Noisy intermediate-scale quantum computing: A post-quantum supremacy perspective

Omar Shehab
IonQ/UMD

27 January 2020
shehab@ionq.com

Quantum Information Seminar Series | The University of Tennessee, Knoxville
Collaborators

**IonQ**
- Matthew Kissan
- Yunseong Nam

**ORNL**
- Raphael Pooser
- Eugene Dumitrescu
- Alex McCaskey
- Pavel Lougovski
- Thomas Papenbrock

**UMD**
- Kevin Landsman
- Norbert Linke
- Cinthia Huerta Alderete
- Nhun H. Nguyen
- Daiwei Zhu
- Chris Monroe

**Stanford**
- Isaac H Kim

**JHU/APL**
- Greg Quiroz
Outline

- What is quantum computing?
- Building trapped-ion quantum computers
- Near-term quantum computing
- An example of NISQ algorithm
- Optimizing NISQ algorithms
- Post-quantum supremacy era
What is quantum computing?

- Proposed by Feynman in 1985 with the motivation of simulating quantum systems.
- Based on the principles of quantum mechanics.
- Potential advantage over classical computing comes from quantum phenomena like quantum superposition, quantum entanglement, and quantum measurement.

Companies: IonQ, Google, IBM, Rigetti.
What is quantum computing?

- Not a silver bullet for all computationally hard (for example NP-complete) problems
- Non-von Neumann architecture
- Data is stored in qubits residing in Hilbert space
- Logical operations are complex unitary matrices
- Readout is probabilistic
- Algorithms are often visualized as “quantum circuits”
What is quantum computing?

Example: Quantum algorithm to find an element from an unstructured database.

Quantum computational complexity

- $n \times n$ chess/Go
- Finding the ground state of $k$-local Hamiltonian
- Traveling salesperson
- Graph isomorphism
- Integer factoring
- Primality checking

Extending the graphics from Scott Aaronson’s lecture notes
Quantum algorithm timeline

- **1985**: Feynman proposes quantum computing
- **1994**: Algorithm for prime factoring with exponential speedup
- **1994**: Unstructured database search with quadratic speedup
- **1996**: Quantum chemistry simulation with exponential speedup
- **2005**: Solving special systems of linear equations with exponential speedup
- **2008**: Non-convex optimization with comparable approximation ratio
- **2014**: Quantum chemistry simulation on near-term quantum computers

*References in the back up slides*
Quantum algorithm timeline

1985

Feynman proposes quantum computing

1994 1996

Algorithm for prime factoring
With exponential speedup

Unstructured database search
With quadratic speedup

1994

Quantum chemistry
Simulation with Exponential speedup

2005

Quantum chemistry
Simulation on near-term quantum computers

2008

Non-convex optimization with Comparable approximation ratio

2014

Triggered funded academic research programs

* References in the back up slides
Quantum algorithm timeline

- Feynman proposes quantum computing
- Algorithm for prime factoring with exponential speedup
- Unstructured database search with quadratic speedup
- Quantum chemistry simulation with exponential speedup
- Solving special systems of linear equations with exponential speedup
- Non-convex optimization with comparable approximation ratio
- Quantum chemistry simulation on near-term quantum computers

Triggered private industrial effort mainly focusing on chemistry/physics simulation

* References in the back up slides
Quantum algorithm timeline

1985
Feynman proposes quantum computing

1994
Quantum chemistry simulation with exponential speedup

1996
Unstructured database search with quadratic speedup

2005
Solving special systems of linear equations with exponential speedup

2008
Non-convex optimization with comparable approximation ratio

2014
Quantum chemistry simulation on near-term quantum computers

2015

Potential for commercial incentives still not clear for ML/DL/optimization

* References in the back up slides
Potential market

- **Pharma**: molecular design, drug delivery
- **Energy**: chemical processes, material design
- **Transportation**: navigation, logistics
- **Finance**: market models, optimization
Quantum computing @ IonQ
Coming soon to a cloud near you!

Microsoft Ignite, Nov 4 2019

AWS Invent, Dec 4 2019
Building quantum computers @ IonQ

Source: arXiv:1903.08181
The quantum processing unit

Linear computation: montage of a photo of the chip containing the trapped ions and an image of the ions in a 1D array (Courtesy: Christopher Monroe)
Trapped Ion Hyperfine Qubit: $^{171}\text{Yb}^+$

- State-dependent fluorescence provides high fidelity detection

\[ \gamma/2\pi = 20 \text{ MHz} \]

\[ 369 \text{ nm} \quad (811.9 \text{ THz}) \]

\[ 2S_{1/2} \quad |\uparrow\rangle = |1,0\rangle \quad |\downarrow\rangle = |0,0\rangle \]

S. Olmschenk et al., PRA 76, 052314, Slide (2007)

From the March Meeting talk of Jungsang Kim

S. Olmschenk et al., PRA 76, 052314, Slide (2007)

From the March Meeting talk of Jungsang Kim
Entangling Trapped Ion Qubits

\[ \Delta E = \frac{e^2}{\sqrt{r^2 + \delta^2}} - \frac{e^2}{r} \approx - \frac{(e\delta)^2}{2r^3} \]

\[ e\delta \sim 500 \text{ Debye} \]

\[ \delta \sim 10 \text{ nm} \]

Native Ion Trap Operation: “Ising” gate

\[ XX[\varphi] = e^{-i\sigma_x^{(1)} \sigma_x^{(2)} \varphi} \]

\[ T_{\text{gate}} \sim 10-100 \mu s \]

\[ F \sim 98\% - 99.9\% \]

From the March Meeting talk of Jungsang Kim

Cirac and Zoller (1995)
Mølmer & Sørensen (1999)
Solano, de Matos Filho, Zagury (1999)
Milburn, Schneider, James (2000)
Assemblying large number of qubits

More ways to apply fundamental logical operation directly on hardware
# Platform comparison

<table>
<thead>
<tr>
<th>QC Platform</th>
<th>Qubit Type</th>
<th>$T_1 / T_2$ (s)</th>
<th>1 &amp; 2 Qubit Gate Error</th>
<th>#2-Q Gates Run in a Circuit</th>
<th>Maximum # Qubits Entangled</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trapped Ions</strong></td>
<td>Natural</td>
<td>$&gt; 10^{10} / 10^3$</td>
<td>$&lt; 10^{-5} / 10^{-3}$</td>
<td>56</td>
<td>20</td>
</tr>
<tr>
<td><strong>Neutral Atoms</strong></td>
<td>Natural</td>
<td>$&lt; 10^{-4} / 10^{-4}$</td>
<td>$&lt; 10^{-3} / 0.03$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Photonic Circuits</strong></td>
<td>Engineered</td>
<td>Photon Loss</td>
<td>Probabilistic</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Superconductors</strong></td>
<td>Engineered</td>
<td>$&lt; 10^{-3} / 10^{-3}$</td>
<td>$&lt; 10^{-3} / 0.005$</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td><strong>Silicon Spin</strong></td>
<td>Engineered</td>
<td>$10^{-3}/10^{-5}$</td>
<td>$&gt; 10^{-3}/0.1$</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
## Nature’s qubit

<table>
<thead>
<tr>
<th>QC Platform</th>
<th>Qubit Type</th>
<th>$T_1/T_2$ (s)</th>
<th>1 &amp; 2 Qubit Gate Error</th>
<th>#2-Q Gates Run in a Circuit</th>
<th>Maximum # Qubits Entangled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trapped Ions</td>
<td>Natural</td>
<td>$&gt;10^{10}/10^3$</td>
<td>$&lt;10^{-5}/10^{-3}$</td>
<td>56</td>
<td>20</td>
</tr>
<tr>
<td>Neutral Atoms</td>
<td>Natural</td>
<td>$&lt;10^{-4}/10^{-4}$</td>
<td>$&lt;10^{-3}/0.03$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Photonic Circuits</td>
<td>Engineered</td>
<td>Photon Loss</td>
<td>Probabilistic</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Superconductors</td>
<td>Engineered</td>
<td>$&lt;10^{-3}/10^{-3}$</td>
<td>$&lt;10^{-3}/0.005$</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Silicon Spin</td>
<td>Engineered</td>
<td>$10^{-3}/10^{-5}$</td>
<td>$&gt;10^{-3}/0.1$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>QC Platform</td>
<td>Qubit Type</td>
<td>$T_1/T_2$ (s)</td>
<td>1 &amp; 2 Qubit Gate Error</td>
<td>#2-Q Gates Run in a Circuit</td>
<td>Maximum # Qubits Entangled</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>---------------</td>
<td>------------------------</td>
<td>-----------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Trapped Ions</td>
<td>Natural</td>
<td>$&gt;10^{10}/10^3$</td>
<td>$&lt;10^{-5}/10^{-3}$</td>
<td>56</td>
<td>20</td>
</tr>
<tr>
<td>Neutral Atoms</td>
<td>Natural</td>
<td>$&lt;10^{-4}/10^{-4}$</td>
<td>$&lt;10^{-3}/0.03$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Photonic Circuits</td>
<td>Engineered</td>
<td>Photon Loss</td>
<td>Probabilistic</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Superconductors</td>
<td>Engineered</td>
<td>$&lt;10^{-3}/10^{-3}$</td>
<td>$&lt;10^{-3}/0.005$</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Silicon Spin</td>
<td>Engineered</td>
<td>$10^{-3}/10^{-5}$</td>
<td>$&gt;10^{-3}/0.1$</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
## Noise rate

<table>
<thead>
<tr>
<th>QC Platform</th>
<th>Qubit Type</th>
<th>$T_1/T_2$ (s)</th>
<th>1 &amp; 2 Qubit Gate Error</th>
<th>#2-Q Gates Run in a Circuit</th>
<th>Maximum # Qubits Entangled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trapped Ions</td>
<td>Natural</td>
<td>$&gt; 10^{10}/10^3$</td>
<td>$&lt; 10^{-5}/10^{-3}$</td>
<td>56</td>
<td>20</td>
</tr>
<tr>
<td>Neutral Atoms</td>
<td>Natural</td>
<td>$&lt; 10^{-4}/10^{-4}$</td>
<td>$&lt; 10^{-3}/0.03$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Photonic Circuits</td>
<td>Engineered</td>
<td>Photon Loss</td>
<td>Probabilistic</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Superconductors</td>
<td>Engineered</td>
<td>$&lt; 10^{-3}/10^{-3}$</td>
<td>$&lt; 10^{-3}/0.005$</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Silicon Spin</td>
<td>Engineered</td>
<td>$10^{-3}/10^{-5}$</td>
<td>$&gt; 10^{-3}/0.1$</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
## Longest sequence of logical operation

<table>
<thead>
<tr>
<th>QC Platform</th>
<th>Qubit Type</th>
<th>$T_1$ / $T_2$ (s)</th>
<th>1 &amp; 2 Qubit Gate Error</th>
<th>#2-Q Gates Run in a Circuit</th>
<th>Maximum # Qubits Entangled</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trapped Ions</strong></td>
<td>Natural</td>
<td>$&gt; 10^{10} / 10^3$</td>
<td>$&lt; 10^{-5} / 10^{-3}$</td>
<td><strong>56</strong></td>
<td>20</td>
</tr>
<tr>
<td><strong>Neutral Atoms</strong></td>
<td>Natural</td>
<td>$&lt; 10^{-4} / 10^{-4}$</td>
<td>$&lt; 10^{-3} / 0.03$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>Photonic Circuits</strong></td>
<td>Engineered</td>
<td>Photon Loss</td>
<td>Probabilistic</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Superconductors</strong></td>
<td>Engineered</td>
<td>$&lt; 10^{-3} / 10^{-3}$</td>
<td>$&lt; 10^{-3} / 0.005$</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td><strong>Silicon Spin</strong></td>
<td>Engineered</td>
<td>$10^{-3} / 10^{-5}$</td>
<td>$&gt; 10^{-3} / 0.1$</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Largest input size used so far

<table>
<thead>
<tr>
<th>QC Platform</th>
<th>Qubit Type</th>
<th>$T_1/T_2$ (s)</th>
<th>1 &amp; 2 Qubit Gate Error</th>
<th>#2-Q Gates Run in a Circuit</th>
<th>Maximum # Qubits Entangled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trapped Ions</td>
<td>Natural</td>
<td>$&gt;10^{10}/10^3$</td>
<td>$&lt;10^{-5}/10^{-3}$</td>
<td>56</td>
<td>20</td>
</tr>
<tr>
<td>Neutral Atoms</td>
<td>Natural</td>
<td>$&lt;10^{-4}/10^{-4}$</td>
<td>$&lt;10^{-3}/0.03$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Photonic Circuits</td>
<td>Engineered</td>
<td>Photon Loss</td>
<td>Probabilistic</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Superconductors</td>
<td>Engineered</td>
<td>$&lt;10^{-3}/10^{-3}$</td>
<td>$&lt;10^{-3}/0.005$</td>
<td>15</td>
<td>9</td>
</tr>
<tr>
<td>Silicon Spin</td>
<td>Engineered</td>
<td>$10^{-3}/10^{-5}$</td>
<td>$&gt;10^{-3}/0.1$</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
History of improvement for gate error
Ions vs. Superconductors


Measured Success Rate

Benchmark Algorithm

F: Failure
Scaling technology

Shuttling of ion chain: tape+head model

Modular shuttling between multiple zones

Modular photonic connections between chips
The promise

Classical computers are a dead end for certain classes of problems.

**Factoring Problem**

- **CLASSICAL COMPUTER: STATE OF THE ART**
- **CLASSICAL COMPUTER: 30 YEARS FROM NOW (MOORE’S LAW)**
- **QUANTUM COMPUTER**
The promise

<table>
<thead>
<tr>
<th>SIZE OF PROBLEM (BITS)</th>
<th>TIME TO SOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 BILLION YRS</td>
<td>1 BILLION YRS</td>
</tr>
<tr>
<td>1 MILLION YRS</td>
<td>1 MILLION YRS</td>
</tr>
<tr>
<td>1,000 YRS</td>
<td>1,000 YRS</td>
</tr>
<tr>
<td>100 YRS</td>
<td>100 YRS</td>
</tr>
<tr>
<td>10 YRS</td>
<td>10 YRS</td>
</tr>
<tr>
<td>1 YR</td>
<td>1 YR</td>
</tr>
<tr>
<td>1 MO</td>
<td>1 MO</td>
</tr>
<tr>
<td>1 DAY</td>
<td>1 DAY</td>
</tr>
<tr>
<td>1 HR</td>
<td>1 HR</td>
</tr>
<tr>
<td>100 SEC</td>
<td>100 SEC</td>
</tr>
<tr>
<td>1 SEC</td>
<td>1 SEC</td>
</tr>
<tr>
<td>1 MO</td>
<td>1 MO</td>
</tr>
<tr>
<td>CLASSICAL COMPUTER:</td>
<td></td>
</tr>
<tr>
<td>STATE OF THE ART</td>
<td></td>
</tr>
<tr>
<td>CLASSICAL COMPUTER:</td>
<td></td>
</tr>
<tr>
<td>30 YEARS FROM NOW</td>
<td></td>
</tr>
<tr>
<td>(MOORE’S LAW)</td>
<td></td>
</tr>
<tr>
<td>QUANTUM COMPUTER</td>
<td></td>
</tr>
</tbody>
</table>

Factoring Problem

So called “quantum supremacy”
Near-term quantum computing
The roadmap (near term)

Algorithm Improvements

- Factoring
- Quantum Machine Learning
- Quantum Chemistry/Simulations
- Nitrogenase
- Simple Organics

Log(Performance)

Computational Power

H₂ HeH H₂O O₃

2015 2017 2019 2021 2023 2025
The roadmap (near term)

- Algorithm Improvements
- Quantum Machine Learning
- Quantum Chemistry/Simulations
- Factoring
- Simple Organics
- Nitrogenase

Published works:
- H₂
- HeH
- H₂O
- O₃
The roadmap (near term)

- Algorithm Improvements
  - Factoring
  - Quantum Machine Learning
  - Quantum Chemistry/Simulations
  - Simple Organics

- Log(Performance) vs. Computational Power

- Molecules: H₂, HeH, H₂O, O₃

- Nitrogenase
- Simple Organics

- IONQ
Practical challenges of building a QC

- Computation makes qubits (unit of quantum registers) interact with each other
- Qubits are susceptible to noise coming from the surrounding environment
- It is hard to filter out unwanted interaction while allowing computationally meaningful interactions

Source: Preskill, 2018
Practical challenges of building a QC

GPU errors (Tiwari et. al 2015)

Quantum processing unit (QPU) errors (Wright et. al 2019)
NISQ computing

- Noisy intermediate-scale quantum (NISQ) computers = First generation quantum computers
- Name coined by John Preskill (2018)
- Amount of computation very limited due to high failure rate
- Upper bound for quantum advantage is quadratic when the structure of the problem is not known
- With knowledge about the structure it may improve
Algorithmic framework on NISQ computer

- The computational problem is encoded in a quantum-native format
- The goal is to construct a quantum state which encodes the solution
- This construction is done variationally using a parametrized quantum circuit (a simple quantum algorithm with lots of variables)
- For many problems, the solution is the ground state of the system
- Classical stochastic optimization routines are used to find the ground state
Domains of applications

- Simulation of physics/chemistry
- Hard optimization problems
Standard NISQ algorithms

- Simulation of physics/chemistry
  - Quantum advantage is well understood
- Hard optimization problems
  - Quantum advantage is still under investigation
NISQ algorithmic schemes

- Simulation of physics/chemistry
  - Variational quantum eigensolver algorithm (McClean et. al 2015)
- Hard optimization problems
  - Quantum approximate optimization algorithm (Farhi et. al 2014)
Quantum approximate optimization algorithm

- Define a measurable quantum quantity which encodes the solution to the problem
- Design a tunable method to explore the quantum search space (aka run a quantum algorithm)
- Keep tuning the method using a classical routine until you get a reasonable solution
NISQ algorithm example
An example NISQ problem

- Compute the MAXCUT of the following graph
An example NISQ problem

- Compute the MAXCUT of the following graph

- General MAXCUT problem is NP-complete
- Application: Pattern recognition, electronic circuitry design, state problems in statistical physics, and combinatorial optimization
An example NISQ problem

- Compute the MAXCUT of the following graph

  ![Graph Image]

- Best known exact algorithm needs exponential time
- Approximation algorithms are used in practice
An example NISQ problem

- Compute the MAXCUT of the following graph

1. MAXCUT = 4
2. Two solutions
An example NISQ problem

- Let’s label the two partitions as 0 and 1.
- 0000 => all vertices belong to partition #0.
- 1010 => the first and third nodes belong to partition #1.
- There are $2^4 = 16$ possible partitions.
- A search problem!
An example NISQ problem

- A classical exact algorithm searches the solution in the integer space
- A weighted version of the problem means searching in the real space
- A quantum search (=optimization algorithm) algorithm looks for the solution in a complex vector space which is huuuuuuge!
An example NISQ problem

- A classical exact algorithm searches the solution in the integer space
- A weighted version of the problem means searching in the real space
- A quantum search (=optimization algorithm) algorithm looks for the solution in a complex vector space which is huuuuuuge!

Good or bad?
The Grover speed up

- Thanks to unique quantum phenomenon - destructive interference while exploring the search tree
- Unstructured search may speed up quadratically
- Theoretical upper limit
- In practice, fractional Grover speedup is all you can hope
Back to the example

- Translate the problem into quantum-friendly language
- For any edge <j,k> between the nodes j and k, define the following:

\[ C_{<j,k>} = \frac{1}{2}(\sigma_j^z \sigma_k^z + 1) \]
An example NISQ problem

- Translate the problem into quantum-friendly language
- For any edge \( <j,k> \) between the nodes \( j \) and \( k \), define the following:

\[
C_{<j,k>} = \frac{1}{2}(-\sigma^z_j \sigma^z_k + 1)
\]

- \( \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \)
- The lowest eigenvalue occurs when the nodes \( j \) and \( k \) are both in the same partition
- The highest eigenvalue occurs when the nodes \( j \) and \( k \) are in different partitions
Tl;dr: Still a hard problem

- $C = \sum_{j,k} C_{j,k}$ gives you the CUT
- $C$ is to be measured on a quantum system
- Maximizing $C$ or minimizing $-C$ gives you the MAXCUT
- Minimizing $-C = \text{Sending the quantum system encoding the problem to lowest energy state}$
- Finding the lowest energy of a system is NP-hard in classical physics
- It is QMA-hard in quantum physics
An example NISQ problem

- For each edge, vary the probability of the connected nodes being in different or same partition
- Find a combination of these probabilities which yields the lowest value for \(-C\)
An example NISQ problem

- Varying the probability of the connected nodes = switching the quantum system from one state to another
- Hopefully one of those states will be the (approximate?) solution

This angle dictates the probability
An example NISQ problem
An example NISQ problem

Creates an initial superposition of complete search space
An example NISQ problem

Corresponds to the edge between node #1 and #2
An example NISQ problem
An example NISQ problem
An example NISQ problem
An example NISQ problem
An example NISQ problem

Checks symmetry
An example NISQ problem

- For each term in C, measure the qubit to determine the probability of being in 1
- Variationally update $\gamma$ and $\beta$ to get to the lowest energy
- Repetition of this layer can be viewed as constrained optimization problem which yields higher approximation ratio
- End result will be close to -4 but never beyond that
Optimizing NISQ algorithms
Past causal cones?

Prof. Guifré Vidal

Prof. Glen Evenbly*

Evenbly–Vidal 2013
A “harder” problem

Shehab et. al 2019
A “harder” problem

Shehab et. al 2019
Improvements by noise reduction?

Shehab et. al 2019
Improvements by noise reduction?

- 40% increase in accuracy
- 78% fewer measurements

Figure 6: Theoretical and experimentally determined MAXCUT for the original QAOA ansatz and the reduced ansatz at the parameters (γ = 1.358, β = 2.462, p = 1). Absolute standard errors for individual Hamiltonian term is reported in Appendix A (Inset: Absolute standard errors).
Solving practical problems?

- Practical problems may contain millions of nodes and edges
- If one node = one qubit, we need millions of qubits
- Millions of edges \( \approx 1/\text{million} \) noise rate in each logical operation
- Theoretical upper limit on quantum advantage is quadratic speedup
- Special knowledge about the structure of the problem might help go beyond that
- Loading classical input onto a quantum computer is not very efficient
When a problem is quantum-worthy?

**Figure:** Hardness to solve 3SAT problems

Randomly chosen instances by Zhou, 2003
Informed hybrid approach?

- Practical NP-complete problems are often easy
- Hard to get the exact solution
- Think about a search problem with gradient descent where most of the search landscape is smooth but the basin of the global minima is rugged
- Classical computer should handle most of it and delegate to quantum when the solution landscape is close to random

Workshop on Quantum Machine Learning, 2018
Rays of hope

- HHL algorithm with exponential speedup for systems of linear equation (Harrow et. al, 2009)
- Quadratic speedup for SAT problems (Dunjko et. al, 2018)
- Quantum gradient descent (Kerenidis 2019)
- Quadratic speedup for semidefinite programming solvers (Brandão et. al, 2019)
- Error reduction with past causal cones for non-convex quantum optimization (Shehab et. al 2019)
- Variational search space reduction with chaos theory for non-convex quantum optimization (Shehab et. al anytime soon at your nearest arXiv)
The era of post-quantum supremacy
How it all started?

QUANTUM COMPUTING AND
THE ENTANGLEMENT FRONTIER

JOHN PRESKILL

Institute for Quantum Information and Matter
California Institute of Technology
Pasadena, CA 91125, USA

Quantum information science explores the frontier of highly complex quantum states, the “entanglement frontier.” This study is motivated by the observation (widely believed but unproven) that classical systems cannot simulate highly entangled quantum systems efficiently, and we hope to hasten the day when well controlled quantum systems can perform tasks surpassing what can be done in the classical world. One way to achieve such “quantum supremacy” would be to run an algorithm on a quantum computer which solves a problem with a super-polynomial speedup relative to classical computers, but there may be other ways that can be achieved sooner, such as simulating exotic quantum states of strongly correlated matter. To operate a large scale quantum computer reliably we will need to overcome the debilitating effects of decoherence, which might be done using “standard” quantum hardware protected by quantum error-correcting codes, or by exploiting the nonabelian quantum statistics of anyons realized in solid state systems, or by combining both methods. Only by challenging the entanglement frontier will we learn whether Nature provides extravagant resources far beyond what the classical world would allow.

Rapporteur talk at the 25th Solvay Conference on Physics
“The Theory of the Quantum World”
Brussels, 19-22 October 2011
Formalization of the experiment

Characterizing Quantum Supremacy in Near-Term Devices

Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, Hartmut Neven

(Submitted on 31 Jul 2016 (v1), last revised 5 Apr 2017 (this version, v3))

A critical question for the field of quantum computing in the near future is whether quantum devices without error correction can perform a well-defined computational task beyond the capabilities of state-of-the-art classical computers, achieving so-called quantum supremacy. We study the task of sampling from the output distributions of (pseudo-)random quantum circuits, a natural task for benchmarking quantum computers. Crucially, sampling this distribution classically requires a direct numerical simulation of the circuit, with computational cost exponential in the number of qubits. This requirement is typical of chaotic systems. We extend previous results in computational complexity to argue more formally that this sampling task must take exponential time in a classical computer. We study the convergence to the chaotic regime using extensive supercomputer simulations, modeling circuits with up to 42 qubits - the largest quantum circuits simulated to date for a computational task that approaches quantum supremacy. We argue that while chaotic states are extremely sensitive to errors, quantum supremacy can be achieved in the near-term with approximately fifty superconducting qubits. We introduce cross entropy as a useful benchmark of quantum circuits which approximates the circuit fidelity. We show that the cross entropy can be efficiently measured when circuit simulations are available. Beyond the classically tractable regime, the cross entropy can be extrapolated and compared with theoretical estimates of circuit fidelity to define a practical quantum supremacy test.
The Google result
Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits

Edwin Pednault*\(^1\), John A. Gunnels\(^1\), Giacomo Nannicini\(^1\), Lior Horesh\(^1\), and Robert Wisnieff\(^1\)

\(^1\)IBM T.J. Watson Research Center, Yorktown Heights, NY

Abstract

In a recent paper, we showed that secondary storage can extend the range of quantum circuits that can be practically simulated with classical algorithms. Here we refine those techniques and apply them to the simulation of Sycamore circuits with 53 and 54 qubits, with the entanglement pattern ABCDCDAB that has proven difficult to classically simulate with other approaches. Our analysis shows that on the Summit supercomputer at Oak Ridge National Laboratories, such circuits can be simulated with high fidelity to arbitrary depth in a matter of days, outputting all the amplitudes.
What is the point?

- Amazing progress in the hardware

Google (The supremacy paper)  
IBM (Jay Gambetta)
What is the point?

- New ideas in theoretical computer science

Complexity-Theoretic Foundations of Quantum Supremacy Experiments

Scott Aaronson* Lijie Chen†

Abstract

In the near future, there will likely be special-purpose quantum computers with 40-50 high-quality qubits. This paper lays general theoretical foundations for how to use such devices to demonstrate “quantum supremacy”: that is, a clear quantum speedup for some task, motivated by the goal of overturning the Extended Church-Turing Thesis as confidently as possible. First, we study the hardness of sampling the output distribution of a random quantum circuit, along

ON THE CLASSICAL HARDNESS OF SPOOFING LINEAR CROSS-ENTROPY BENCHMARKING

SCOTT AARONSON* AND SAM GUNN†

Abstract. Recently, Google announced the first demonstration of quantum computational supremacy with a programmable superconducting processor (Arute et al. [2019]). Their demonstration is based on collecting samples from the output distribution of a noisy random quantum circuit, then applying a statistical test to those samples called Linear Cross-Entropy Benchmarking (Linear XEB). This raises a theoretical question: How hard is it for a classical computer to spoof the results of the Linear XEB test? In this short note, we adapt an analysis of Aaronson and Chen [2017] to prove a conditional hardness result for Linear XEB spoofing. Specifically, we show that the problem is classically hard, assuming a gap in the complexity hierarchy that is currently supported only by speculation.
Thank you and we are hiring!

Open Positions

<table>
<thead>
<tr>
<th>Position</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational Physicist</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>FPGA Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Front End Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Full Stack Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Hardware Design Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>High Performance Computing Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Optical Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Principal Operations Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Product Designer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Quantum Applications and Solutions Scientist</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Quantum Circuit Compiler Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Quantum Control Theorist</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Recruiting Intern (Part-Time)</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Scientific Computing Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Senior Analog Mixed-Signal Design Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Senior FPGA Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Senior Hardware Design Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Senior Mechanical Engineer</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Senior Physicist</td>
<td>College Park, Maryland</td>
</tr>
<tr>
<td>Senior Software Engineer</td>
<td>College Park, Maryland</td>
</tr>
</tbody>
</table>