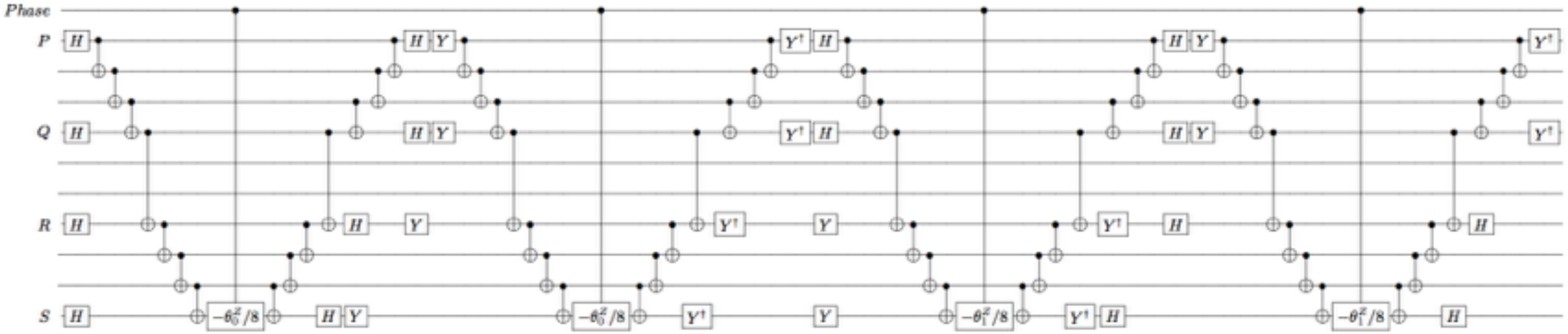


What Can We Do with a Quantum Computer?

Matthias Troyer – Station Q, ETH Zurich

Classical computers have come a long way

Antikythera mechanism

astronomical positions

(100 BC)

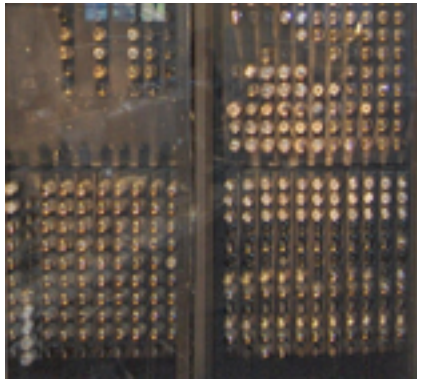
Kelvin's harmonic analyzer

prediction of tides

(1878)

Difference Engine

(1822)



ENIAC

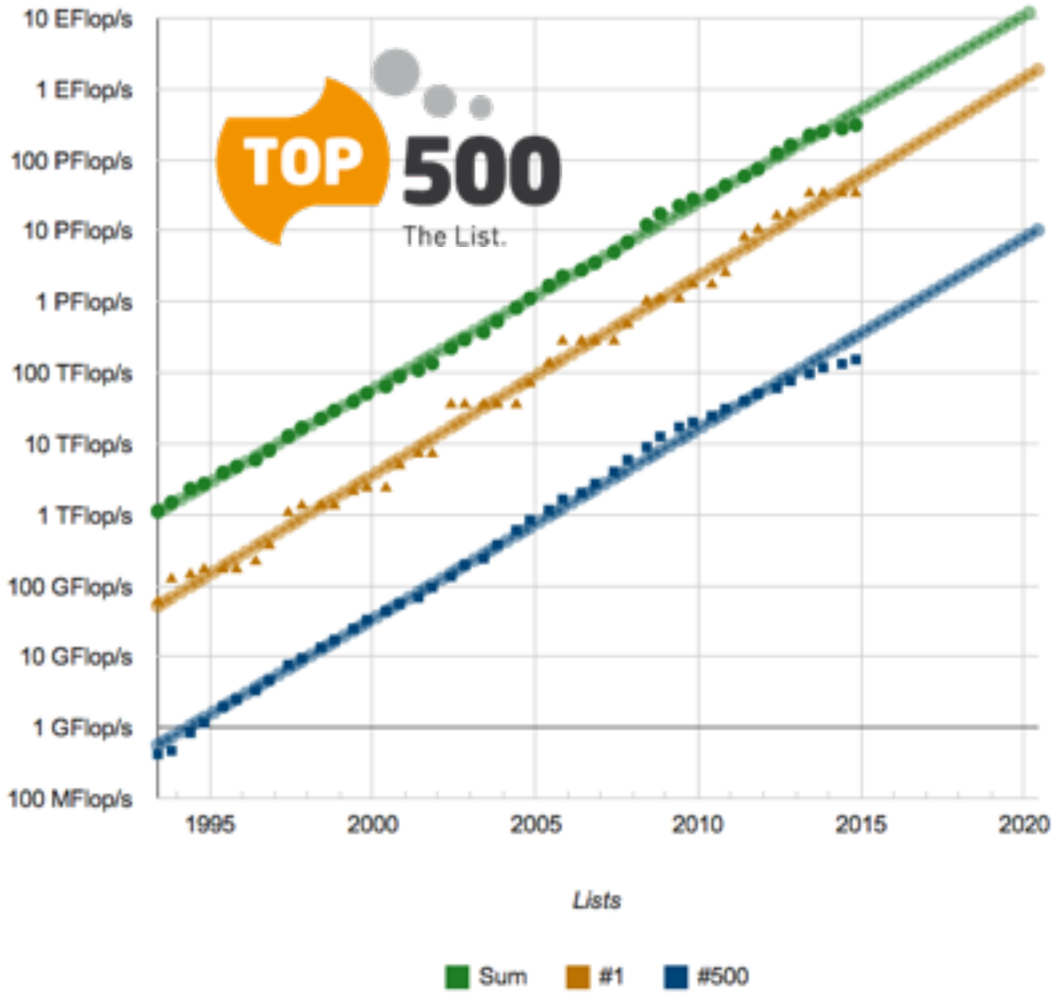
(1946)

Titan, ORNL

(2013)

Is there anything that we cannot solve on future supercomputers?

How long will Moore's law continue?



Do we see signs of the end of Moore's law?

Can we go below 7nm feature size?

Can we use more than 3 million cores?

Can we fight the recent exponential increase in power consumption?

Enabling technologies for beyond exascale computing

- **We are not referring to 10**21 flops**
- **"Beyond exascale" systems as we are defining them will be based on new technologies that will finally result in the much anticipated (but unknown) phase change to truly new paradigms/methodologies.**

Paul Messina
Director of Science
Argonne Leadership Computing Facility
Argonne National Laboratory

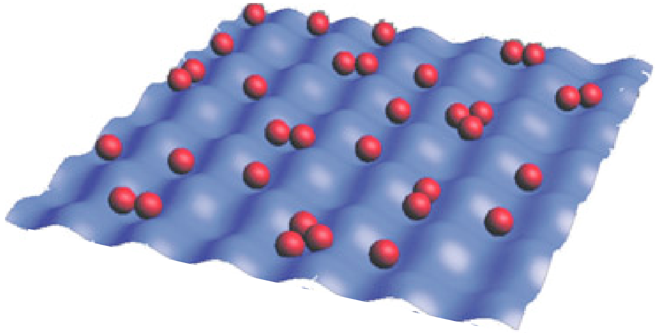
July 9, 2014
Cetraro

Our bet: quantum devices

Quantum randomness

Quantum communication

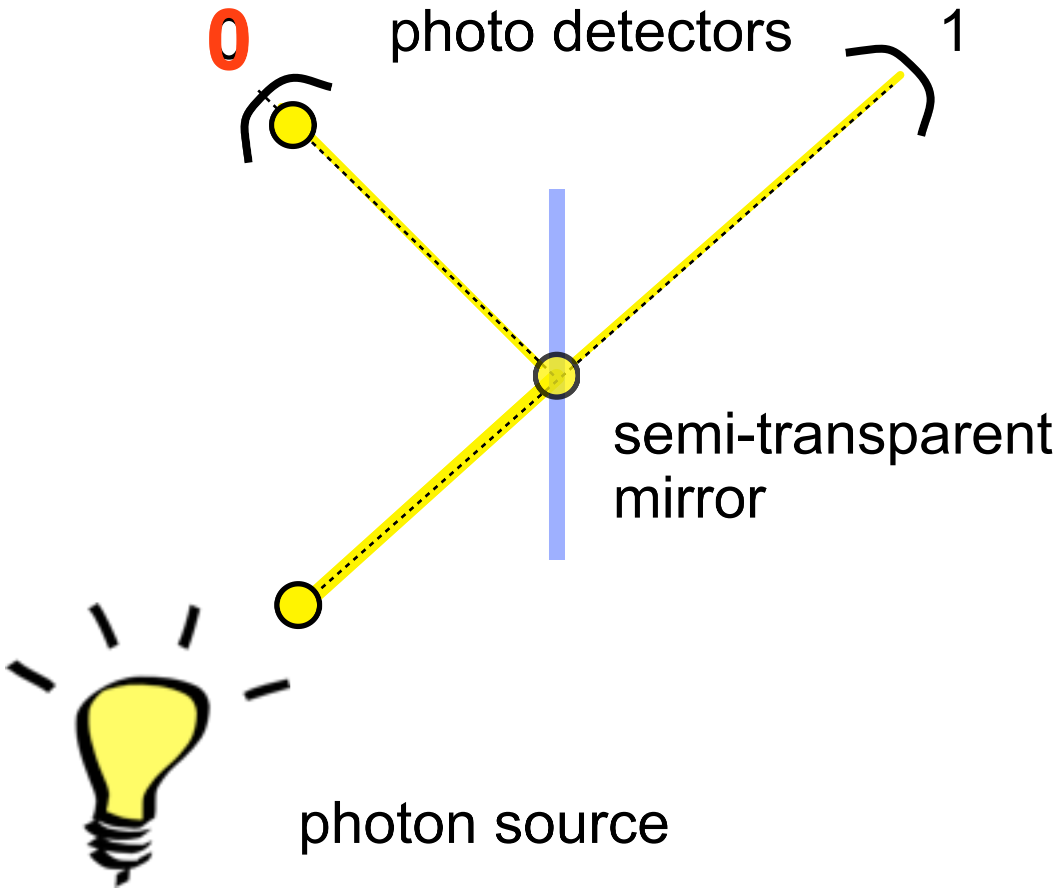
Quantum computing



Quantum simulation

Quantum optimization(?)

True and perfect randomness



1. Photon source emits a photon
2. Photon hits semi-transparent mirror
3. Photon follows both paths
4. The photo detectors see the photon only in one place: **a random bit**

The quantum bit (qubit)

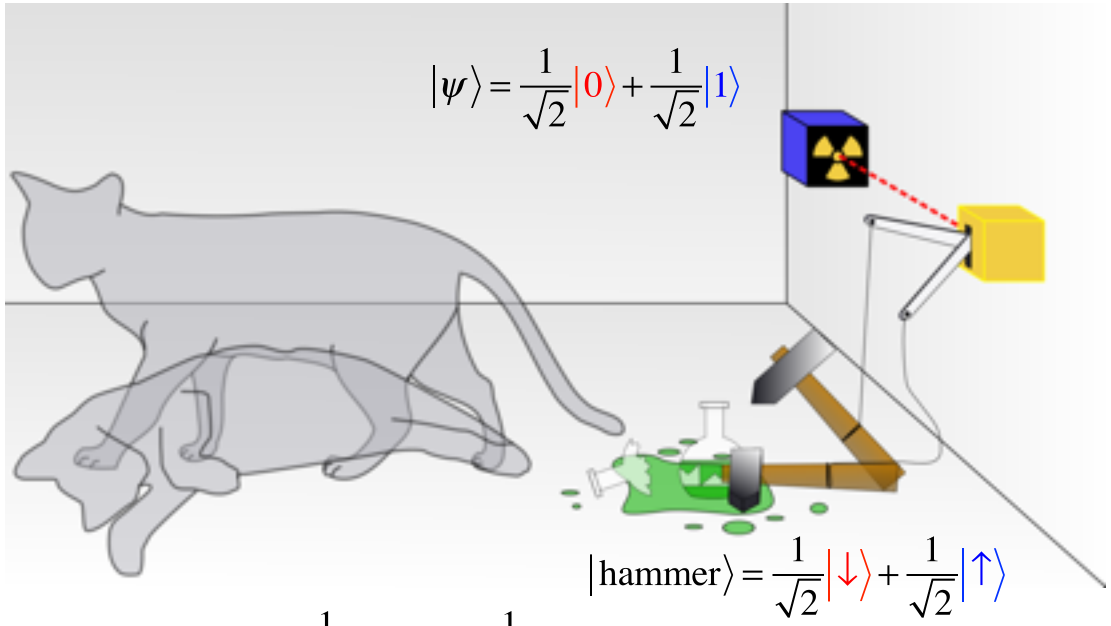
Schrödinger's cat paradoxon

Classical bits can be $|0\rangle$ or $|1\rangle$

Qubits can be both at once

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

"quantum superposition"



$$|\text{glass}\rangle = \frac{1}{\sqrt{2}}|\text{open}\rangle + \frac{1}{\sqrt{2}}|\text{closed}\rangle$$

$$|\text{cat}\rangle = \frac{1}{\sqrt{2}}|\text{dead}\rangle + \frac{1}{\sqrt{2}}|\text{alive}\rangle$$

Measuring a quantum superposition

- when measuring (looking) we only ever get one classical bit: 0 or 1

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \longrightarrow \begin{array}{ll} 0 & \text{with probability } |\alpha|^2 \\ 1 & \text{with probability } |\beta|^2 \end{array}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

- When we look the cat is always either dead or alive!

- Quantum random number generator:

  prepare and the state $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and measure

An application for a 1-qubit quantum computer!

The incomprehensible magic of “quantum entanglement”

A single qubit gives a random bit when measured

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle + |1\rangle\right]$$

“Entangled states” can give random but identical results

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B\right]$$

Measuring qubit *A* gives a random result *a*
Measuring qubit *B* gives a random result *b*

However, always *a*=*b* no matter how far apart the qubits are

A shared secret key that an be made provably secure!

A serious restriction: no-cloning theorem

$$C|0\rangle \to |0\rangle|0\rangle$$
$$C|1\rangle \to |1\rangle|1\rangle$$

~~$C|\psi\rangle \to C|\psi\rangle|\psi\rangle$~~

A quantum state cannot be copied!

Bad news for quantum programmers

Excellent news for cryptographers

Information content of a quantum register

A 2-qubit register

needs four complex numbers to be represented

but when measured only gives two bits of information

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

An *N* qubit register

needs 2^N complex numbers to be represented

but when measured only gives *N* bit of information

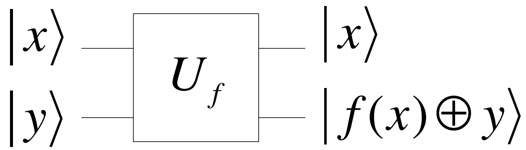
$$|\psi\rangle = \sum_{i_1, i_2, \ldots, i_N} \alpha_{i_1 i_2 \ldots i_N} |i_1 i_2 \ldots i_N\rangle$$

Exponential intrinsic parallelism: operate on 2^N inputs at once

But very limited readout of only *N* bits

Calculating in superposition

Quantum computers can work on all possible inputs in superposition

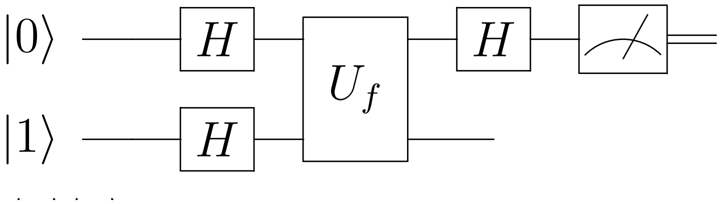


$$U_f|x\rangle|y\rangle \to |x\rangle|f(x)\oplus y\rangle$$

$$U_f\left(\alpha|0\rangle+\beta|1\rangle\right)|0\rangle \to \alpha|0\rangle|f(0)\rangle+\beta|1\rangle|f(1)\rangle$$

Measuring the result one only gets **either** f(0) **or** f(1), chosen **randomly**!

Smartly compute one global result based on all inputs and measure it!



$$f(0)\oplus f(1)$$

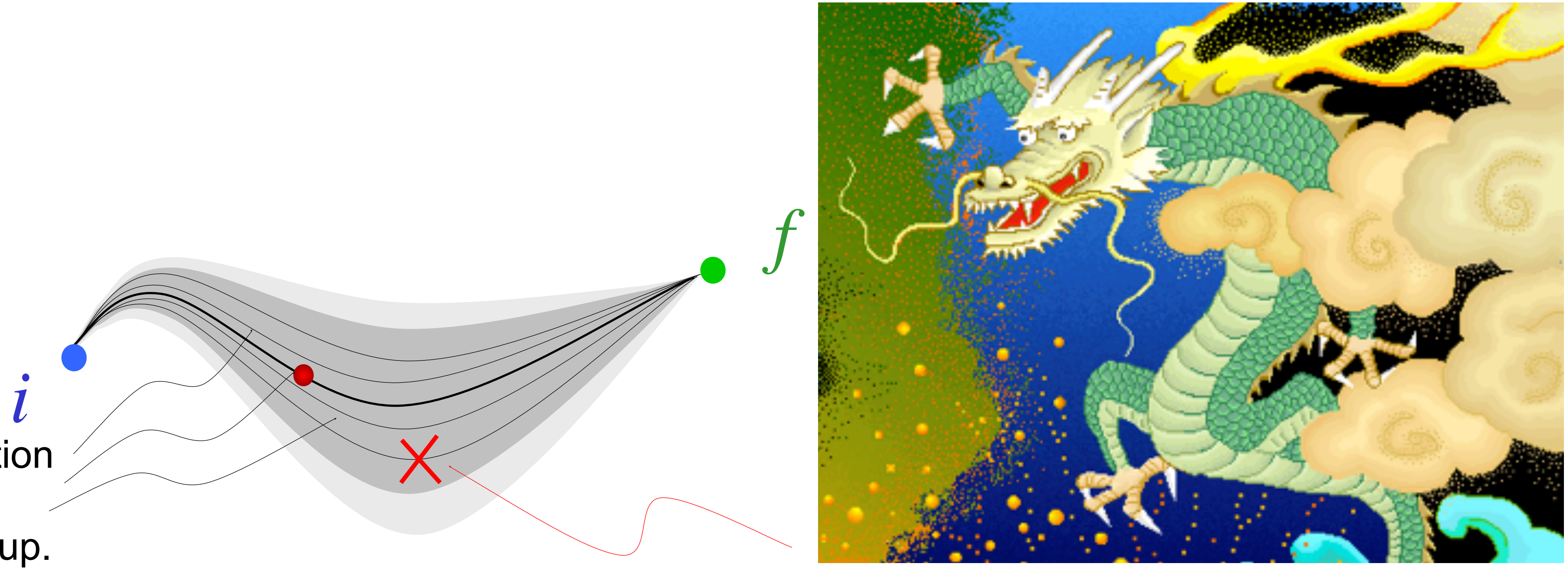
Determine whether f(0)=f(1)
with one function call

(Deutsch&Jozsa, 1992)

Interlude: quantum hardware

Observing the cat made it be either dead or alive!

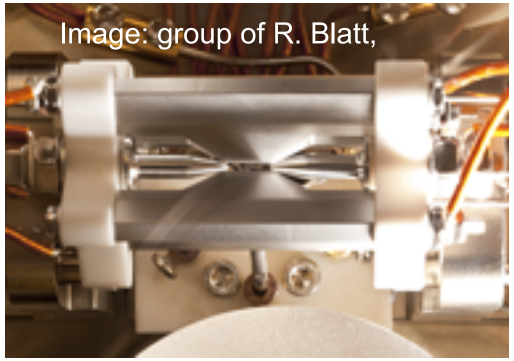
Qubits need to be well isolated from the environment!



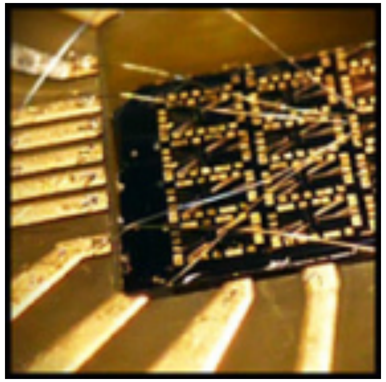
"Parallel" evolution providing quantum speedup.

Perturbations from the environment destroy the "parallel" quantum evolution of the computation

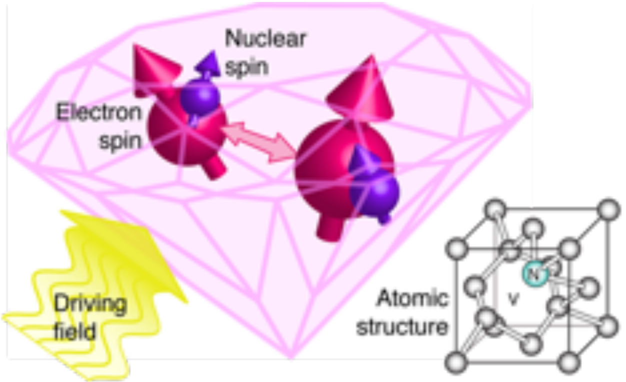
Many different platforms



trapped ions
20 qubits
(R. Blatt, Innsbruck)

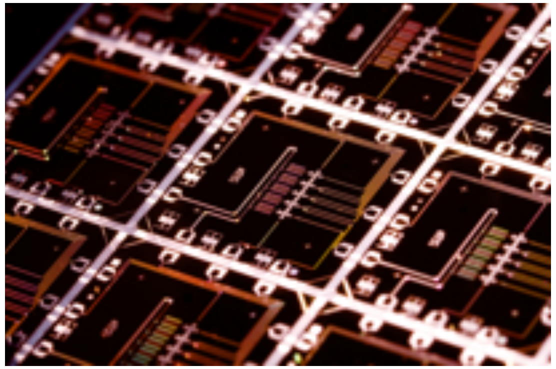


quantum dots
(C. Marcus, Copenhagen)

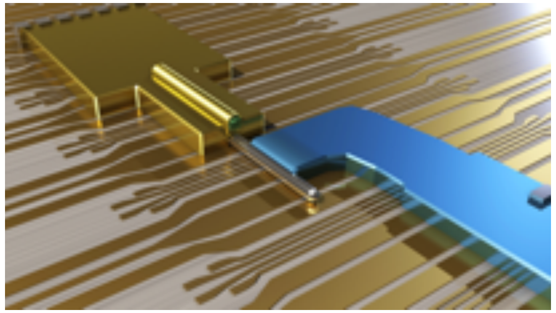


defects in diamond

100 gate operations on 20 qubits



superconductors
9 qubits
(J. Martinis, UCSB)



Topological quantum bits
(L. Kouwenhoven, Delft)

Simulating quantum computers on classical computers

Simulating a quantum gate acting on N qubits needs $O(2^N)$ memory and operations

Qubits	Memory	Time for one gate operation
10	16 kByte	microseconds on a watch
20	16 MByte	milliseconds on smartphone
30	16 GByte	seconds on laptop
40	16 TByte	minutes on supercomputer
50	16 PByte	hours on top supercomputer
60	16 EByte	long long time
80	size of visible universe	age of the universe

Why should we build a quantum computer?

Simply because we can!

Somebody smart will figure out a use!

These arguments are not enough to justfy the money it will cost

Quantum computing beyond exa-scale

What are the important applications …

… that we can solve on a quantum computer …

… but not special purpose post-exa-scale classical hardware that we may build in ten years?

Microsoft IBM LOCKHEED MARTIN Google NOKIA intel

What problems do we want to solve on a quantum computer?

- design better drugs
- cure cancer
- counter climate change
- fold proteins
- fight hunger
- design better batteries
- optimize hard problems
- eradicate diseases
- realize artificial intelligence

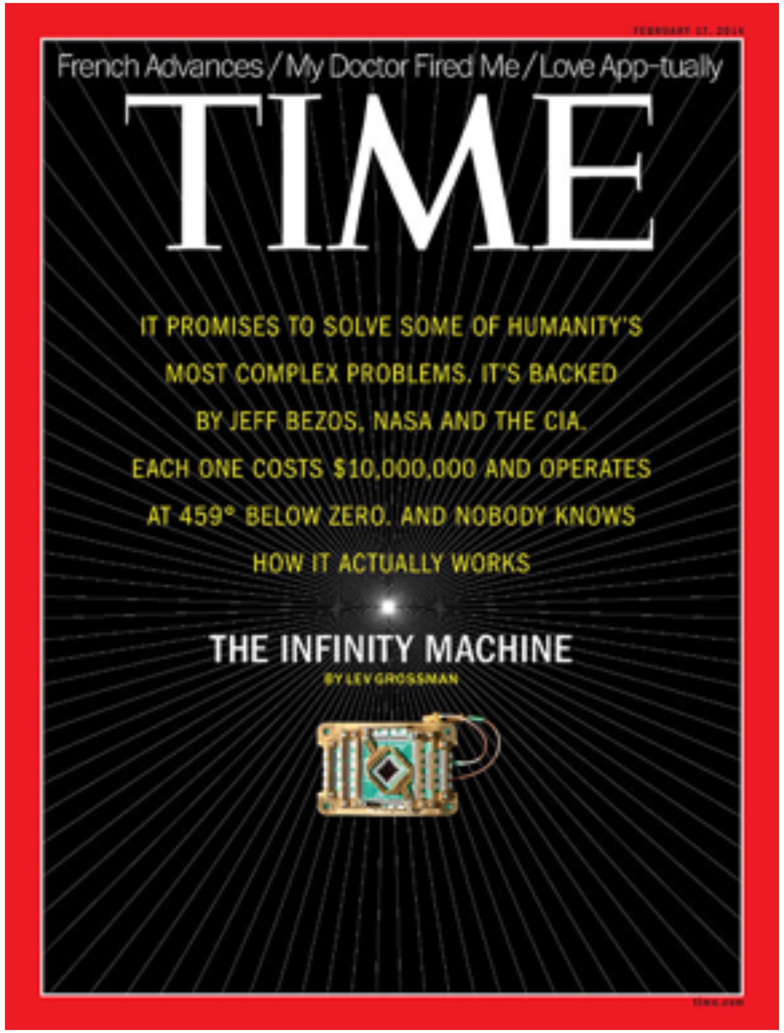
What problems do we want to solve on a quantum computer?

- design better drugs
- cure cancer
- counter climate change
- fold proteins
- fight hunger
- design better batteries
- optimize hard problems
- eradicate diseases
- realize artificial intelligence

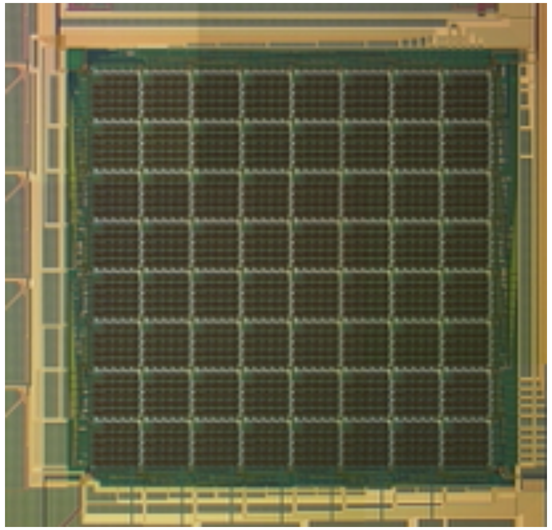
This is a list for a quantum wishing well

Which of these can actually profit from quantum computers?

A quantum machine to solve hard optimization problems



The D-Wave quantum annealer



A device to solve quadratic binary optimization problems

$$C(x_1,\dots,x_N) = \sum_{ij} a_{ij} x_i x_j + \sum_i b_i x_i$$

$$\text{with} \qquad x_i = 0,1$$

Can be built with imperfect qubits

Significant engineering achievement to scale it to one thousand qubits

Nobody knows if it can solve NP-hard problems better than a classical computer

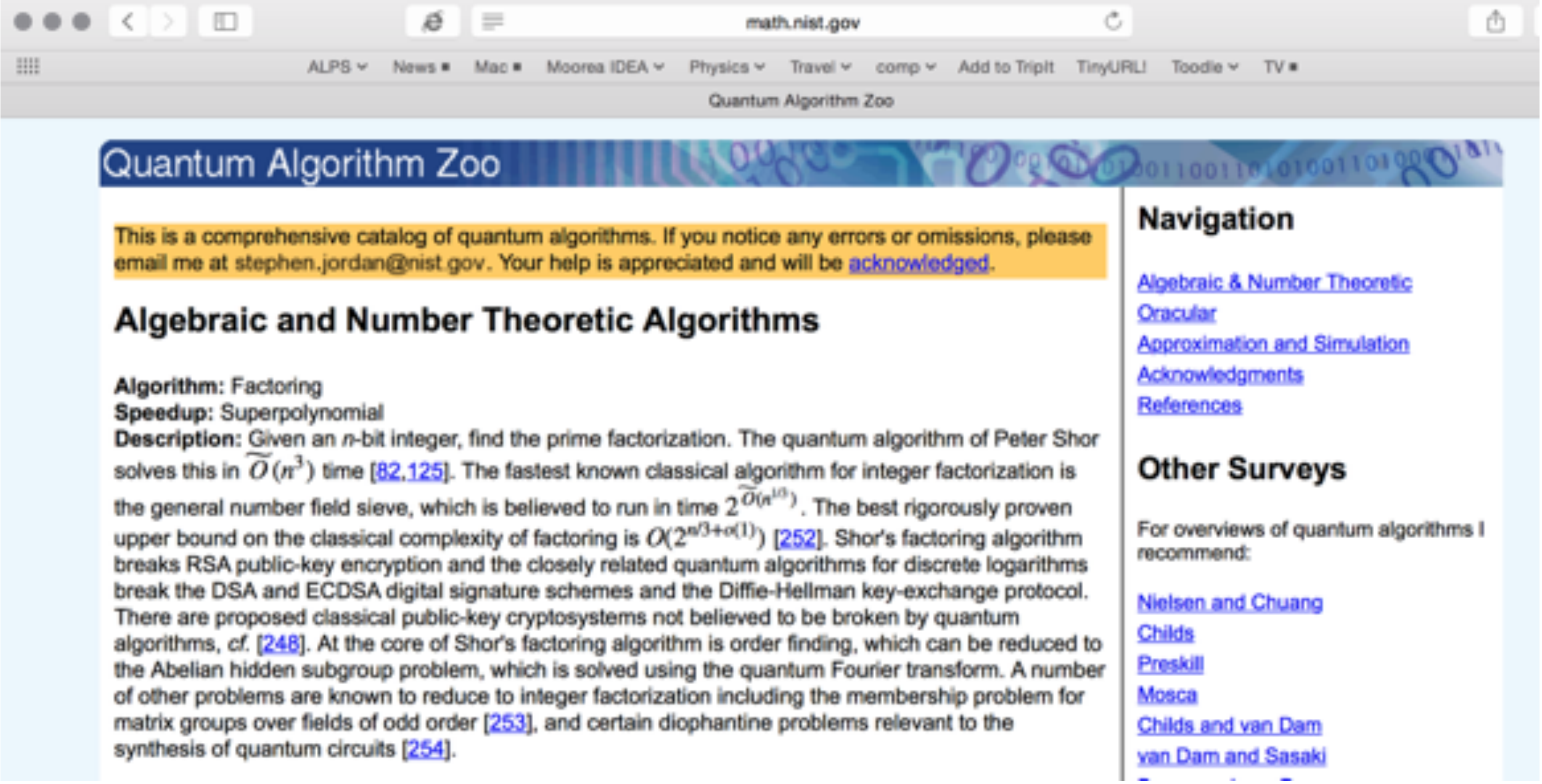
So far no scaling advantage has been observed

Better look at algorithms with known quantum speedup

50+ quantum algorithms with known speedup

Can we use any of them in real-world applications?

http://math.nist.gov/quantum/zoo/


math.nist.gov

ALPS News Mac Moorea IDEA Physics Travel comp Add to Triplt TinyURL! Toodle TV

Quantum Algorithm Zoo

Quantum Algorithm Zoo

This is a comprehensive catalog of quantum algorithms. If you notice any errors or omissions, please email me at stephen.jordan@nist.gov. Your help is appreciated and will be acknowledged.

Algebraic and Number Theoretic Algorithms

Algorithm: Factoring
Speedup: Superpolynomial
Description: Given an n-bit integer, find the prime factorization. The quantum algorithm of Peter Shor solves this in $\widetilde{O}(n^3)$ time [82,125]. The fastest known classical algorithm for integer factorization is the general number field sieve, which is believed to run in time $2^{\widetilde{O}(n^{1/3})}$. The best rigorously proven upper bound on the classical complexity of factoring is $O(2^{n/3+o(1)})$ [252]. Shor's factoring algorithm breaks RSA public-key encryption and the closely related quantum algorithms for discrete logarithms break the DSA and ECDSA digital signature schemes and the Diffie-Hellman key-exchange protocol. There are proposed classical public-key cryptosystems not believed to be broken by quantum algorithms, cf. [248]. At the core of Shor's factoring algorithm is order finding, which can be reduced to the Abelian hidden subgroup problem, which is solved using the quantum Fourier transform. A number of other problems are known to reduce to integer factorization including the membership problem for matrix groups over fields of odd order [253], and certain diophantine problems relevant to the synthesis of quantum circuits [254].

Navigation

Algebraic & Number Theoretic
Oracular
Approximation and Simulation
Acknowledgments
References

Other Surveys

For overviews of quantum algorithms I recommend:

Nielsen and Chuang
Childs
Preskill
Mosca
Childs and van Dam
van Dam and Sasaki

Shor’s algorithm for factoring

Factoring small numbers is easy: 15 = 3 x 5

Factoring large numbers is hard classically: O(exp($N^{1/3}$)) time for N digit-numbers

536939683642691194607950541533260051860418183893023116620231731884706135841697779
81247775554355964649044526158042091770292405381561410352725541976253778624830290
518096150501270434149272610204114236496946309670910771714302797950221151202416796
22849447805650987368350247829683054309216276674509735105639240298977591783205062
1619158848593319454766098482875128834780988979751083723214381986678381350567167

=

4363637625931498167701061252972058930130370651588109946621952523434903606572651613287
3421237667900245913537253744354928238018040554845306796065865605354860834270732796989
4210413710440109013191728001673

*

1230486419064350262435007521990111788816176581586683476039159532309509792696707176253
0052007668467350605879541695798973080376300970096911310297914332946223591672260748684
8670728527914505738619291595079

Polynomial time on a quantum computer (P. Shor)

Breaking RSA encryption with Shor's algorithm?

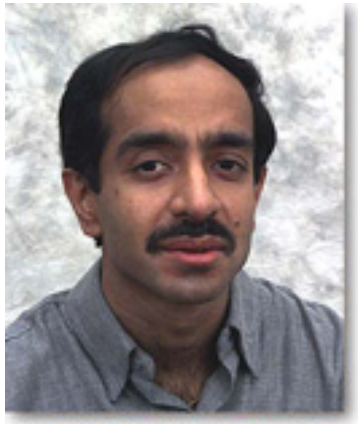
RSA	cracked in	CPU years	Shor
453 bits	1999	10	1 hour
768 bits	2009	2000	5 hours
1024 bits		1000000	10 hours

estimates based on 10 ns gate time
and minimal number of $2N+3$ qubits

Not a long-term "killer-app" since we can switch to post-quantum encryption

- quantum cryptography
- post-quantum encryption (e.g. lattice based cryptography)

Grover search



Search an unsorted database of N entries with $\sqrt{N}$ queries

However, the query needs to be implemented!

- Querying an N-entry database needs at least O(N) hardware resources
- Can perform the query classically in log(N) time given O(N) resources

Only useful if the query result can be efficiently calculated on the fly!

What are the important applications satisfying this criterion?

International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

Received May 7, 1981

Feynman invented quantum computers to simulate quantum physics

We can surpass the best classical computers with only 50 qubits!

This will make physicists happy
but is it enough to motivate

Microsoft IBM intel

to to build one?

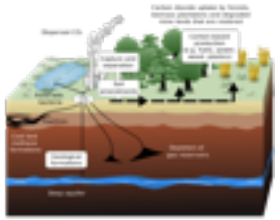
First applications that reached a petaflop on Jaguar @ ORNL

Domain area	Code name	Institution	# of cores	Performance	Notes
Materials	DCA++	ORNL	213,120	**1.9 PF**	2008 Gordon Bell Prize Winner
Materials	WL-LSMS	ORNL/ETH	223,232	**1.8 PF**	2009 Gordon Bell Prize Winner
Chemistry	NWChem	PNNL/ORNL	224,196	1.4 PF	2008 Gordon Bell Prize Finalist
Materials	DRC	ETH/UTK	186,624	1.3 PF	2010 Gordon Bell Prize Hon. Mention
Nanoscience	OMEN	Duke	222,720	> 1 PF	2010 Gordon Bell Prize Finalist
Biomedical	MoBo	GaTech	196,608	780 TF	2010 Gordon Bell Prize Winner
Chemistry	MADNESS	UT/ORNL	140,000	550 TF	
Materials	LS3DF	LBL	147,456	442 TF	2008 Gordon Bell Prize Winner
Seismology	SPECFEM3D	USA (multiple)	149,784	165 TF	2008 Gordon Bell Prize Finalist

Simulating quantum materials on a quantum computer

Can we use quantum computers to design new quantum materials?

- A room-temperature superconductor?
- Non-toxic designer pigments?
- A catalyst for carbon sequestration?
- Better catalysts for nitrogen fixation (fertilizer)?



Solving many materials challenges has

- exponentially complexity on classical hardware
- polynomial complexity on quantum hardware!

Can quantum chemistry be performed on a small quantum computer?

Dave Wecker,[1] Bela Bauer,[2] Bryan K. Clark,[2,3] Matthew B. Hastings,[2,1] and Matthias Troyer[4]

[1] *Quantum Architectures and Computation Group, Microsoft Research, Redmond, WA 98052, USA*
[2] *Station Q, Microsoft Research, Santa Barbara, CA 93106-6105, USA*
[3] *Kavli Institute for Theoretical Physics, University of California, Santa Barbara, CA 93106, USA*
[4] *Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland*

Phys. Rev. A **90**, 022305 (2014)

Can a classically-intractable problem be solved on a small quantum computer?

Can a classically-intractable problem be solved on a huge quantum computer?

Can a classically-intractable problem be solved on the largest imaginable quantum computer?

Simulating a quantum system on quantum computers

There are O(N^4) interaction terms in an N-electron system

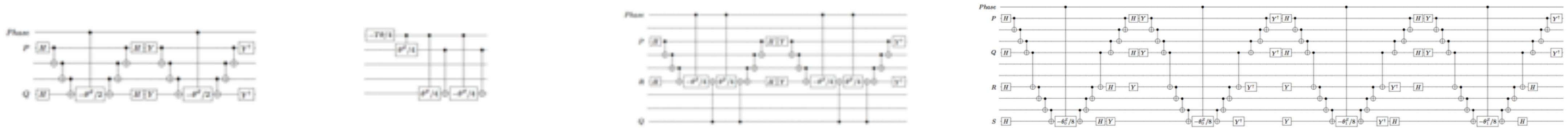
$$H = \sum_{pq} t_{pq} c_p^\dagger c_q + \sum_{pqrs} V_{pqrs} c_p^\dagger c_q^\dagger c_r c_r \equiv \sum_{m=1}^{M} H_m$$

$M = O(N^4)$ terms

We need to evolve separately under each of them

$$e^{-i\Delta t H} \approx \prod_{m=1}^{M} e^{-i\Delta\tau H_m}$$

Efficient circuits available for each of the N^4 terms



Runtime estimates turn out to be O(NM^2) = $O(N^9)$

It's efficient since it's polynomial!

Really?

The polynomial time quantum shock

- Estimates for an example molecule: Fe_2S_2 with 118 spin-orbitals

Gate count	10^{18}
Parallel circuit depth	10^{17}
Run time @ 10ns gate time	30 years

Quantum information theorists declare victory
proving the existence of polynomial time algorithms

We need quantum software engineers to develop
better algorithms and implementations

The result of quantum software optimization

- Estimates for an example molecule: Fe_2S_2 with 118 spin-orbitals

Gate count	10^{18}
Parallel circuit depth	10^{17}
Run time @ 10ns gate time	30 years

Reduced gate count	10^{11}
Parallel circuit depth	10^{10}
Run time @ 10ns gate time	2 minutes

- Attempting to reduce the horrendous runtime estimates we achieved
  Wecker *et al.*, PRA (2014), Hastings *et al.*, QIC (2015), Poulin *et al.*, QIC (2015)
  - Reuse of computations: O(*N*) reduction in gates
  - Parallelization of terms: O(*N*) reduction in circuit depth
  - Optimizing circuits: 4x reduction in gates
  - Smart interleaving of terms: 10x reduction in time steps
  - Multi-resolution time evolution: 10x reduction in gates
  - Better phase estimation algorithms: 4x reduction in rotation gates

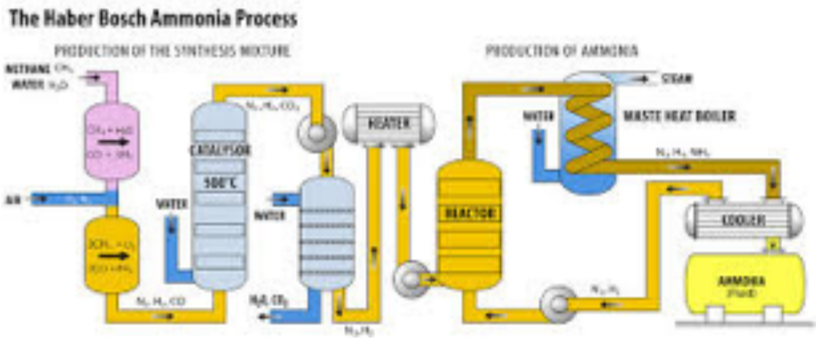
Nitrogen fixation: a potential killer-app

Fertilizer production using Haber-Bosch process (1909)

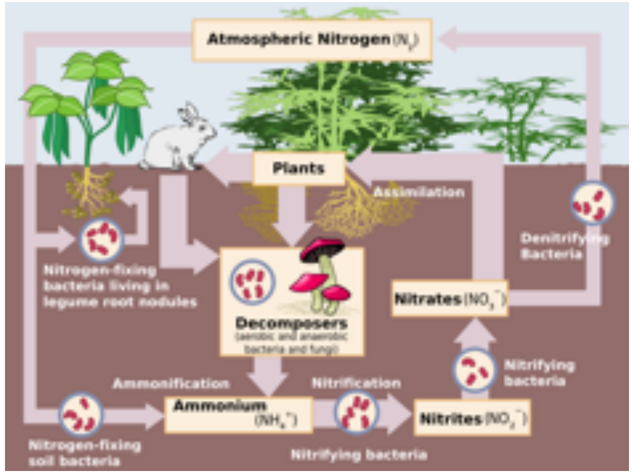
Requires high pressures and temperatures

3-5% of the world's natural gas

1-2% of the world's annual energy



But bacteria can do it cheaply at room temperature!



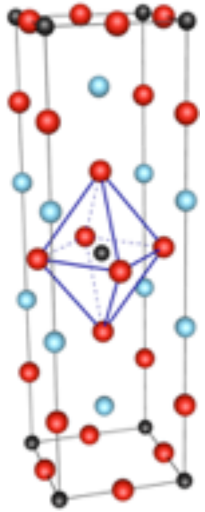
Quantum solution using about 400 qubits

- Understand how bacteria manage to turn air into ammonia
- Design a catalyst to enable inexpensive fertilizer production

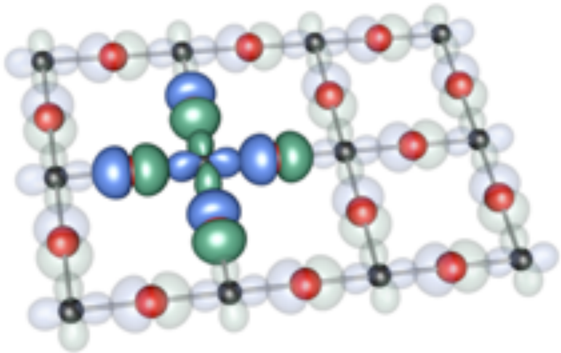
What about a high temperature superconductor?

Orbitals per unit cell	≈ 50
Unit cells needed	20 x 20
Number of orbitals	$N \approx$ 20’000
Number of terms	N^4
Scaling of algorithm	$O(N^{5.5})$
Estimated runtime	age of the universe

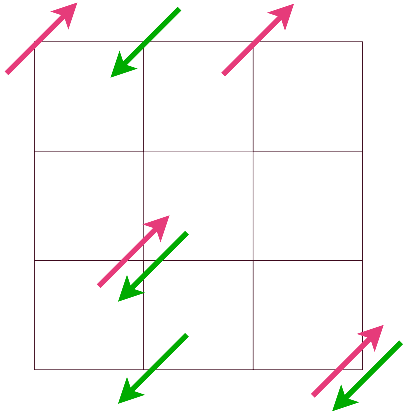
Reduction to a simplified model



3D crystal structure



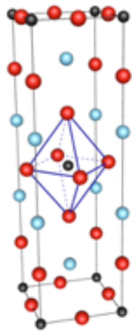
single 2D layer

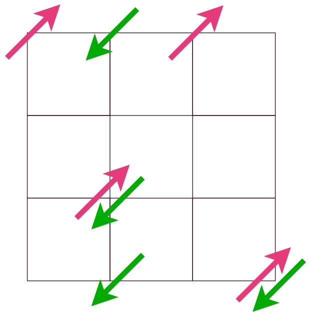


simplified model

From materials to models on quantum computers

	Material	Model
Orbitals per unit cell	≈ 50	1
Unit cells needed	20x20	20x20
Number of orbitals	$N \approx 20'000$	$N \approx 800$
Number of terms	N^4	$O(N)$
Scaling of algorithm	$O(N^{5.5})$	$O(N^{0.5})$
Estimated runtime	age of the universe	seconds





Hybrid quantum classical approaches

Classical algorithms

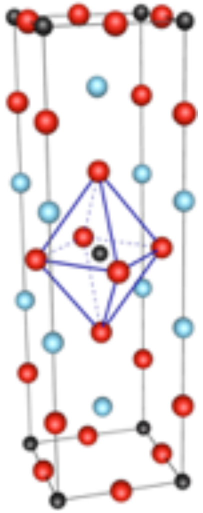
realistic but approximate material simulation

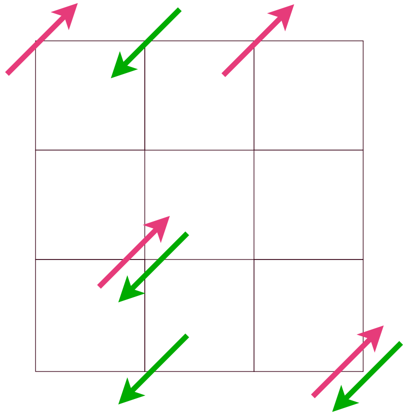
→ extract simple model →

← improve classic simulation ←

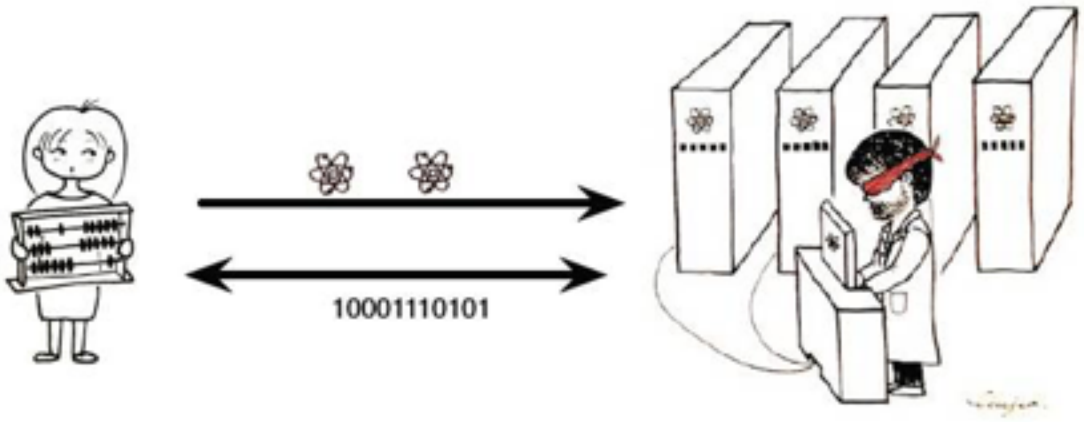
Quantum algorithm

accurately solve simplified model





There is much more!



Blind quantum computing and search
(Broadbent, Fitzsimons, Kashefi)

Quantum money
(Aaronson, Farhi *et al*)

Cloud provides **cannot** know what the user does

What will we do with a quantum computer?

True random numbers with just one qubit

Secure communication with just a few qubits

Interesting real-world applications for a quantum computer

- Breaking of RSA encryption (?)
- Design of catalysts and materials
- Provably secure cloud computing

We need quantum software engineers to explore more potential applications!

The quantum algorithms team


Dave Wecker


Matt Hastings


Nathan Wiebe

(Microsoft Research Redmond)


Bela Bauer


Chetan Nayak

(Microsoft Station Q Santa Barbara)


David Poulin
(Sherbrooke)


Andrew Doherty
(Sydney)


Bryan Clark
(UIUC)


Andy Millis
(Columbia)