAP(p,q)	CNOT(p,q)CNOT(q,p)CNOT(p,q)
G2	S(p)	T(p)T(p)
G3	X(p)	H(p)S(p)S(p)H(p)
G4	Z(p)	S(p)S(p)
G5	CZ(p,q)	H(q)CX(p,q)H(q)
G6	CZ(p,q)CZ(p,q)	merged to Identity Matrix
G7	CCNOT(p,q)	6 CNOT gates with 9 one-qubit gates
…	…	…

2. K. Hietala, R. Rand, S.-H. Hung, X. Wu, and M. Hicks, “A verified optimizer for quantum circuits,”Proceedings of the ACM on Programming Languages, vol. 5, no. POPL, pp. 1–29, 2021.

Approach 2: Differential testing with distribution comparison



QDiff uses K-S test [1] to compare the distributions. We report the divergence if the K-S statistic is larger than a threshold.

Approach 2: KS based comparison

- **Cumulative Probability** with respect to the total trial number **N.**
- **K-S Statistic** is the max distance **D** between two measurement results.
- QDiff checks if **D** > a user-defined threshold **t.**
- The total trial number **N** can be determined by **t** and the confidence level **p,** which are given by users [3].

Measurement Distribution	'00'	'01'	'10'	'11'
N1	250	250	250	250
N2	247	247	253	253

Cumulative Probability	'00'	'01'	'10'	'11'
S1	0.250	0.500	0.750	1.00
S2	0.247	0.494	0.747	1.00

3. S.-O. Chan, I. Diakonikolas, P. Valiant, and G. Valiant, "Optimal algorithms for testing closeness of discrete distributions," in Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms.SIAM, 2014, pp. 1193–1203.

Approach 3: Filtering mechanism

- In the quantum theory, a qubit can retain a correct state for only a limited amount of time, called T1 time.
- 2-qubit gate dominates the error rate of the circuit.

Filtering mechanism: execution_time < T1 & #(2qgate) < th

Quantum circuits

Analyzing the number of 2-qubit gates, a circuit depth and execution time

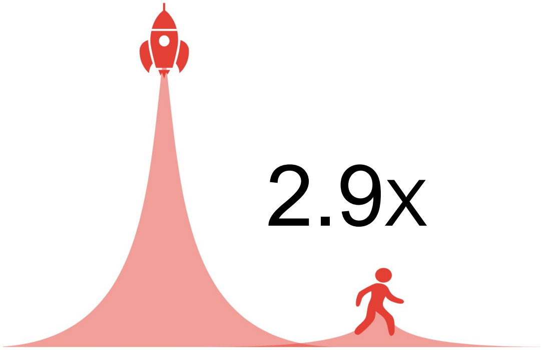
Executing on quantum hardware

Evaluation: Generation, speed up, and divergence

730

14799

We generated 730 different quantum programs with semantic modifying mutations. The programs in turn produced 14799 circuits with equivalent transformations.



We reduced 66% unnecessary quantum hardware and simulator invocations.

We found 4 crashes in the quantum simulators and 2 root causes of 29 divergences with the quantum hardware.

Differential testing on quantum simulators

4 crashes we found in quantum simulators and they have been confirmed with QSS developers.

```
qvm = get_qc('3q-qvm')
try:
    qvm.run(Program())
except:
    print("empty")
…..
qvm.run(Program(H(1)))
```

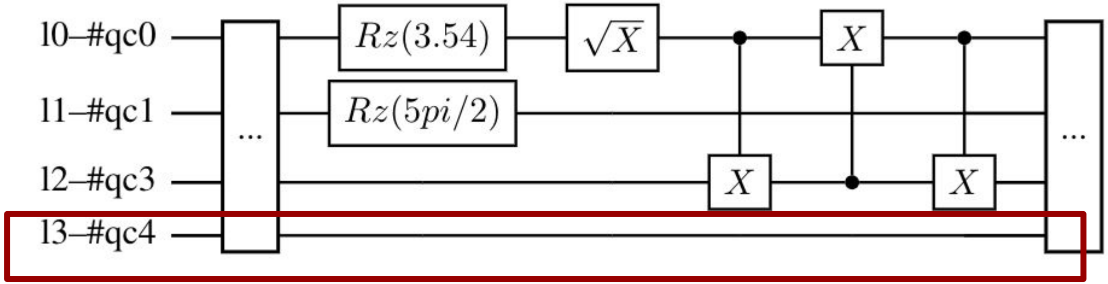
Example: Pyquil's simulator is stuck after executing an empty circuit https://github.com/rigetti/pyquil/issues/1034.

Differential testing on IBM quantum hardware

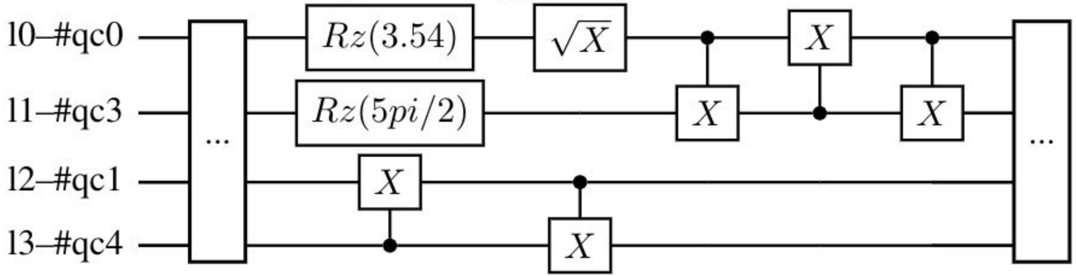
- We found 29 divergence cases in hardware executions:
  - These divergence cases might due to quantum circuit synthesis or compilation.
  - Equivalent programs produce measurements that are flagged as divergence in terms of K-S statistics.
- We speculated 2 possible root causes for 25 of them:
  - qubit dephasing [4]: 9 cases.
  - 2-qubit gate mapping [5]: 16 cases.

4. https://quantumai.google/cirq/google/best_practiceskeep_qubits_busy,2021
5. A. Hashim, R. K. Naik, A. Morvan, J.-L. Ville, B. Mitchell, J. M.Kreikebaum, M. Davis, E. Smith, C. Iancu, K. P. O'Brienet al.,"Randomized compiling for scalable quantum computing on a noisy superconducting quantum processor,"arXiv preprint arXiv:2010.00215,2020.

Differential testing on IBM quantum hardware



(a) "Divergent" circuits



(b) "Good" circuits

Qubit dephasing:
no operation on #qc4
for too long time [4].

4. https://quantumai.google/cirq/google/best_practiceskeep_qubits_busy,2021

QDiff: Differential Testing of Quantum Software Stacks

- We are the first one to adapt differential testing to quantum software stacks.
- QDiff is effective in
  - generating quantum program variants;
  - avoiding unnecessary invocations of the quantum simulator and hardware;
- We need deeper hardware knowledge to understand the root causes of our findings.
- We need further effort to isolate software stack errors from hardware noises.
- QDiff on Github: https://github.com/UCLA-SEAL/QDiff

Thanks for listening!

Q&A