Resource-Efficient Quantum Computing by Breaking Abstractions

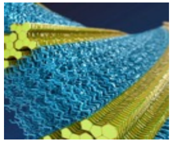
Fred Chong
Seymour Goodman Professor
Department of Computer Science
University of Chicago

Lead PI, the EPiQC Project, an NSF Expedition in Computing
NSF 1730449/1730082/1729369/1832377/1730088
NSF Phy-1818914/OMA-2016136
DOE DE-SC0020289/0020331/QNEXT

With Ken Brown, Ike Chuang, Diana Franklin, Danielle Harlow,
Aram Harrow, Andrew Houck, John Reppy, David Schuster, Peter Shor

Why Quantum Computing?

- Fundamentally change what is computable
  - The only means to potentially scale computation exponentially with the number of devices
- Solve currently intractable problems in chemistry, simulation, and optimization
  - Could lead to new nanoscale materials, better photovoltaics, better nitrogen fixation, and more

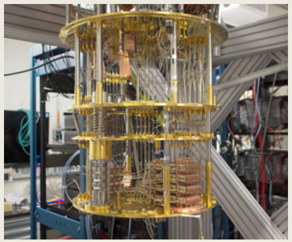


- A new industry and scaling curve to accelerate key applications
  - Not a full replacement for Moore's Law, but perhaps helps in key domains
- Lead to more insights in classical computing
  - Previous insights in chemistry, physics and cryptography
  - Challenge classical algorithms to compete w/ quantum algorithms

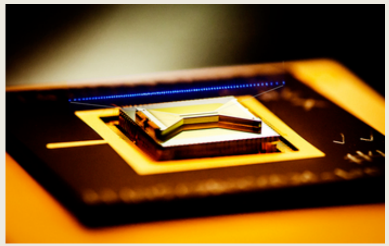
NISQ

Now is a privileged time in the history of science and technology, as we are witnessing the opening of the NISQ era (where NISQ = noisy intermediate-scale quantum).

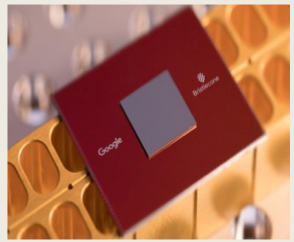
– John Preskill, Caltech



IBM
53 superconductor qubits



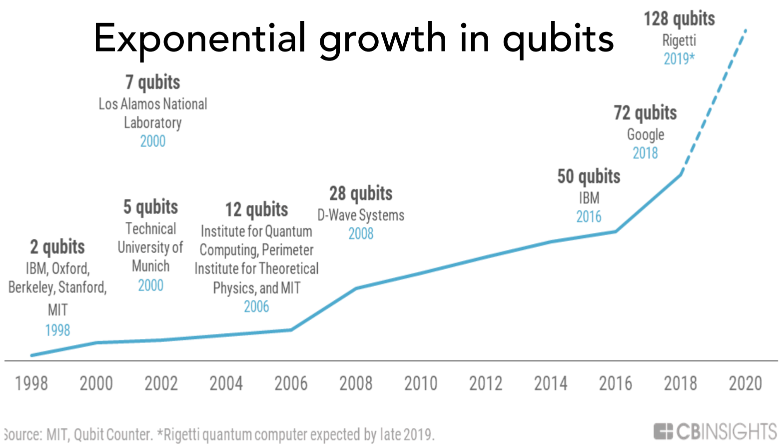
IonQ
79 atomic ion qubits
(11 controllable)



Google
53 supercond qubits

Quantum computing is at the cusp of a revolution

Every qubit doubles computational power



… led to quantum supremacy with 53 qubits

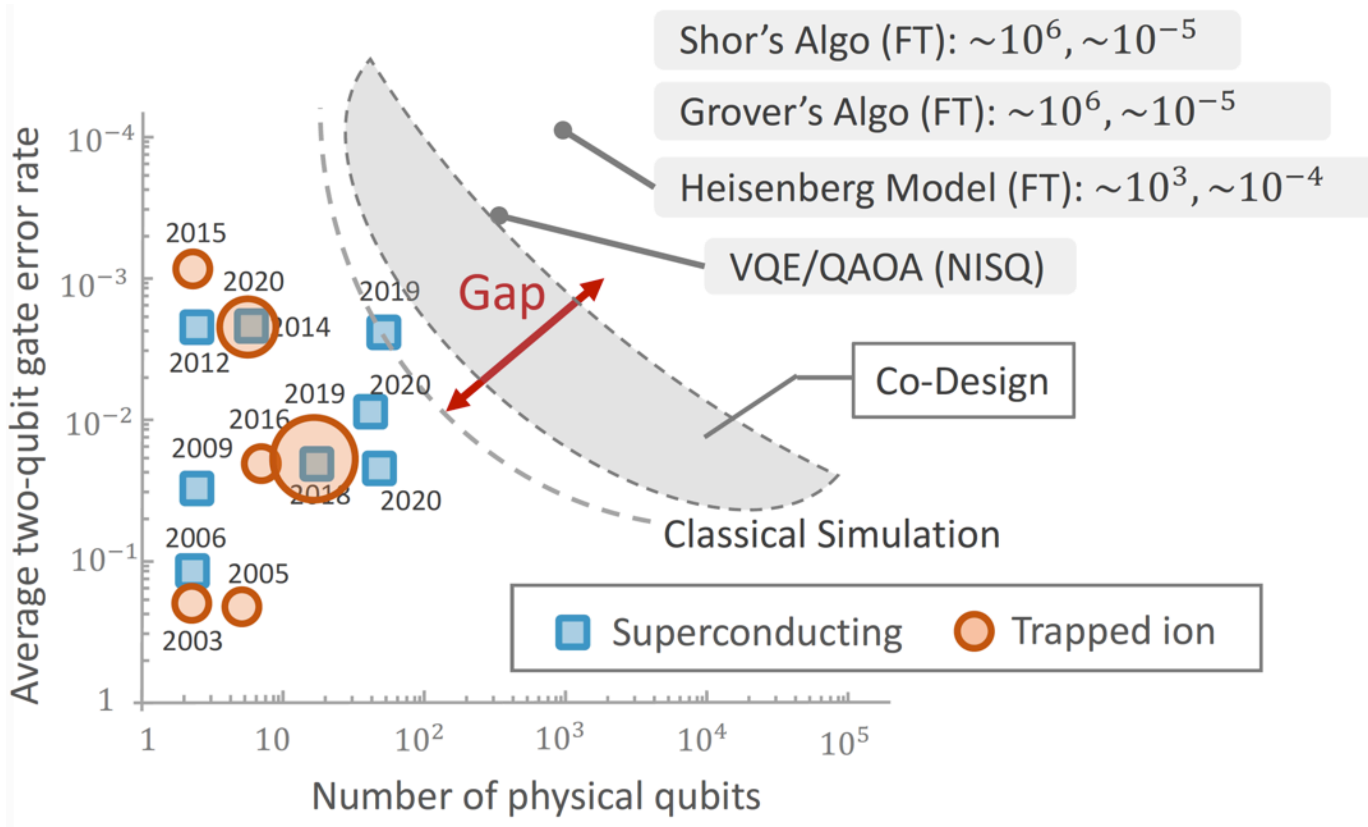
vs.

Seconds

Days

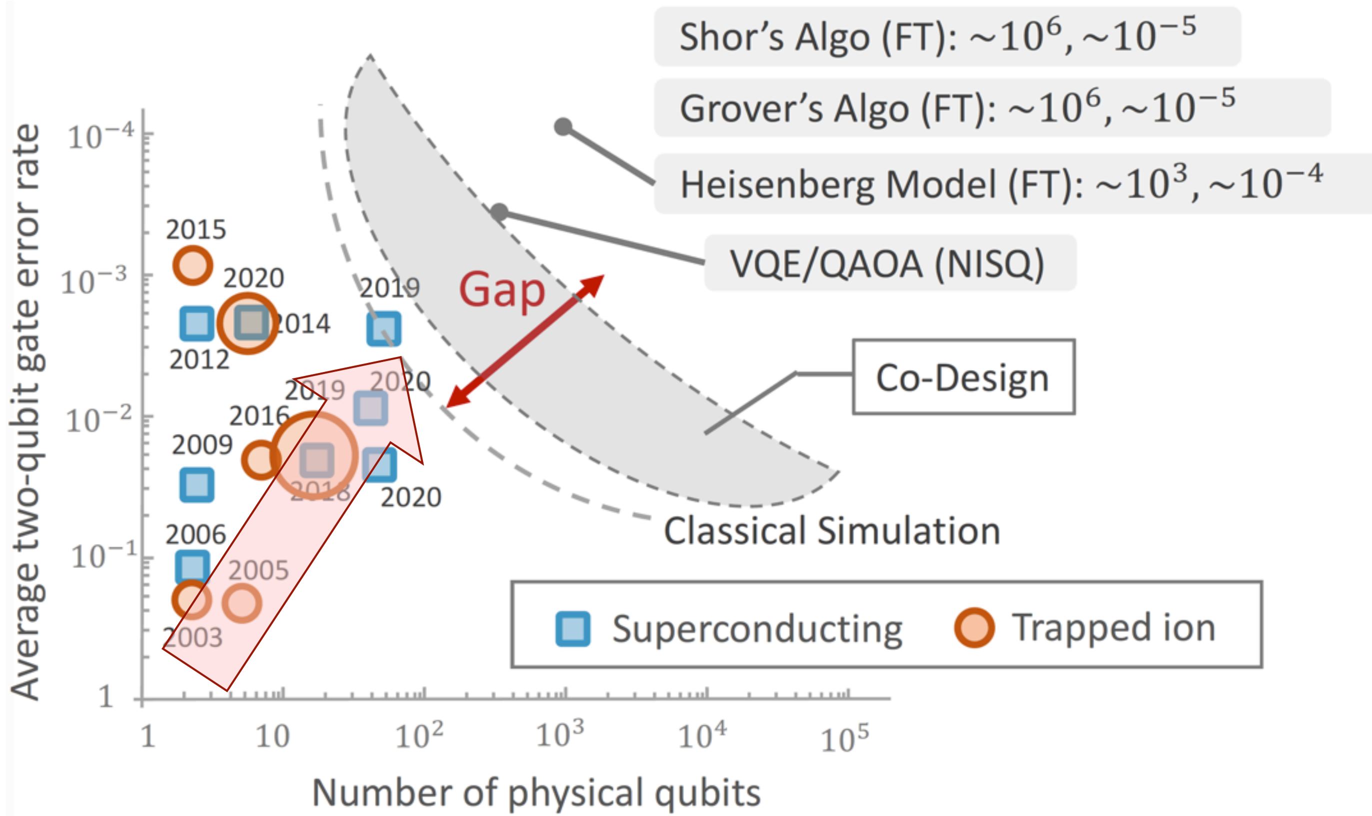
Double exponential growth!

The Gap between Algorithms and Hardware



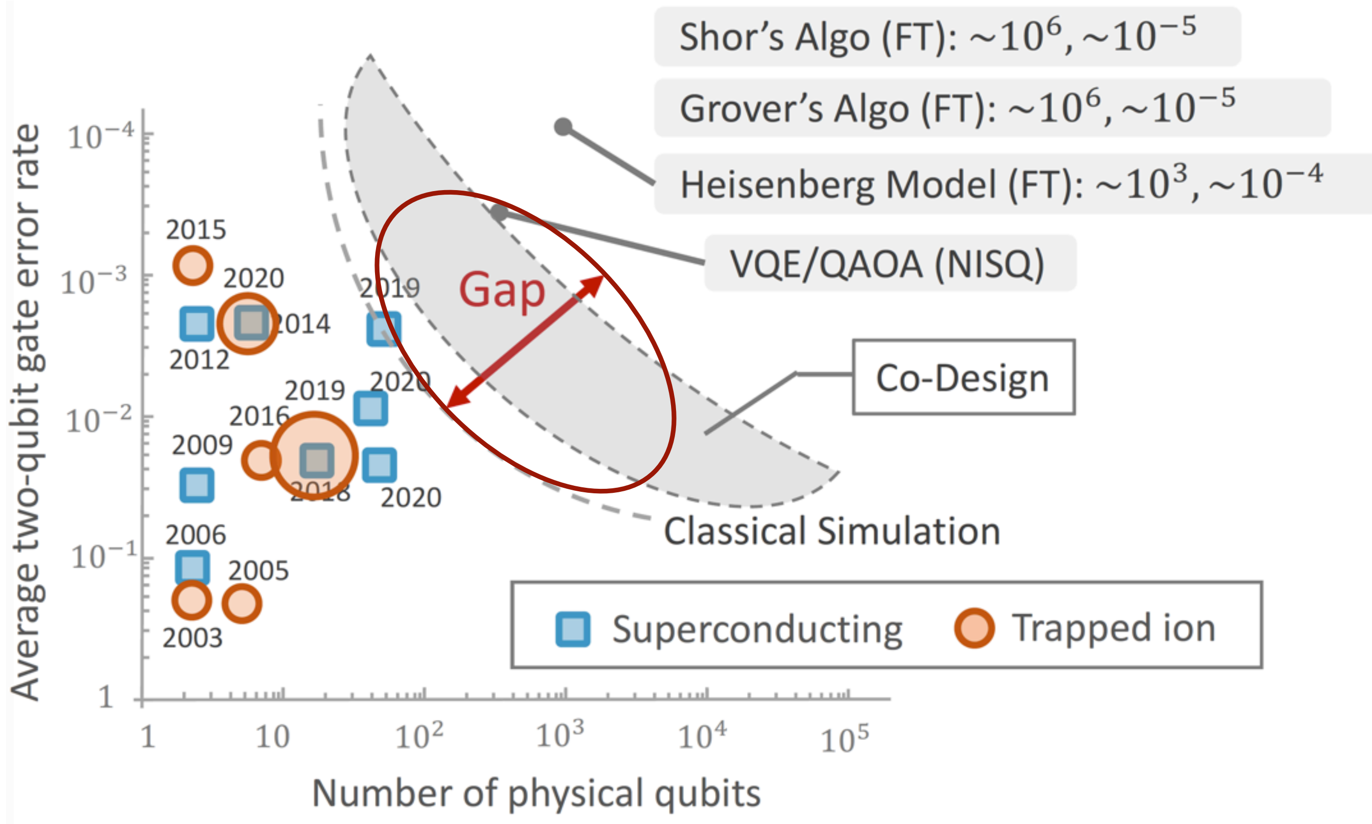
*Size of data point indicates connectivity; larger means denser connectivity.

The Gap between Algorithms and Hardware



*Size of data point indicates connectivity; larger means denser connectivity.

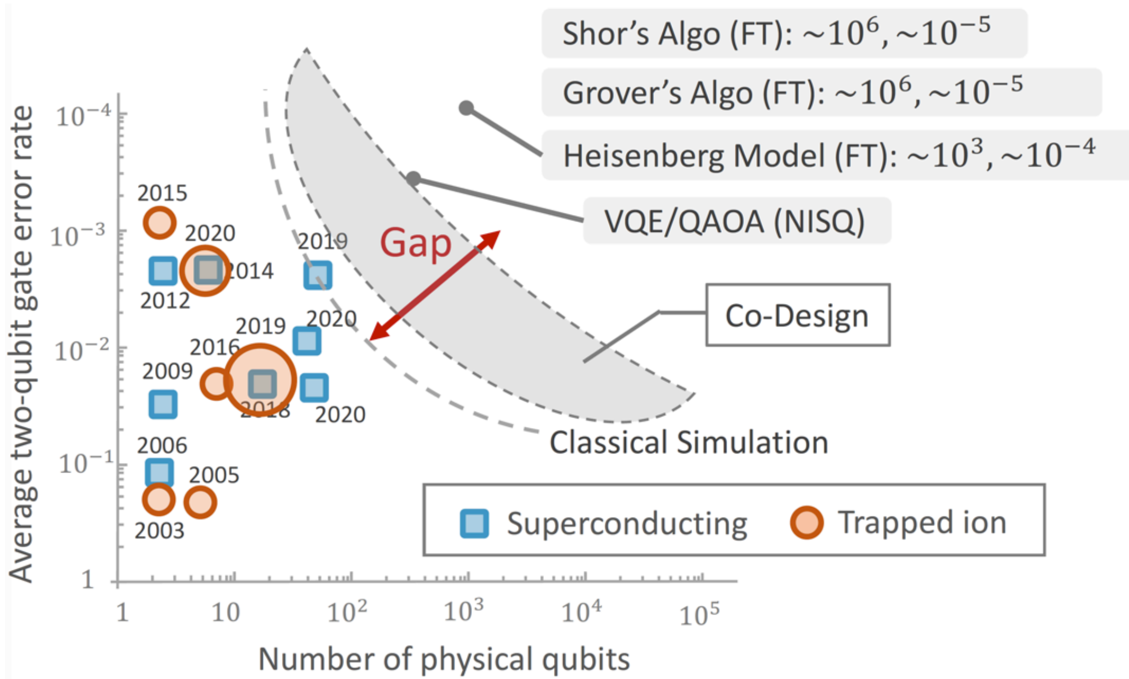
The Gap between Algorithms and Hardware



*Size of data point indicates connectivity; larger means denser connectivity.

The EPiQC Goal

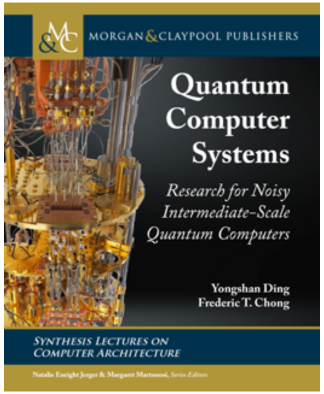
Develop algorithms, software, and hardware in concert to close the gap between algorithms and devices by 100-1000X, accelerating QC by 10-20 years.



*Size of data point indicates connectivity; larger means denser connectivity.

EPiQC Results

- In 2.5 years:
  - Many optimizations, each 2-10X, up to 10000X
  - 60+ papers, 5 best paper awards
  - 6 patents pending
  - 1 startup
  - 1 textbook (EdX courses forthcoming)
  - Techniques integrated into IBM QISKit and Google Cirq



Highlights

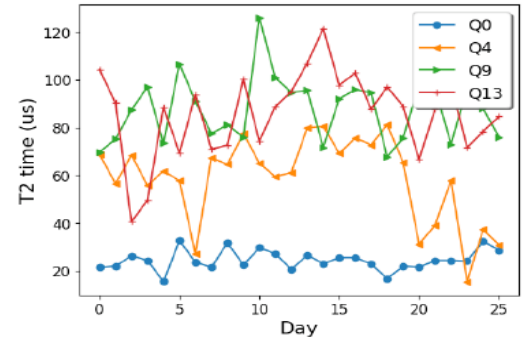
	Activity	Outcomes
Research	Minimizing measurements for quantum chemistry (**IBM Q Best Paper**, **QCE20 Best Paper**)	8-30X fewer measurements
	Direct-to-pulse compilation (ASPLOS19,MICRO19,MICRO20)	Up to 10X lower time
	Noise-adaptive mapping and scheduling (ASPLOS19, ISCA19, **Micro Top Pick 20**)	Up to 28X reliability
	Qutrit circuits (**QIP19 best poster**, ISCA19, **Micro Top Pick 20,** ACM TQC20)	Up to 70X fewer devices
	Technology-Aware Error Correction (stabilizer slicing, PRL18)	90X increased reliability
	Scheduling for crosstalk mitigation (ASPLOS20)	5.6X reduced error
	Frequency assignment for crosstalk mitigation (MICRO20)	75X reduced error
	Qubit reuse with uncomputation (ISCA20)	1.5-9.6X reduced resources
	Superconducting hardware error decoder (ISCA20)	1000-10000X increased qubits*gates
	Shuttling-based, trapped-ion architecture (ISCA20)	1000-10000X reduced error
Education	Industry summit (FCRC19) and advisory board	Create a Quantum Computer Systems Discipline and a Workforce Pipeline.
	Open-source SW and Tutorials (100's participants, 1000's downloads, 1000's youtube views)	
	Quora Quantum Computing Session (**150K+ views**)	
	QIS K-12 Key Concepts and Q-12 Partnership	

Highlights

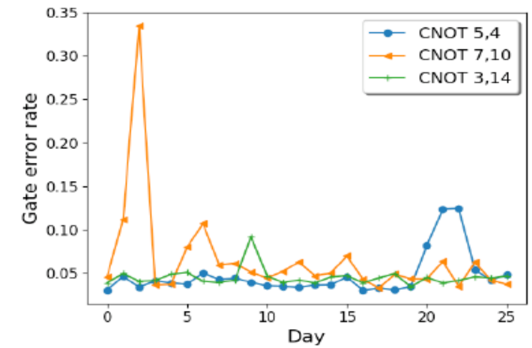
	Activity	Outcomes
Research	**Minimizing measurements for quantum chemistry (IBM Q Best Paper, QCE20 Best Paper)**	**8-30X fewer measurements**
	Direct-to-pulse compilation (ASPLOS19,MICRO19,MICRO20)	**Up to 10X lower time**
	Noise-adaptive mapping and scheduling (ASPLOS19, ISCA19, Micro Top Pick 20)	**Up to 28X reliability**
	Qutrit circuits (QIP19 best poster, ISCA19, Micro Top Pick 20, ACM TQC20)	**Up to 70X fewer devices**
	Technology-Aware Error Correction (stabilizer slicing, PRL18)	90X increased reliability
	Scheduling for crosstalk mitigation (ASPLOS20)	5.6X reduced error
	Frequency assignment for crosstalk mitigation (MICRO20)	**75X reduced error**
	Qubit reuse with uncomputation (ISCA20)	**1.5-9.6X reduced resources**
	Superconducting hardware error decoder (ISCA20)	1000-10000X increased qubits*gates
	Shuttling-based, trapped-ion architecture (ISCA20)	1000-10000X reduced error
Education	Industry summit (FCRC19) and advisory board	Create a Quantum Computer Systems Discipline and a Workforce Pipeline.
	Open-source SW and Tutorials (100's participants, 1000's downloads, 1000's youtube views)	
	Quora Quantum Computing Session (**150K+ views**)	
	QIS K-12 Key Concepts and Q-12 Partnership	

1. Software can Adjust to Hardware

- Quantum hardware varies day to day
- IBM publishes calibration data for their machines every day



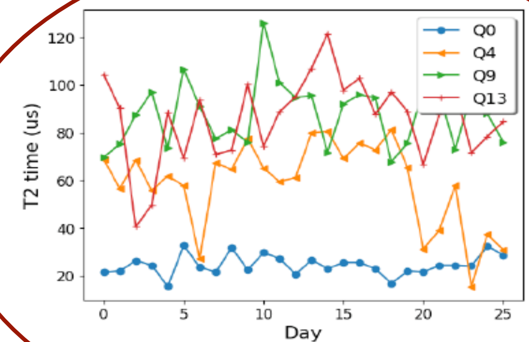
(a) Coherence time (T2)



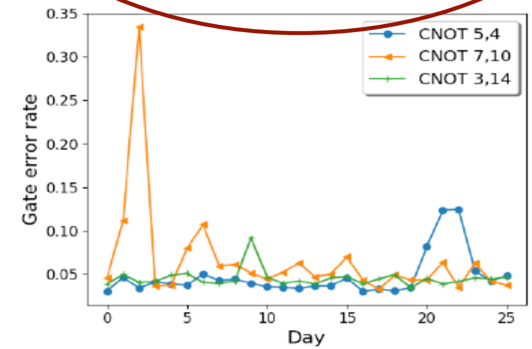
(b) CNOT gate error rate

1. Software can Adjust to Hardware

Quantum Bits vary in quality day to day



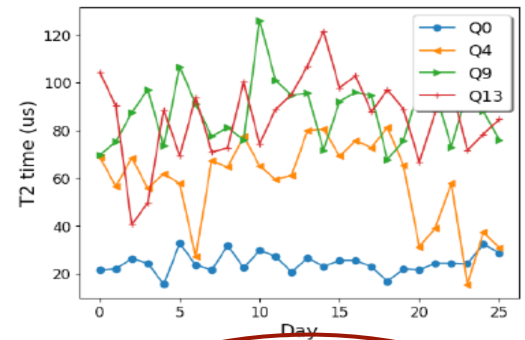
(a) Coherence time (T2)



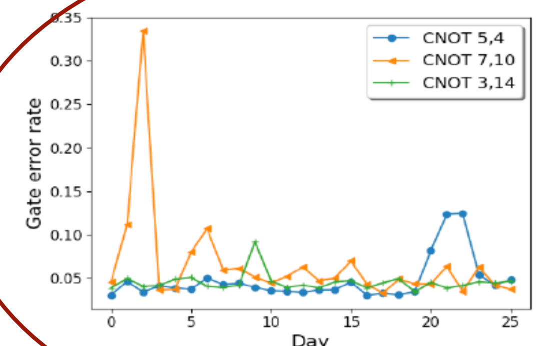
(b) CNOT gate error rate

1. Software can Adjust to Hardware

Operations between different pairs of quantum bits vary in quality day to day



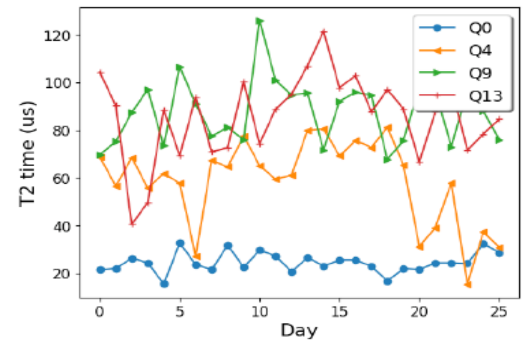
(a) Coherence time (T2)



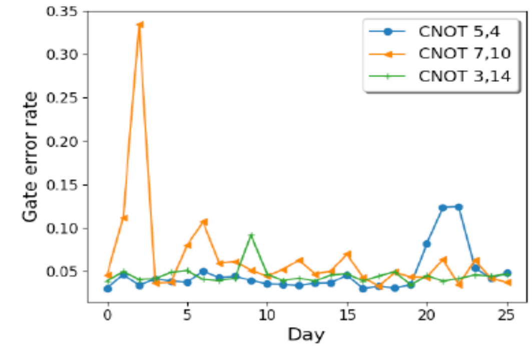
(b) CNOT gate error rate

1. Software can Adjust to Hardware

- Quantum hardware varies day to day
- *Software can target a specific machine for a specific day*

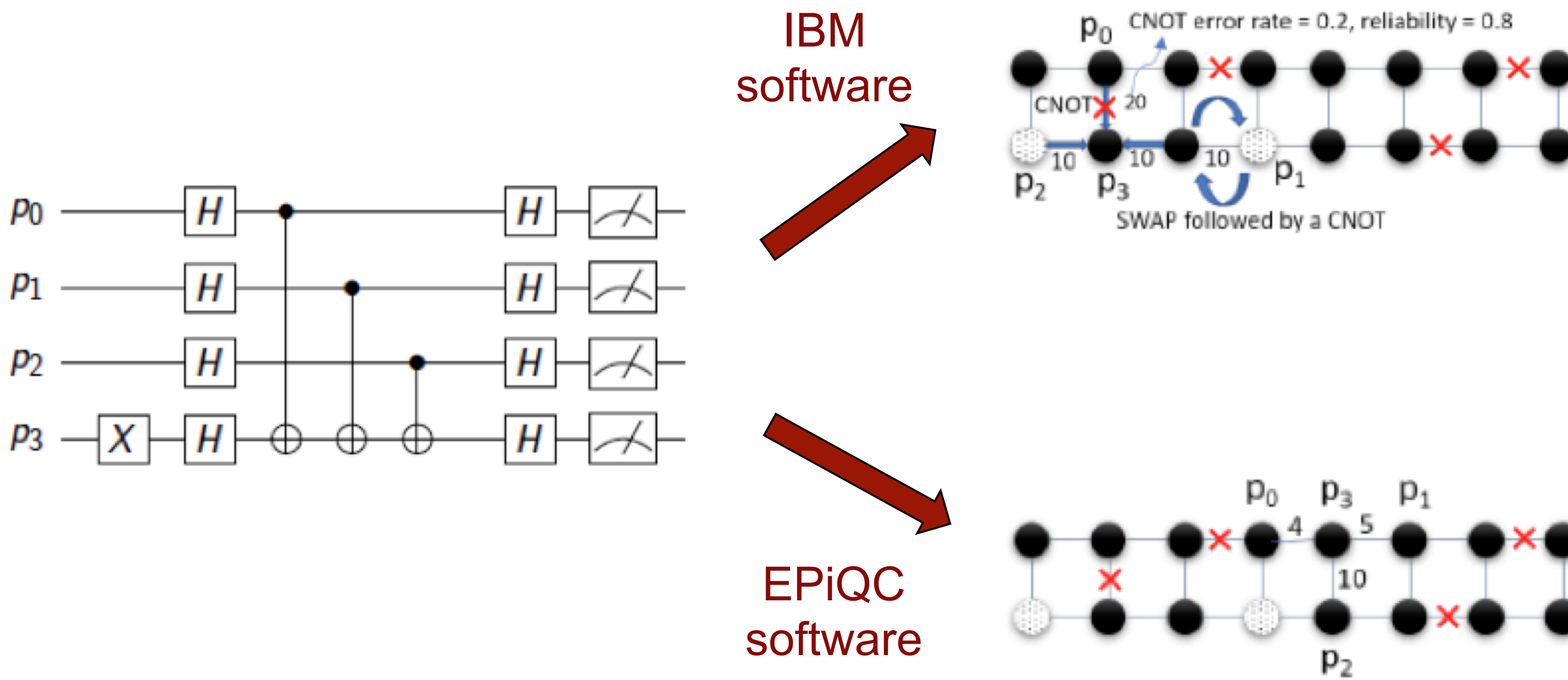


(a) Coherence time (T2)



(b) CNOT gate error rate

Avoiding Bad Hardware

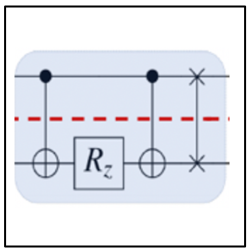


The Result

- Up to 28X better reliability (2.9X mean)
- Now integrated into IBM QISKit
- Changed how quantum programs are compiled
  - Won “Top Picks” best papers for 2019 award
- The key was to break the compile-once model

2. Direct-to-Pulse Compilation

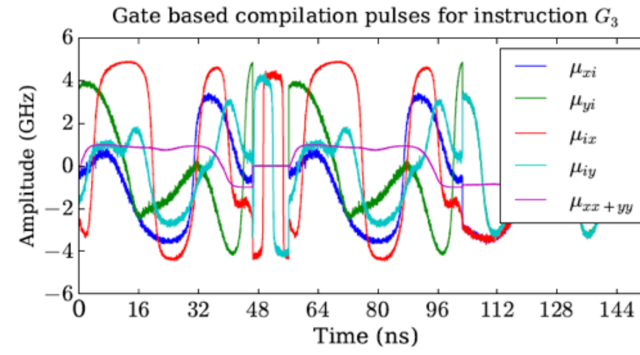
Quantum Function



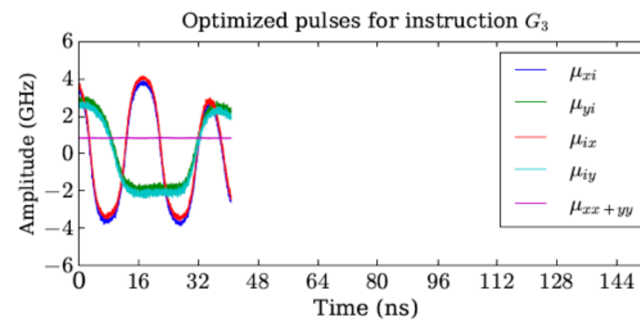
Quantum Assembly

CNOT q1, q2
RZ t1, q2
CNOT q1, q2
SWAP q1, q2

Direct to Pulse



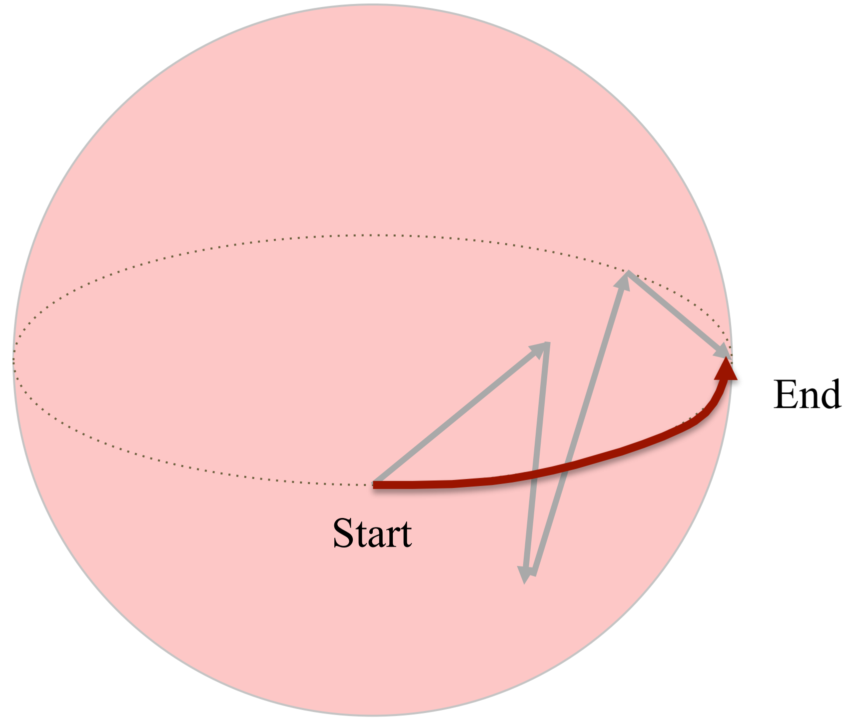
(c)



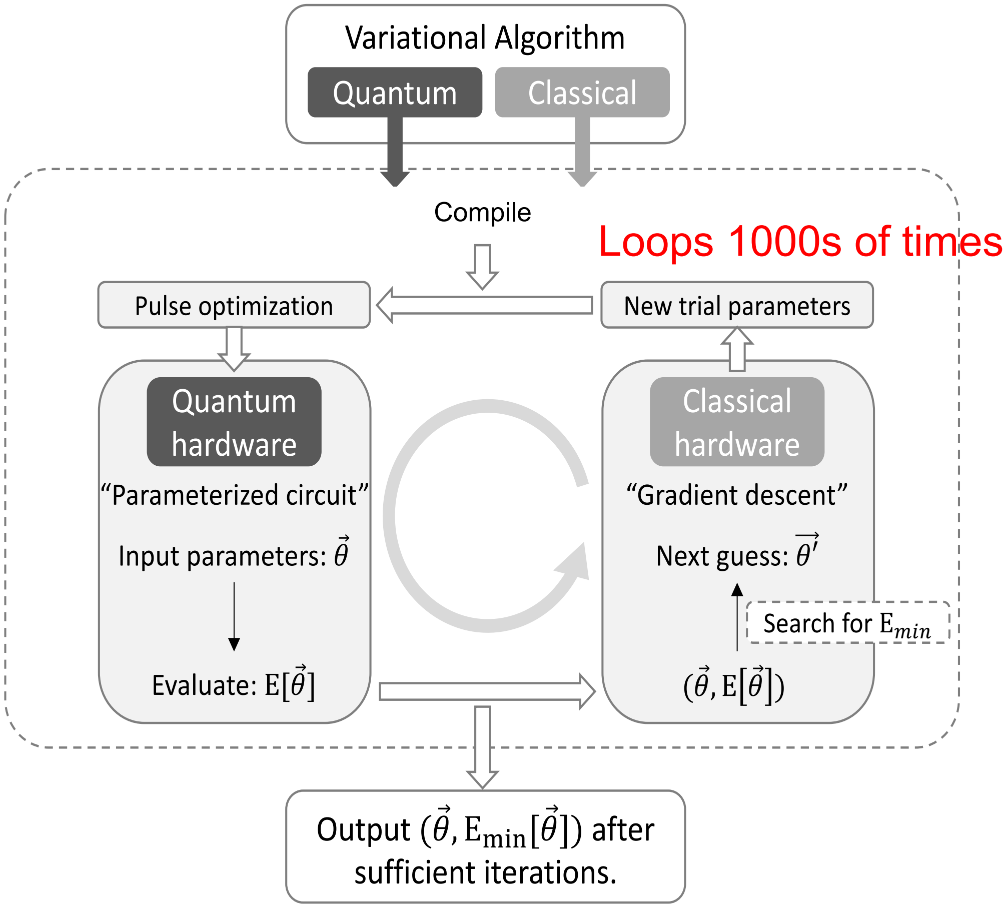
(d)

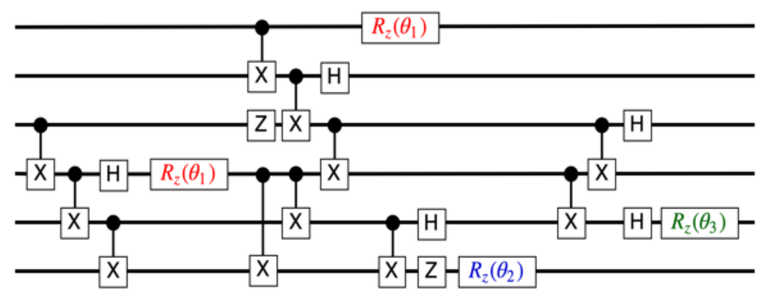
Direct-to-Pulse Results

- **2X to 10X faster**
- But it can take hours to compile a program before we can run it
- This is a problem for an important class of algorithms that alternates between classical and quantum computing

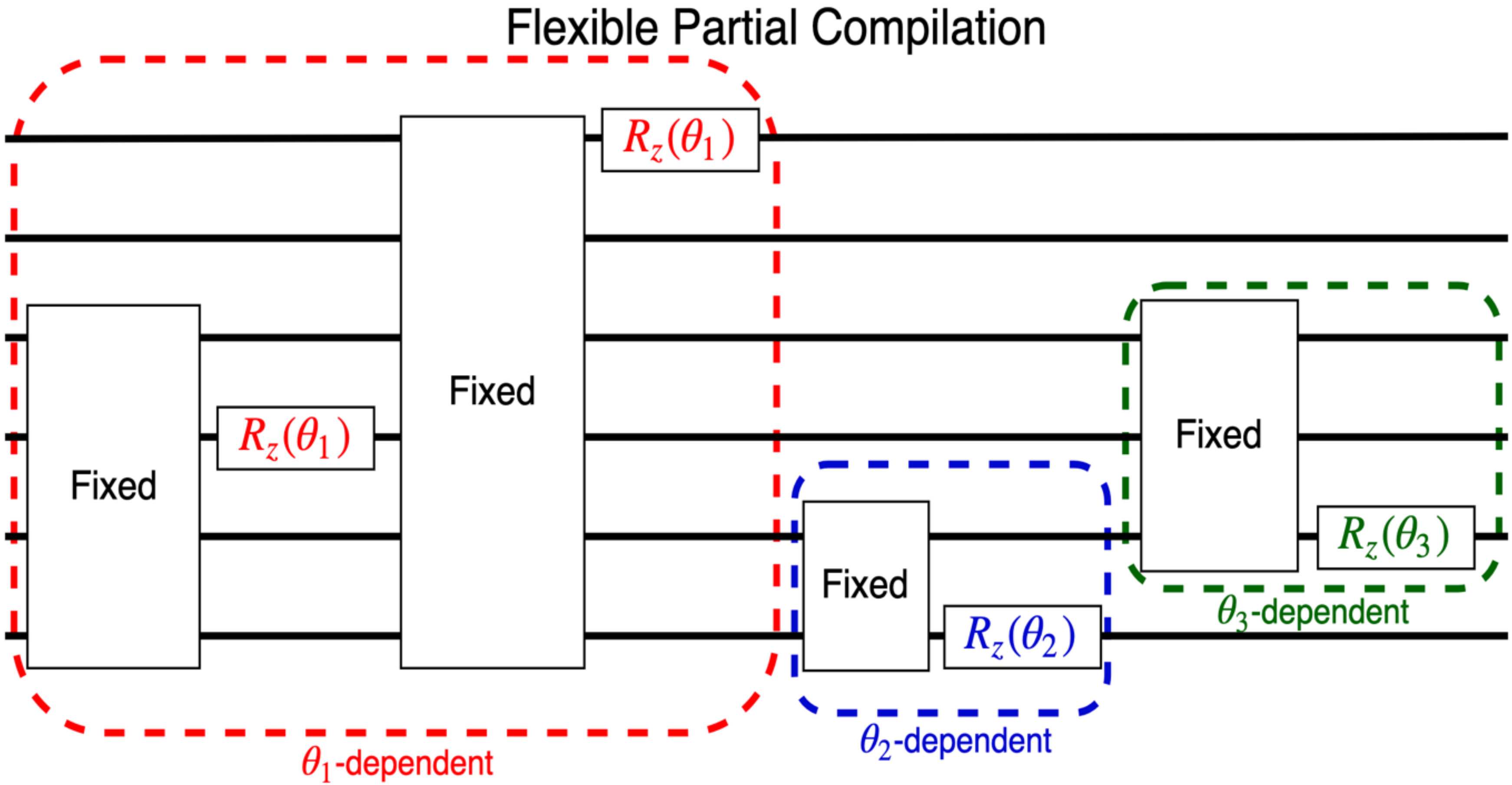


Variational Quantum Algorithms





Flexible Partial Compilation



Partial Compilation Results

- **2x** pulse speedups
- **10-80x** faster compilation than previous method
- **2 patents** pending
- *The key was to break the abstraction of machine instructions and target pulses*

3. Simultaneous Measurement


arXiv:1907.13623v1 [quant-ph] 31 Jul 2019

Minimizing State Preparations in Variational Quantum Eigensolver by Partitioning into Commuting Families

Pranav Gokhale*[1], Olivia Angiuli[2], Yongshan Ding[1], Kaiwen Gui[3], Teague Tomesh[4,5], Martin Suchara[3,4], Margaret Martonosi[5], and Frederic T. Chong[1]

[1]Department of Computer Science, University of Chicago
[2]Department of Statistics, University of California, Berkeley
[3]Pritzker School of Molecular Engineering, University of Chicago
[4]Argonne National Laboratory
[5]Department of Computer Science, Princeton University

August 1, 2019

Abstract

Variational quantum eigensolver (VQE) is a promising algorithm suitable for near-term quantum machines. VQE aims to approximate the lowest eigenvalue of an exponentially sized matrix in polynomial time. It minimizes quantum resource requirements both by co-processing with a classical processor and by structuring computation into many subproblems. Each quantum subproblem involves a separate state preparation terminated by the measurement of one Pauli string. However, the number of such Pauli strings scales as N^4 for typical problems of interest—a daunting growth rate that poses a serious limitation for emerging applications such as quantum computational chemistry. We introduce a systematic technique for minimizing requisite state preparations by exploiting the simultaneous measurability of partitions of commuting Pauli strings. Our work encompasses algorithms for efficiently approximating a MIN-COMMUTING-PARTITION, as well as a synthesis tool for compiling simultaneous measurement circuits. For representative problems, we achieve 8-30x reductions in state preparations, with minimal overhead in measurement circuit cost. We demonstrate experimental validation of our techniques by estimating the ground state energy of deuteron on an IBM Q 20-qubit machine. We also investigate the underlying statistics of simultaneous measurement and devise an adaptive strategy for mitigating harmful covariance terms.

1. Introduction

The present Noisy Intermediate-Scale Quantum (NISQ) era [1] is distinguished by the advent of quantum computers comprising tens of qubits, with hundreds of qubits expected in the next five years. Although several thousand logical error-corrected qubits, backed by millions of device-level physical qubits, are needed to realize the originally-envisioned quantum applications such as factoring [2] and database search [3], a new generation of *variational* algorithms have been recently introduced to match the constraints of NISQ hardware.

Variational Quantum Eigensolver (VQE) [4] is one such algorithm that is widely considered a top contender, if not the top contender, for demonstrating a useful quantum speedup. VQE is used to approximate the lowest eigenvalue of a matrix that is exponentially sized in the number of qubits. This is a very generic eigenvalue problem with a wide class of applications such as molecular ground state estimation [4]; maximum 3-satisfiability, market split, traveling salesperson [5]; and maximum cut [6]. In this paper, we focus on the molecular ground state estimation problem which has already been demonstrated experimentally, though we underscore that the full range of VQE applications is very broad.

VQE solves a similar problem as Quantum Phase Estimation (QPE) [7, 8], an older algorithm that requires large gate counts and long qubit coherence times that are untenable for near-term quantum computers. VQE mitigates these quantum resource requirements by shifting some computational burden to a classical co-processor. As a result, VQE achieves low gate count circuits and error resilience, but at the cost of requiring many iterations where each iteration measures one of $O(N^4)$ terms.

This is a daunting scaling factor that poses practical limitations. It was observed that this N^4 scaling could be partly mitigated by performing simultaneous measurement: when two terms correspond to commuting observables, they can be measured in a single state preparation. Our work starts from this observation and we seek to exploit this idea to minimize the total number of state preparations needed.

Our specific contributions include:

1. Efficient approximation algorithms for partitioning the N^4 terms into commuting families, i.e. approximating the MIN-COMMUTING-PARTITION.
2. A circuit synthesis tool for simultaneous measurement.
3. Statistical analysis of simultaneous measurement and a procedure for guarding against harmful covariance terms.
4. Validation of these techniques through benchmarks, simulations, and experiments.

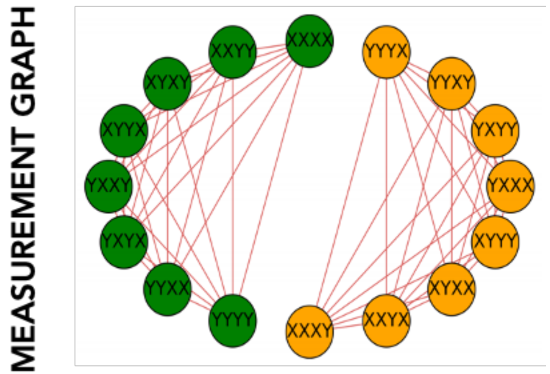
The rest of this paper is structured as follows. Section 2

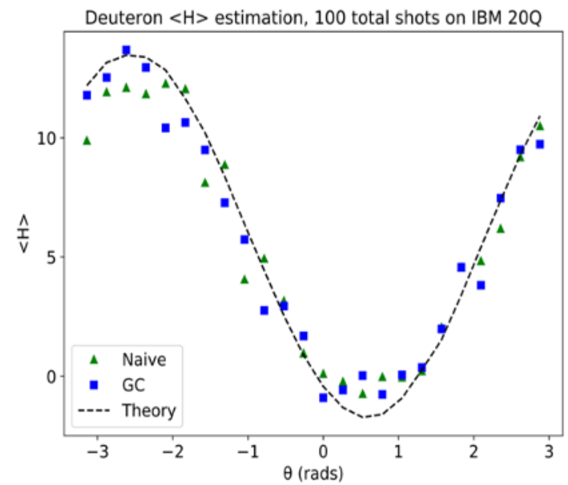
*Corresponding author: pranavgokhale@uchicago.edu

QCE 2020 Best Paper + IBM Q Best Paper

- Variational Quantum Eigensolver finds ground state energy of molecules.
- But the number of repeated measurements is enormous
- We restructure the algorithm to get 10X more out of each measurement

Simultaneous Measurement





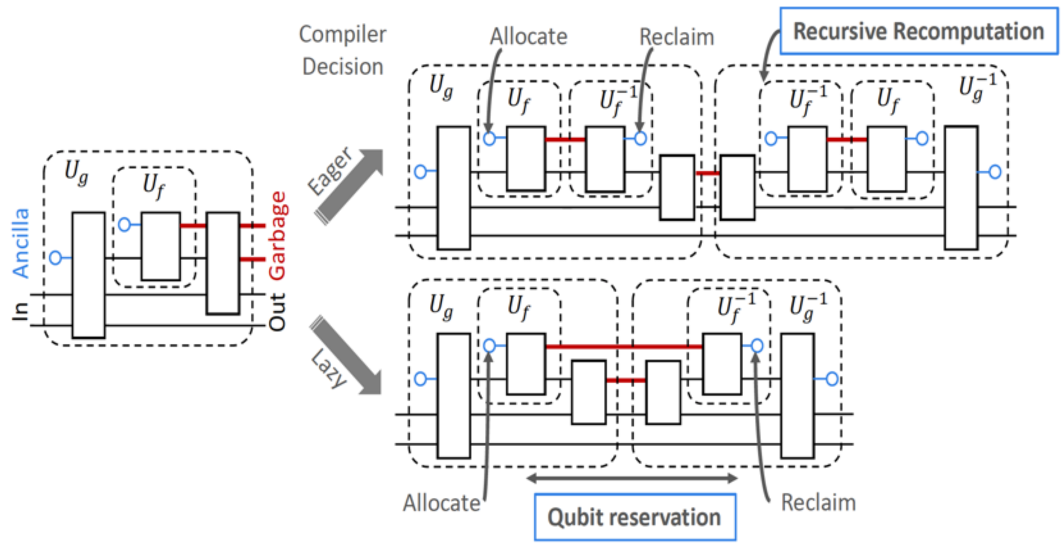
- Graph analysis to group terms to measure
- Deuteron on IBM shows less error even on a small problem
- Example of how algorithms can be restructured for greater efficiency
- Optimization algorithms are likely variational also

4. Reusing Quantum Bits

- Similar to classical memory management or garbage collection
- But in quantum programs, you must “uncompute” part of your work in order to reuse a quantum bit

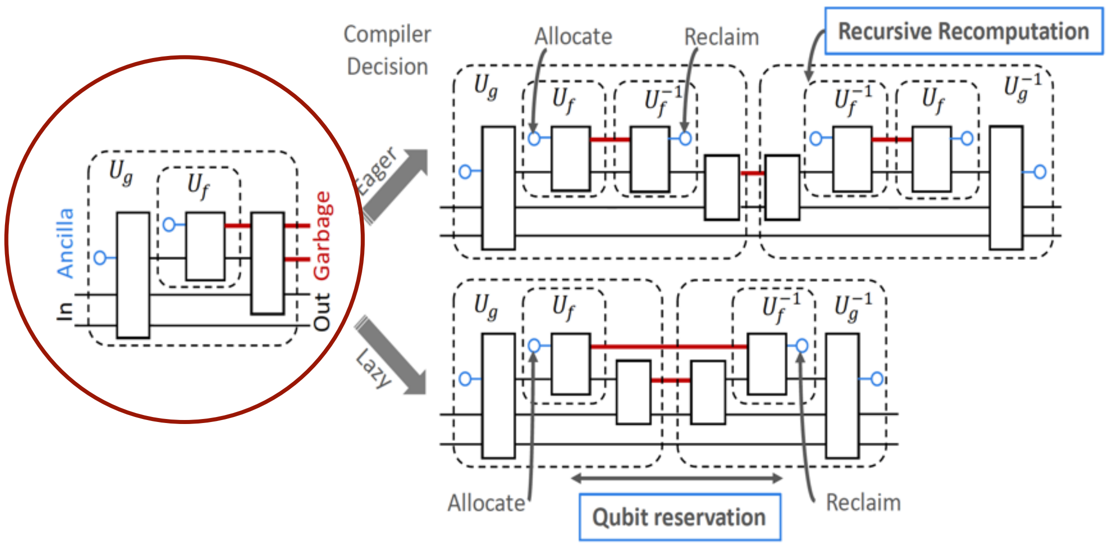
Uncomputation

- Mapping and scheduling for uncomputation and reuse
- Dealing with nested functions



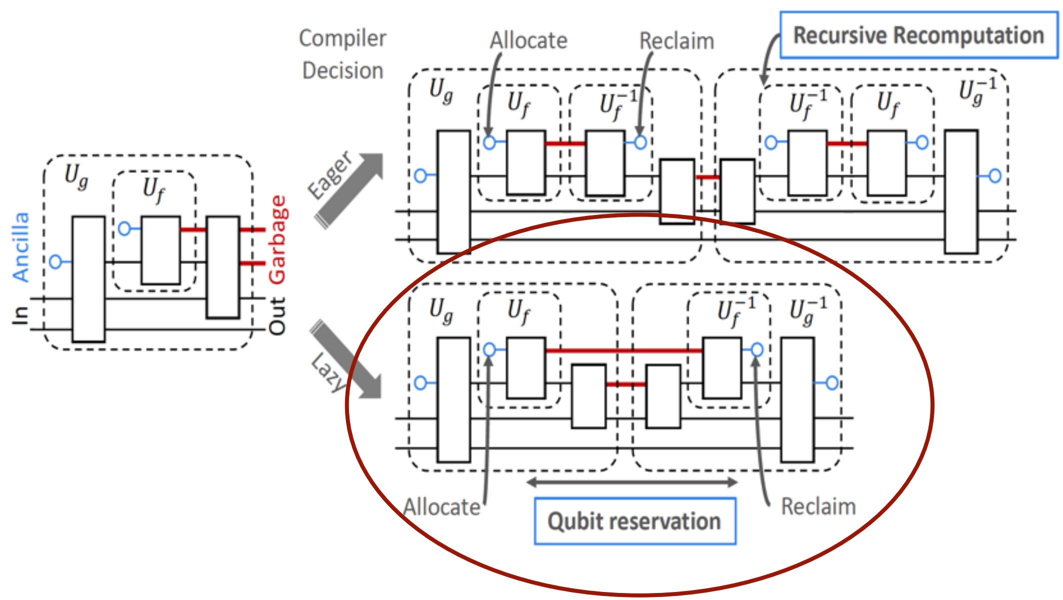
Uncomputation

- Mapping and scheduling for uncomputation and reuse
- Dealing with nested functions



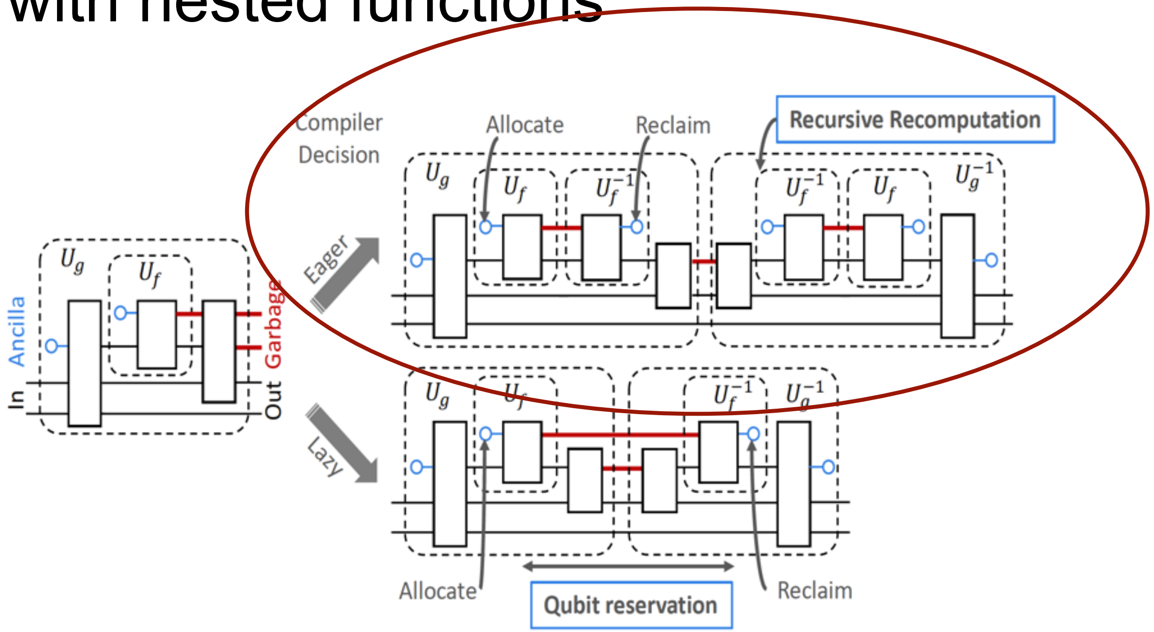
Uncomputation

- Mapping and scheduling for uncomputation and reuse
- Dealing with nested functions



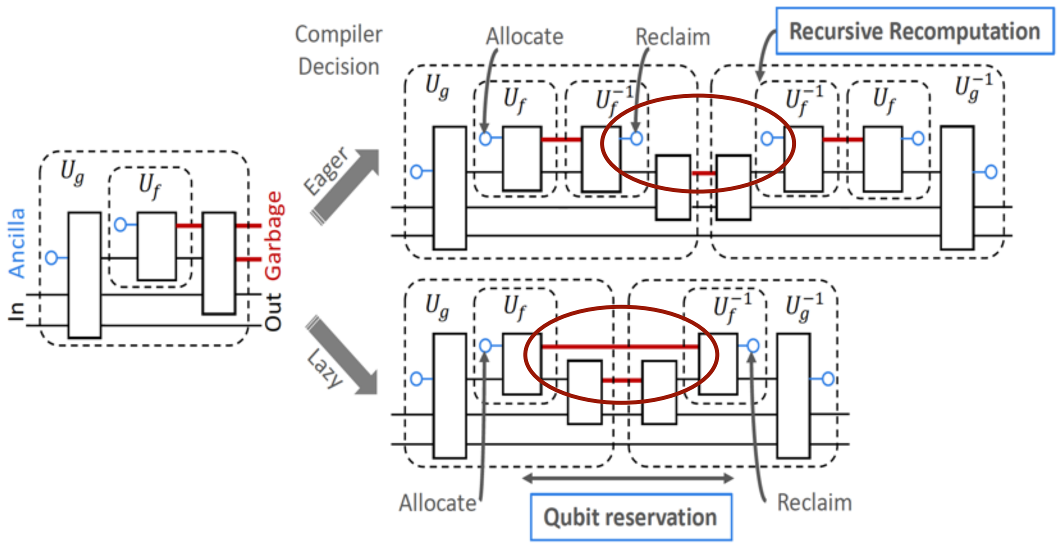
Uncomputation

- Mapping and scheduling for uncomputation and reuse
- Dealing with nested functions



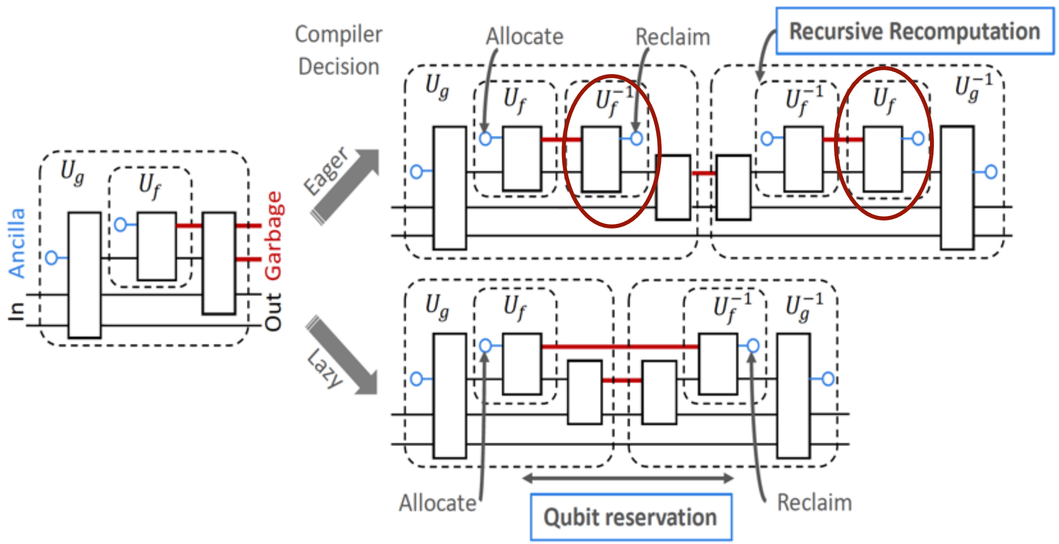
Uncomputation

- Mapping and scheduling for uncomputation and reuse
- Dealing with nested functions



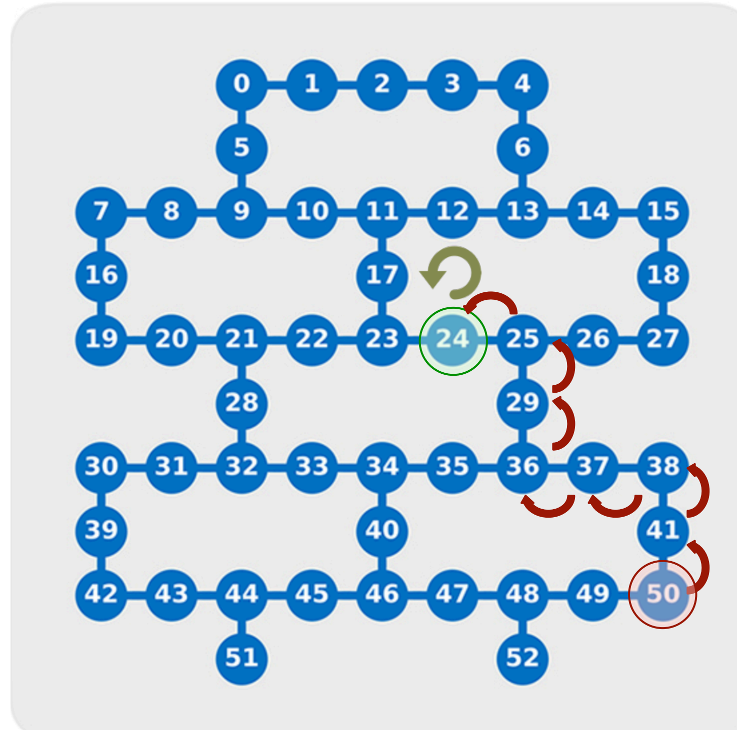
Uncomputation

- Mapping and scheduling for uncomputation and reuse
- Dealing with nested functions



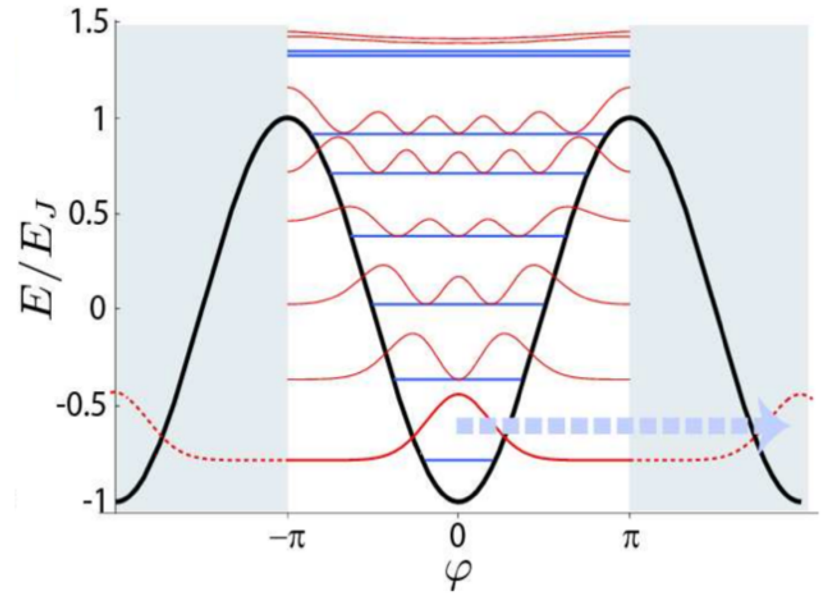
Reuse Results

- Surprisingly, 50% more accurate on current NISQ machines
  - Operations cause errors
  - But uncompute can be cheaper than moving qubits
- 10X more accurate on future machines



5. Qutrits instead of Qubits

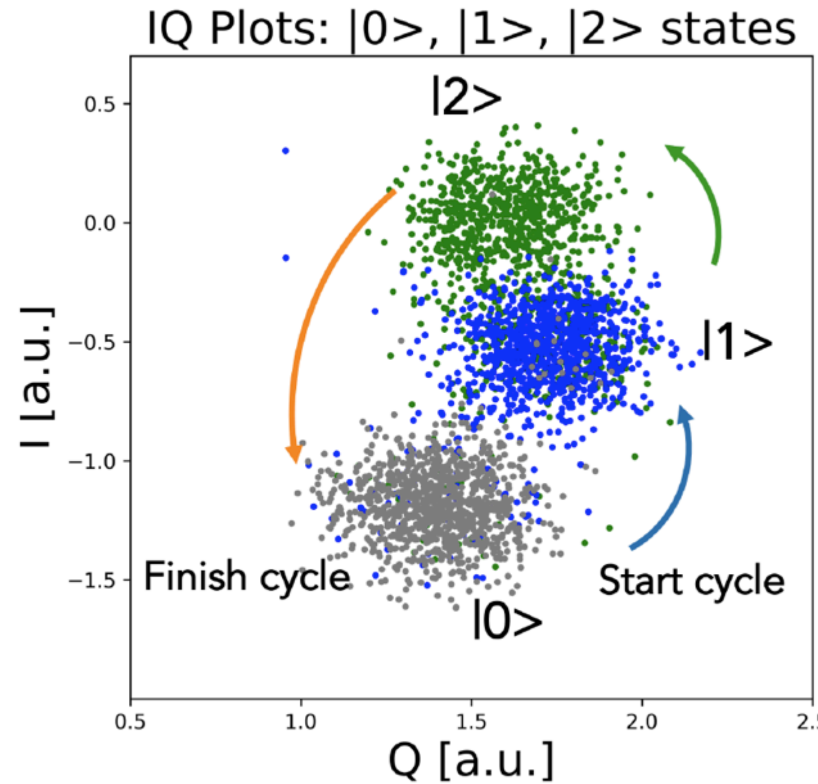
- Store 3 values instead of 2 in each hardware device
- 3-level logic is not new, but makes more sense for quantum devices
- Especially useful for programs that need some extra quantum bits to be more efficient (some temporary space)



[Koch 07]

Qutrit Results

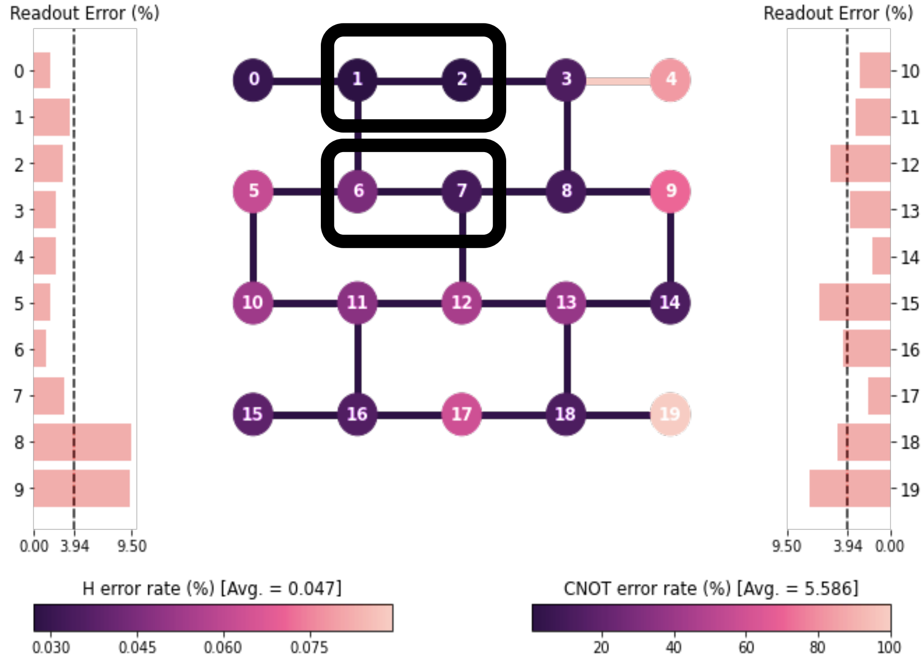
- Fewer devices needed
  - Up to 70X reduction for some programs
- A lot of interest from hardware platforms
  - IBM OpenPulse experiment
- Also won the “Top Picks” best papers for 2019 award



[Gokhale+ Micro20]

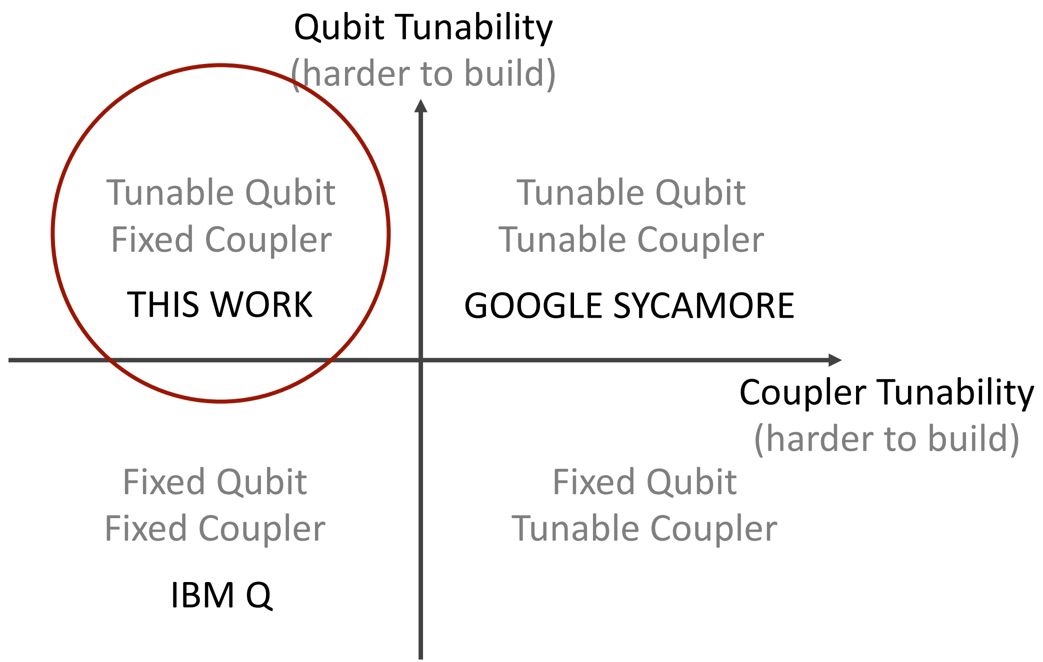
6. Crosstalk Mitigation

Run CNOT 1,2 **in parallel with CNOT 6,7**
Error rate increases by 10X due to crosstalk!

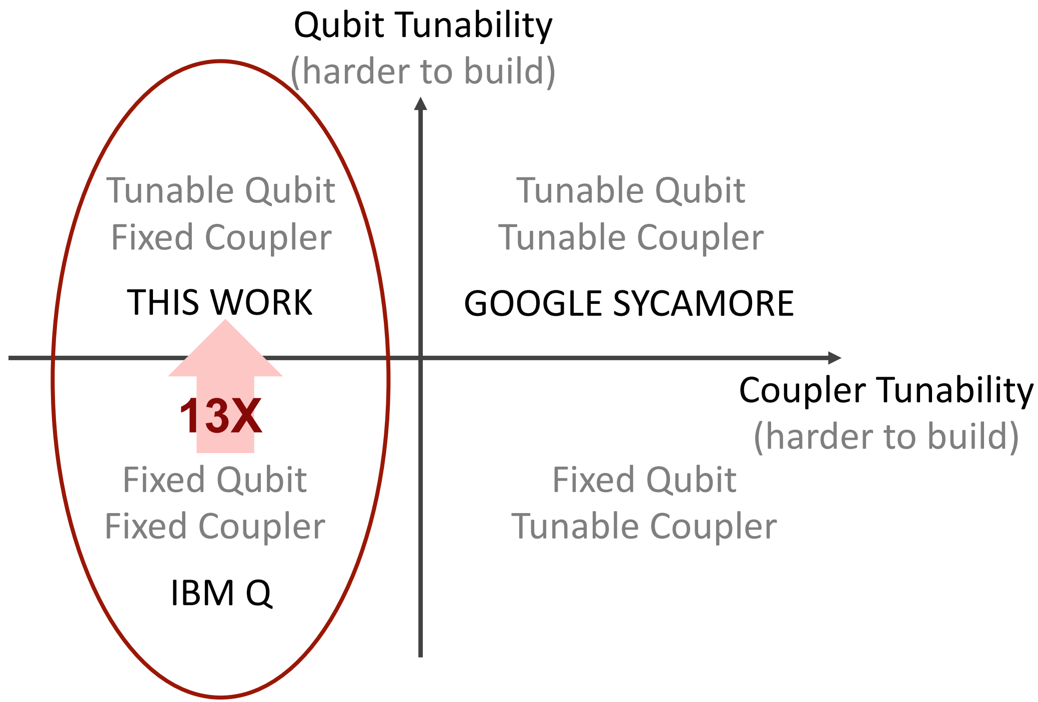


IBMQ Boeblingen 20-qubit device

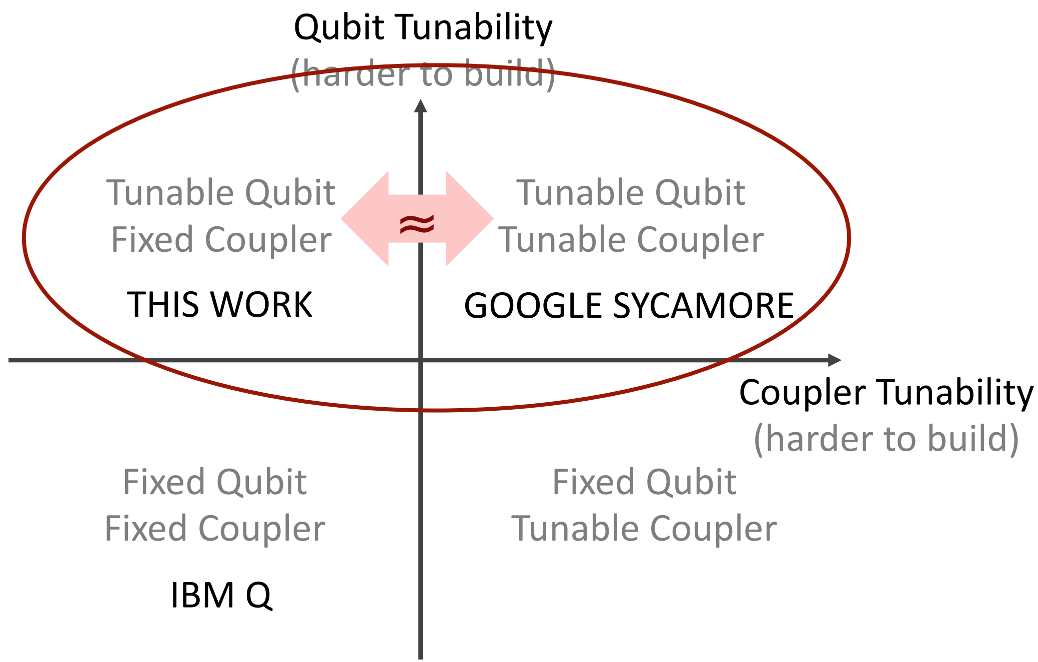
Crosstalk Mitigation Design Space



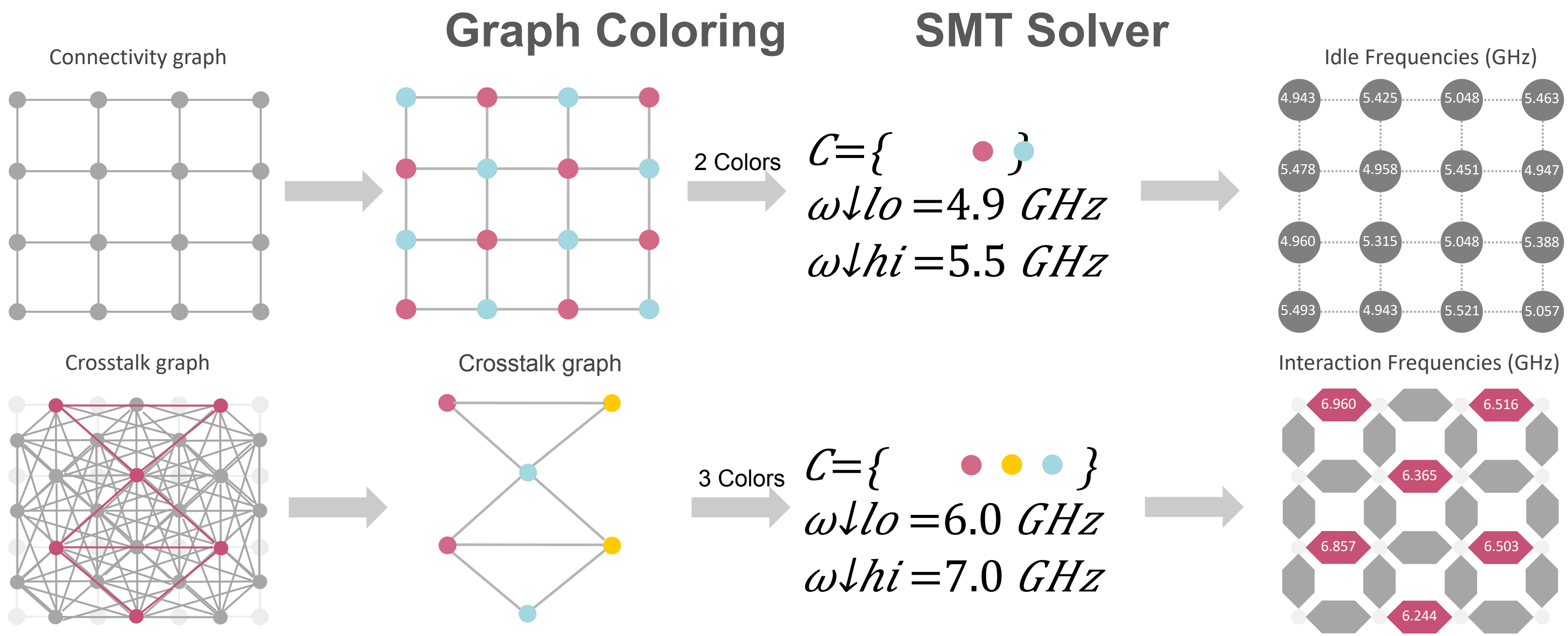
Crosstalk Mitigation Design Space



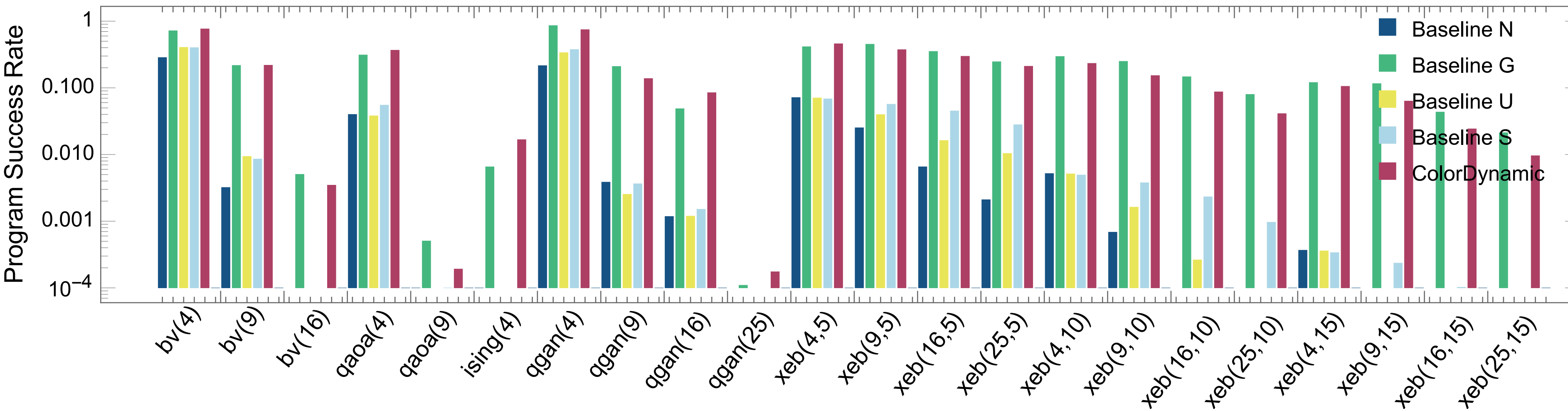
Crosstalk Mitigation Design Space



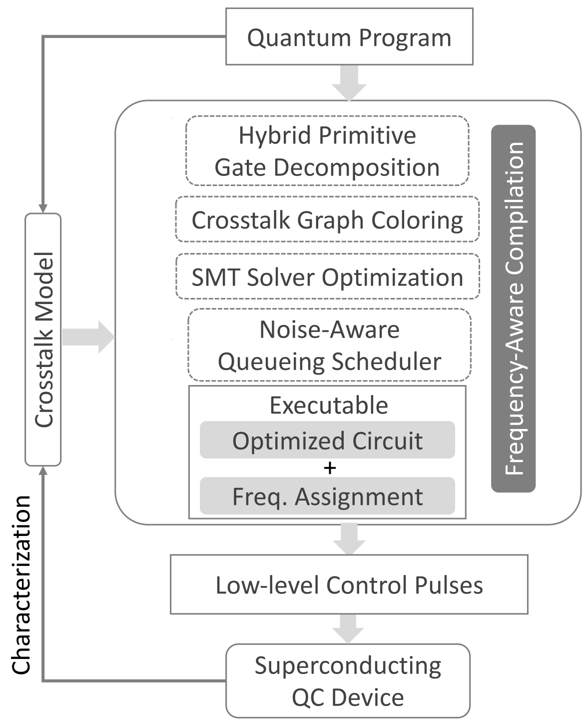
Frequency Scheduling



Crosstalk Mitigation Results

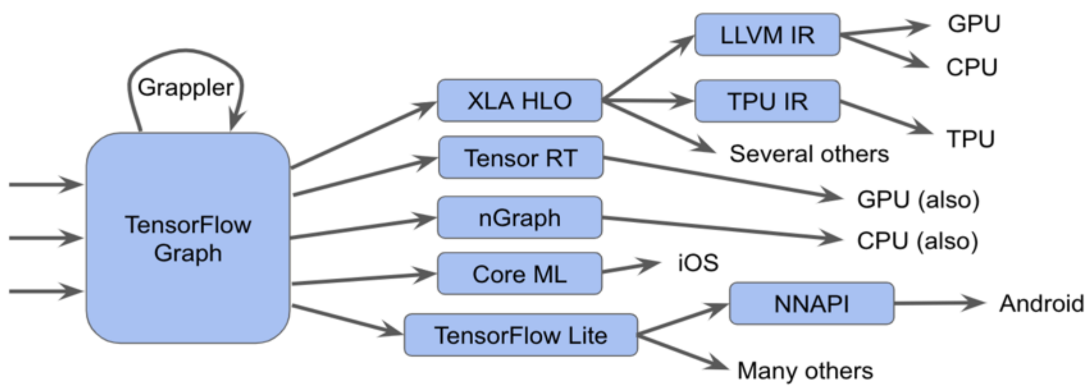


Recap: Quantum Computer Systems Design



Domain-Specific System Design

- Vertically-integrated, physics-aware software stacks
- Analogous to trends in classical systems
  - Hennessey-Patterson Turing Award Lecture



Tensorflow Software Stack for Machine Learning [Tensorflow Blog]

OPEN PROBLEMS

How do I know if my QC program is correct?

- Bootstrapping problem…
  - Quantum hardware will be barely reliable
  - Quantum software will be untested at a scale
- Quantum assertions
  - [Huang ISCA'19, Zhou ASPLOS'20]
- Formal methods (verified compilation):
  - Qwire [Paykin POPL'17], sQIRe [Hietala]
  - Error Bounds [Hung POPL'19]
  - Certiq [Shi]
- Can we check useful properties in polynomial time for programs with quantum supremacy?

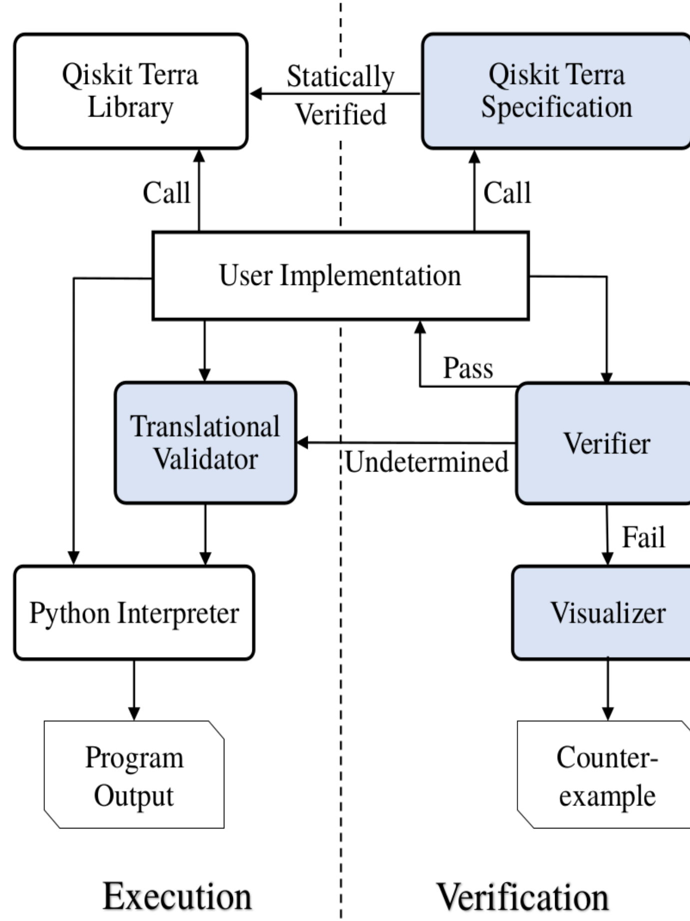
Formal Verification

- *Contract-based Verification of a Realistic Quantum Compiler*, Yunong Shi, Xupeng Li, Runzhou Tao, Ali Javadi-Abhari, Andrew W. Cross, Frederic T. Chong, and Ronghui Gu. arXiv:1908.08963

CertiQ

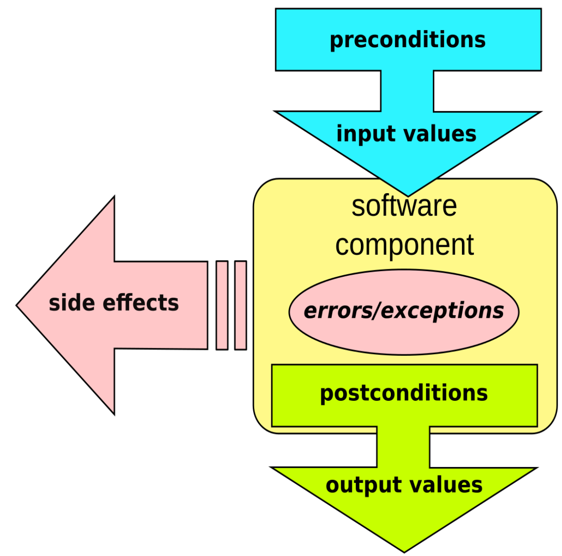
A verification framework based on SMT reasoning in Z3.

- Mostly focus on compilation passes
- For new code submission
- Automated and scalable



Design-by-Contract
Methodology: modular verification

- Pre-conditions
- Post-conditions
- Invariant



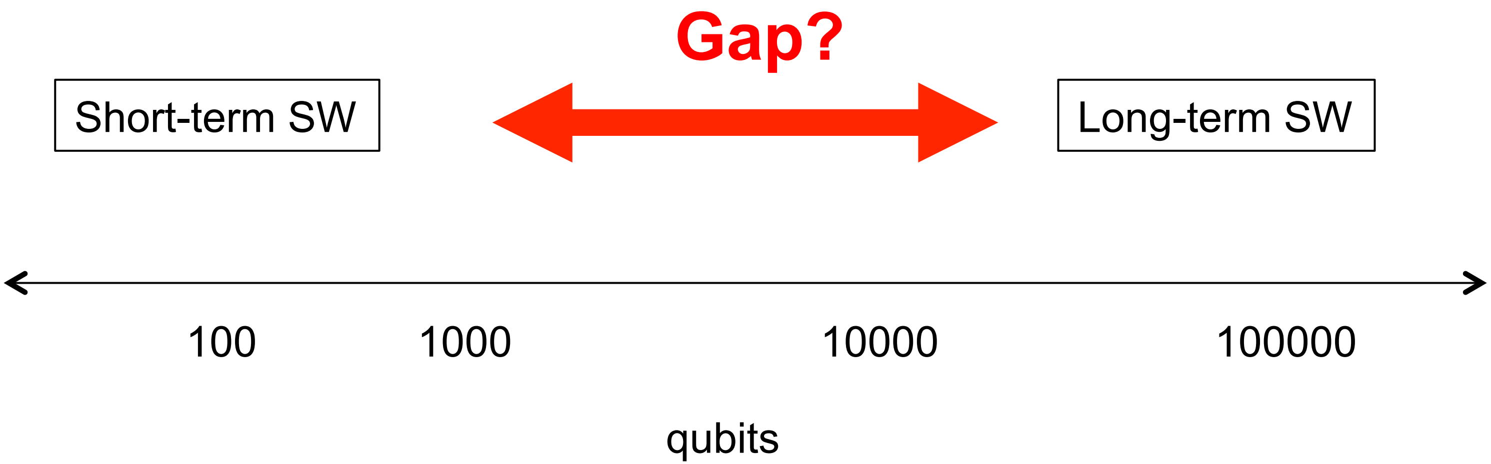
Certiq Summary

- **4 QISKit bugs found: Non-terminating mapper, 2-qubit opt, Commutation pass**
- We should provide safe atomic circuit rewriting methods in quantum software development.
- We should be very careful about the quantum data structure's scope and their equivalence.
- The method of CertiQ can be reused in other layers in the quantum stack and might pave the way to a fully verified quantum system.

Challenge:

Cross-layer optimization versus modular verification

Specialization vs Abstraction



Summary

- QC is at a historic time
- A computer systems view is critical:
  - To accelerate progress
  - To develop in the workforce
- More info: epiqc.cs.uchicago.edu