Quantum Computational Science

Itay Hen

2014 OLCF Users Meeting
Oak Ridge, TN
July 24, 2014

Information Sciences Institute
USC School of Engineering

Collaborators

- Daniel Lidar,
  Federico Spedalieri,
  Rosa Di Felice,
  Tameem Albash, USC.
- Travis Humble,
  Bobby Sumpter, ORNL.
- Greg Tallant, Lockheed-Martin.
- Matthias Troyer, ETH Zurich.





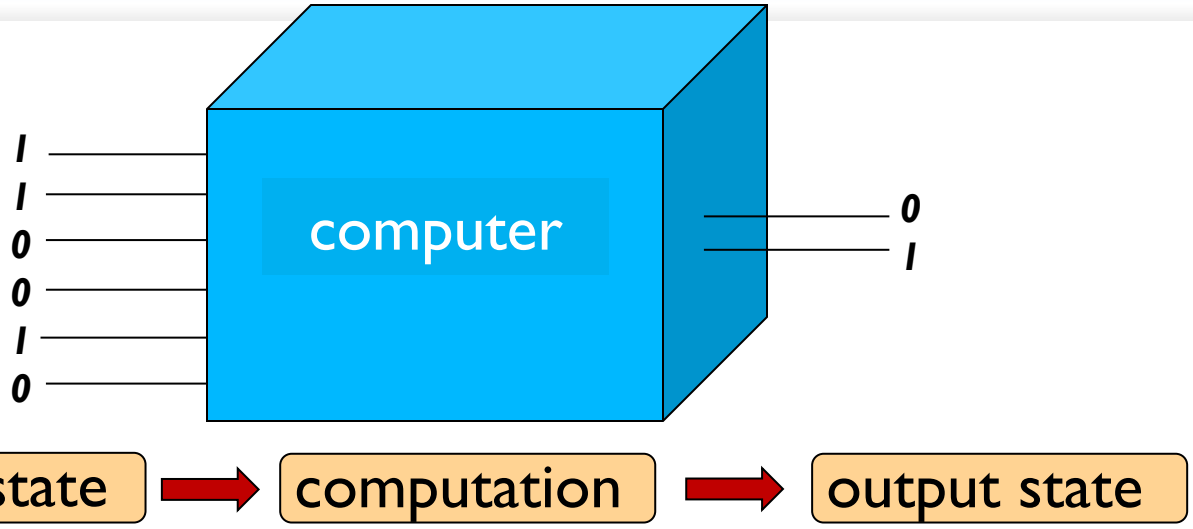



Outline

- introduction:
  - what is a quantum computer?
  - adiabatic quantum computing
- the D-Wave chip: a machine worth studying
- simulating quantum annealers:
  understanding and improving quantum computers
- benchmarking the D-Wave chip:
  determining the usefulness of quantum annealers
- conclusions and outlook

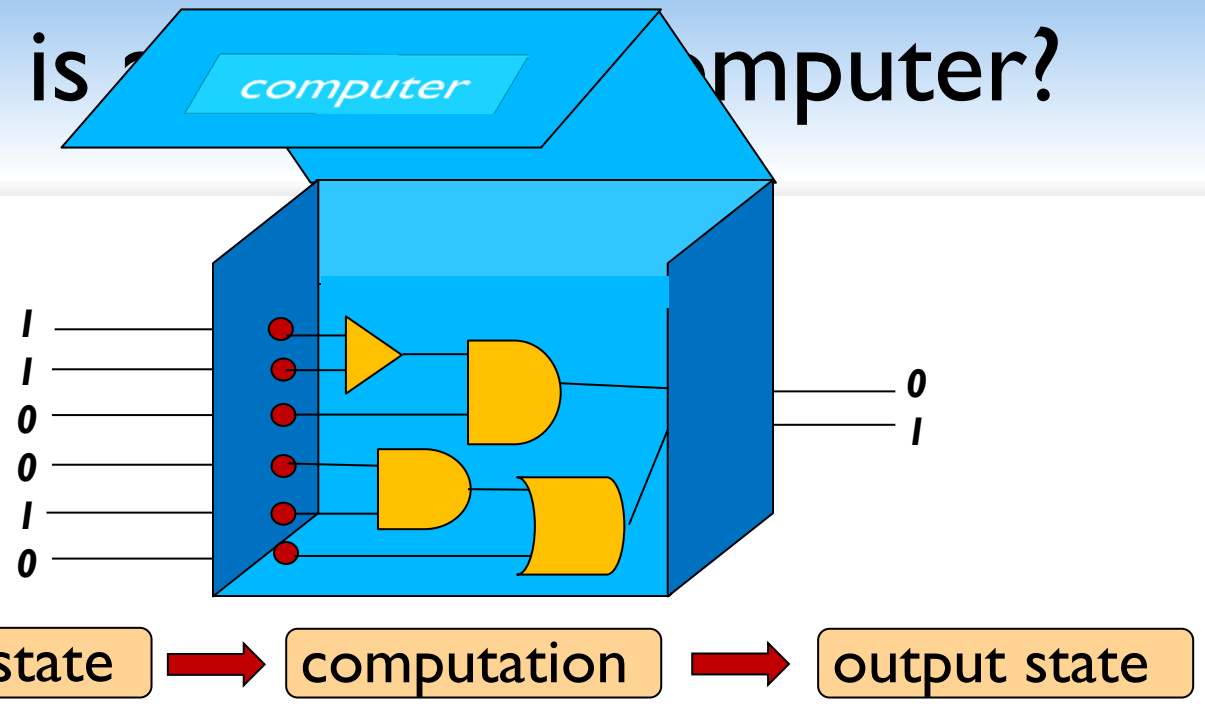
Introduction

What is a quantum computer?



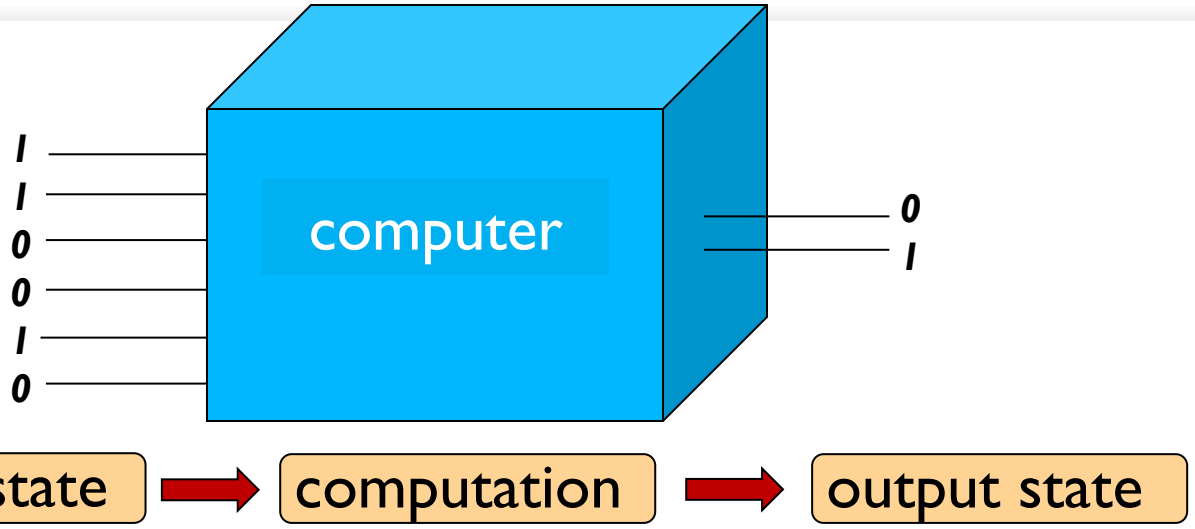
- both classical and quantum computers may be viewed as machines that perform computations on given inputs and produce outputs. inputs and outputs are strings of bits (0's and 1's).
- classical computers are based on manipulations of bits. bits are are equivalent to Ising spins (−1's and 1's).

What is a computer?



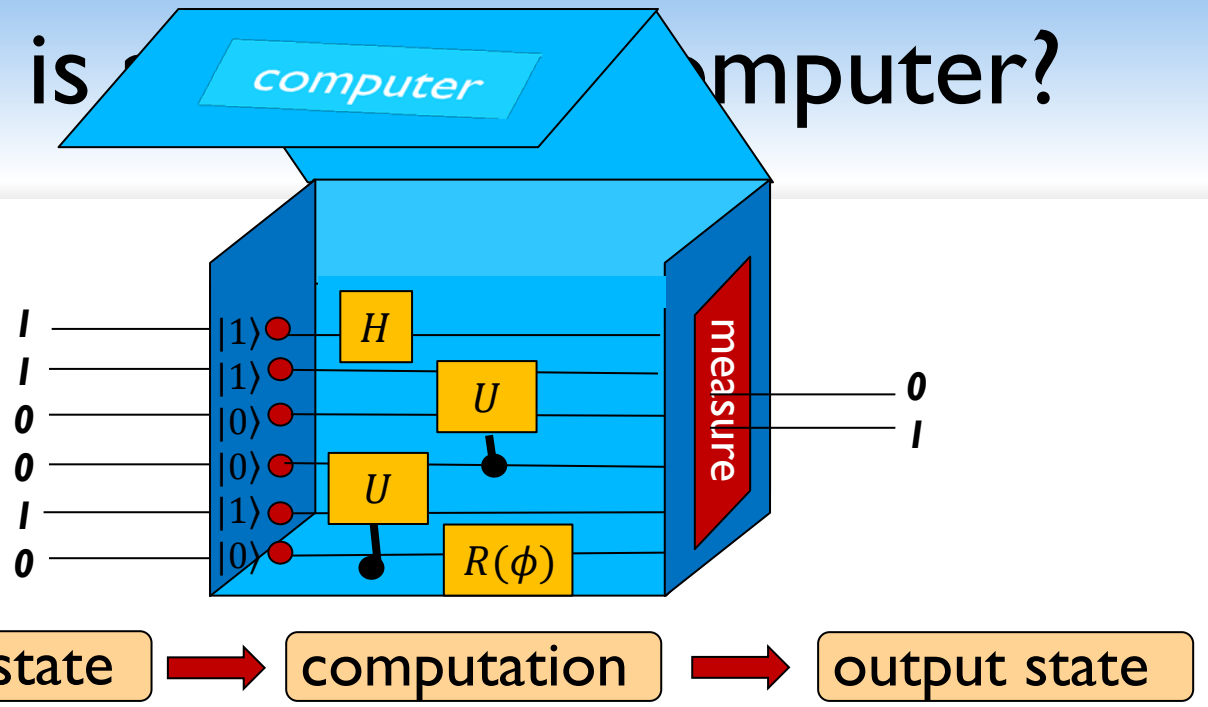
- in classical computers, computation is usually done by applying local gates to the state of the system.
- these gates (AND, OR, NOT, XOR,…) advance the state of the system until the result of the desired computation is achieved.
- at any given time, the system is in a unique classical configuration (i.e., in a state that is a string of 0's and 1's).

What is a quantum computer?



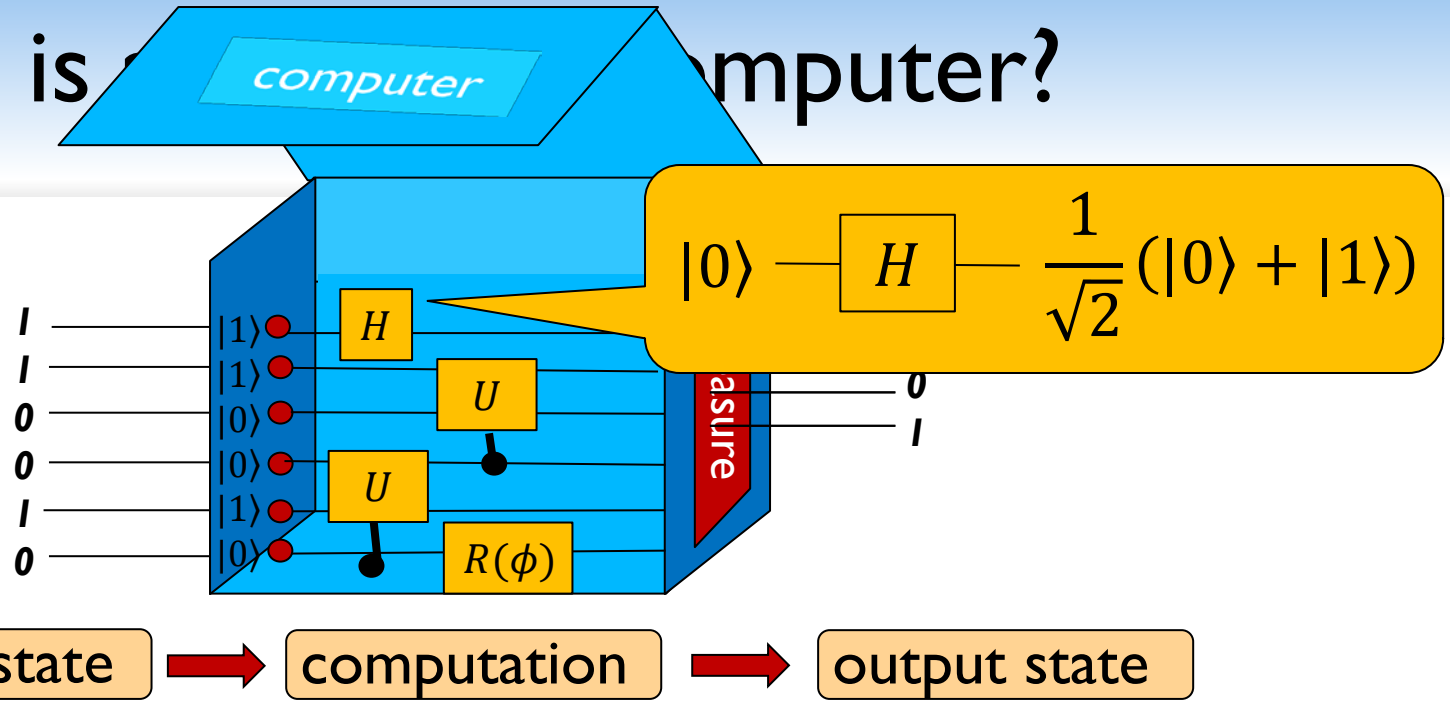
- quantum computers on the other hand manipulate quantum bits (or qubits for short) – equivalent to spin-1/2 particles.
- qubits can be in a superposition of up and down for example.
- the range of intermediate states that a quantum computer can be in and the range of operations it can perform are much larger than those of a classical computer (superposition, entanglement, etc.).

What is a quantum computer?



- much like classical gates, there are also quantum gates (unitary operators) with which one can construct quantum circuits.
- the Hadamard gate gives us superpositions. other gates entangle.
- advantage: these unique properties can be utilized towards faster computations.
- disadvantage: quantum computers don't really exist.
  many technological difficulties. noise/interaction with environment.

What is a quantum computer?



- much like classical gates, there are also quantum gates (unitary operators) with which one can construct quantum circuits.
- the Hadamard gate gives us superpositions. other gates entangle.
- advantage: these unique properties can be utilized towards faster computations.
- disadvantage: quantum computers don't really exist.
  many technological difficulties. noise/interaction with environment

Introduction

turns out that at least in theory quantum computers are faster than classical computers

- best-known example is

  Shor's algorithm for integer factorization:

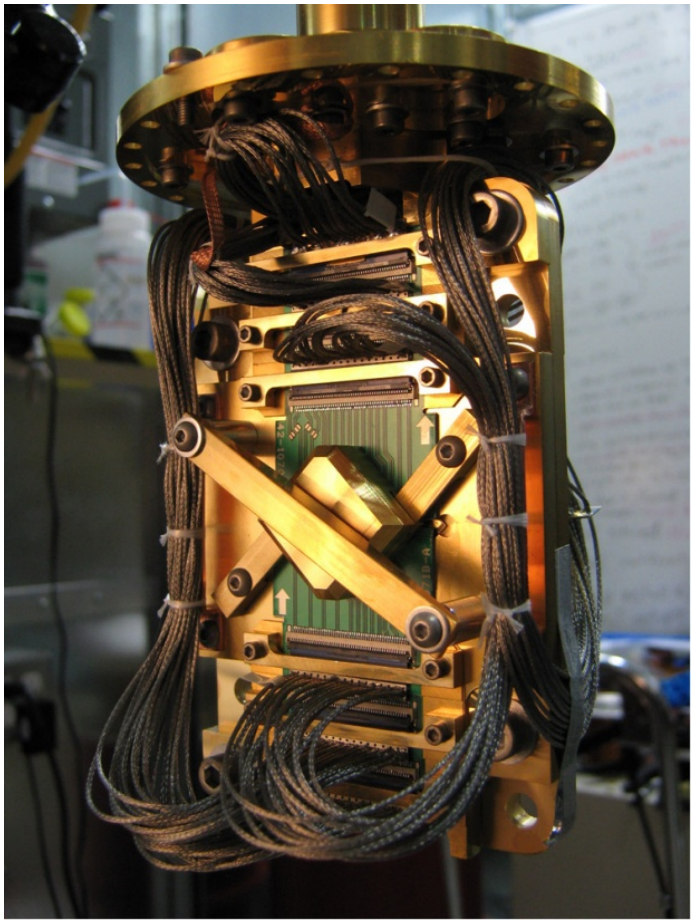
  - solves the problem in polynomial time: exponential speedup.
  - even super-computers will not be able to compete against a working quantum computer!
  - of practical importance (RSA code breaking).
  - current quantum computers can factor integers up to *21*.
  - there may be a classical algorithm that is just as fast (or even faster) but that has not yet been found.

Adiabatic Quantum Computing

- quantum computation does not have to be done via gates.
- there are other ways to quantum-compute, that may be more practical.
- one such way is Adiabatic Quantum Computing (AQC).

AQC is a general approach to solve a broad range of hard optimization problems using "quantum annealing"
[Farhi et al.,2001]

- USC's ISI hosts one such machine "D-Wave Two" (only two available).



Adiabatic Quantum Computing

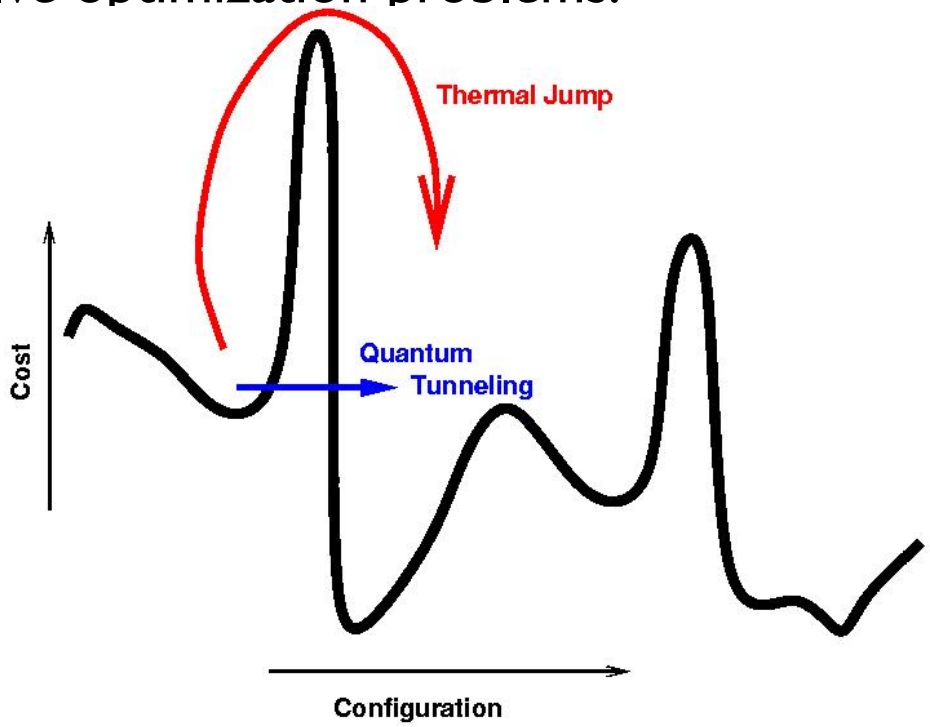
- the general mechanism of AQC:

1. take a difficult (classical) optimization problem, generically: find minimizing configuration of the cost function $E = f(x_1, x_2, \ldots, x_n)$ where x_i are binary variables $(0,1)$ and $f(\cdots)$ is given.
2. encode the problem in a "problem" Hamiltonian, $\widehat{H}^{(p)}$, such that its ground state encodes the solution.
3. prepare the system in the ground state of another, easily solvable, "beginning (driver) Hamiltonian" $\widehat{H}^{(b)}$.
4. vary the Hamiltonian slowly and smoothly from $\widehat{H}^{(b)}$ to $\widehat{H}^{(p)}$ until ground state of $\widehat{H}^{(p)}$ is reached.

The D-Wave chip: a machine worth studying

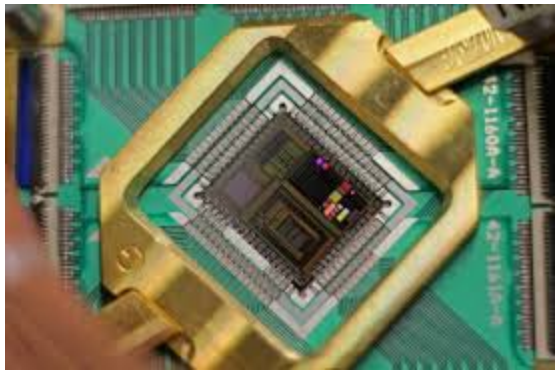
The D-Wave Two Chip

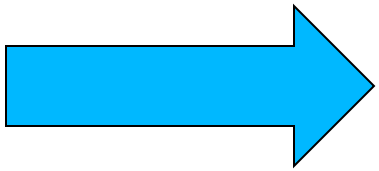
- D-Wave Two is a Quantum Annealer. based on a continuous interpolation between an easy problem we know the solution to and a hard problem whose solution we'd like to find.
- (presumably) uses the advantages of Quantum Mechanics (such as tunneling and entanglement) to solve optimization problems.
- the quantum features of the system are not available to classical traditional machines.
- therefore, there is the hope that this machine (and quantum computers in general) could solve certain problems faster than any classical device.



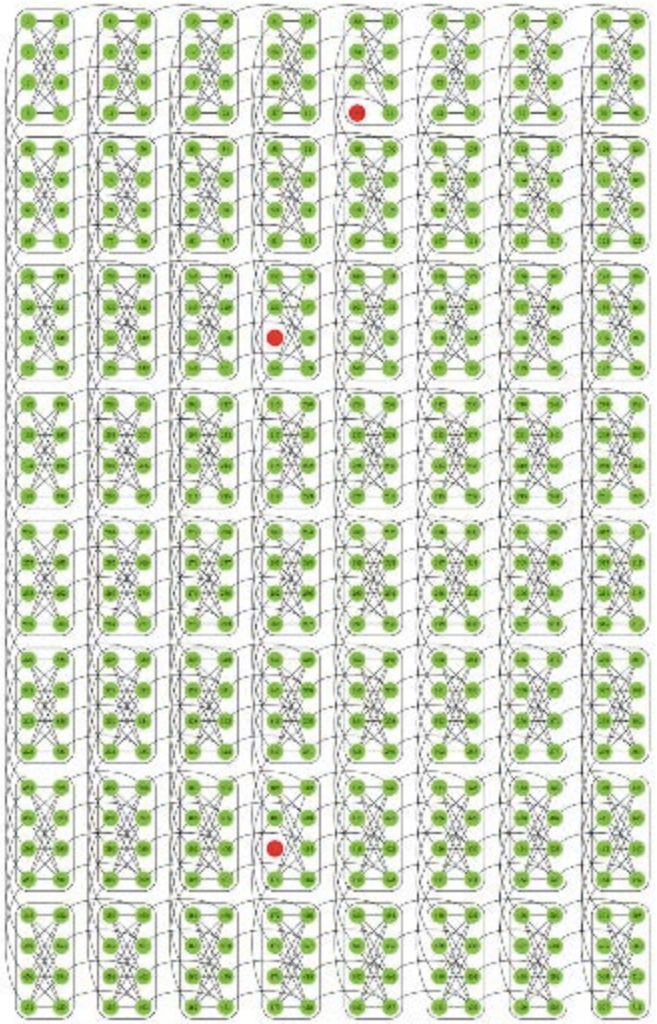
The D-Wave Two Chip

- D-Wave Two is designed to solve optimization problems of very specific type and form.





the Chimera architecture



- architecture of the chip and physical constraints are very limiting.
- fortunately, within the limitations there is enough room for solving non-trivial problems.

The D-Wave Two Chip

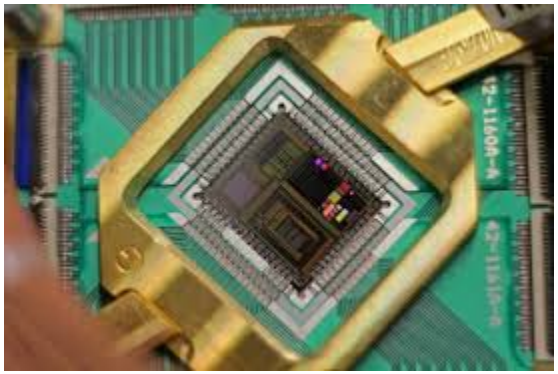
- the D-Wave chip finds (or attempts to find) global minima of cost functions of the Ising type (equivalent to QUBO problems):

$$H = \sum_{\langle i,j \rangle} J_{ij} S_i S_j + \sum_i h_i S_i$$

- the S_i's are Ising spin variables ± 1. J_{ij}'s and h_i's are programmable parameters.
- in $20\mu sec$, finds the 512-bit/spin configuration (out of 2^{512}) that minimizes the cost function (often producing erroneous results though).
- some of these problems are (classically) computationally-hard, so there's the hope that certain problems won't be as hard for the quantum chip.

The D-Wave Two Chip

- the usefulness of D-Wave Two can be determined by the existence and scope of problems it can solve faster than classical computers.
- finding such problems is the "holy grail" for all those involved in studying the chip. most recent studies (Science ,June 2014) reported no-success.
- what do we mean by "faster"? the hope is that we see difference in scaling behavior, not necessarily in absolute times.

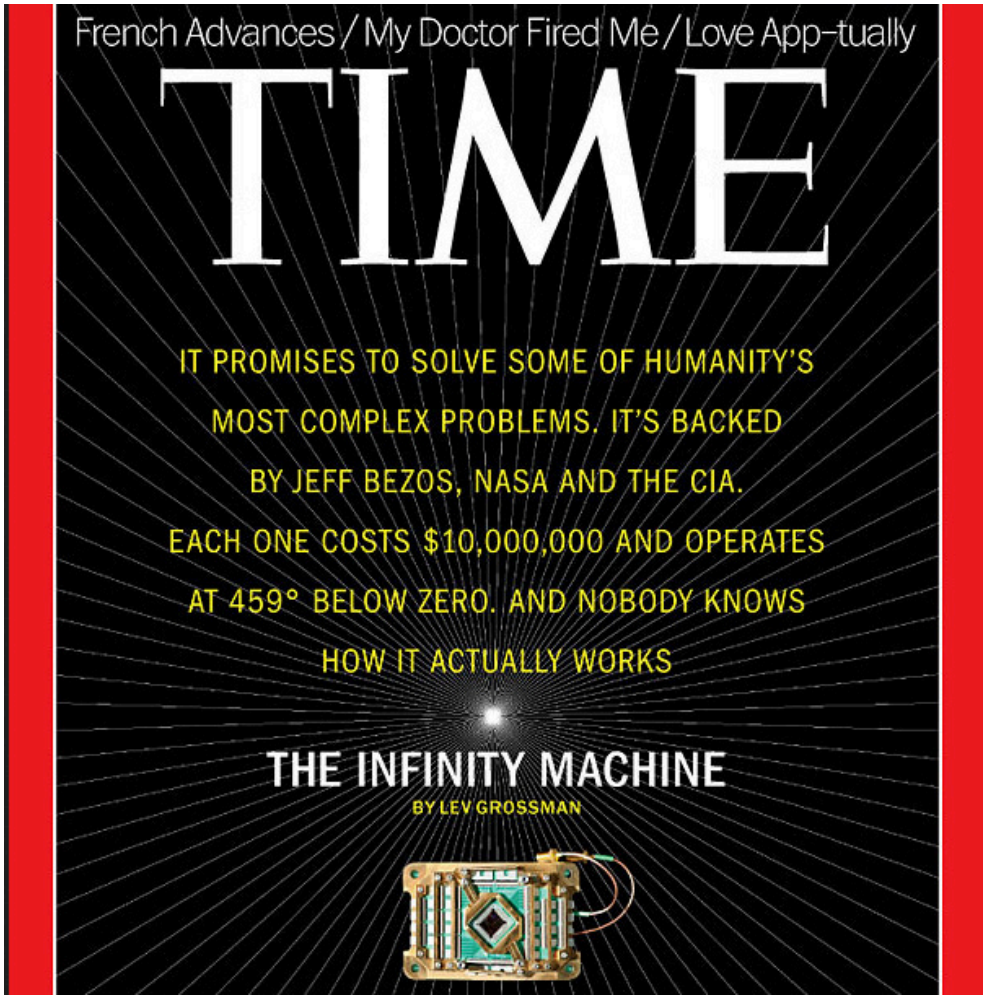


?

=

The D-Wave Two Chip

- lots of interest and attention from the scientific community as well as the general public
- lots of press (NY Times, CNN, Forbes, the economist…)
- a lot of controversy
- a lot still needs to be clarified…



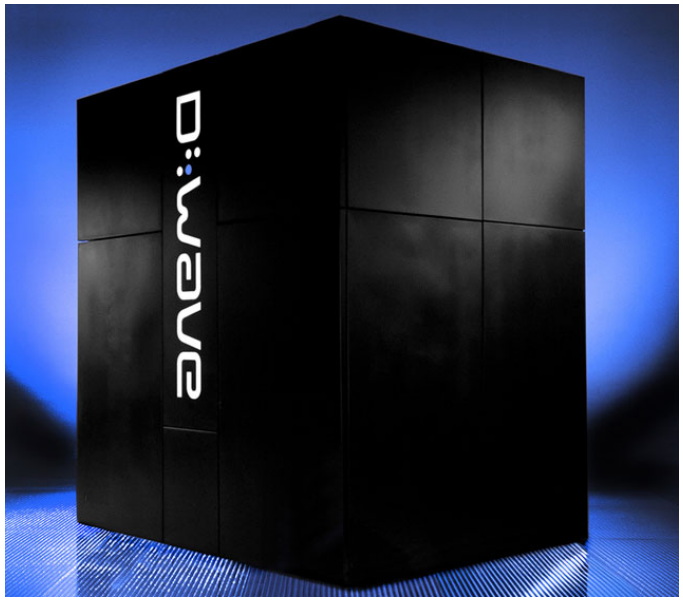
The D-Wave Two Chip

plenty of questions/controversies revolving around the nature of the D-Wave chip

how quantum is it?

how useful is it?

these questions (and more) may be (partially) answered with the aid of super-computing resources



Simulating quantum annealers:

understanding and improving quantum computers

Studying the D-Wave chip

- being able to simulate quantum annealers, specifically the D-Wave chip is an important task for us.

why important?

- the chip itself is only very indirectly accessible.
- understanding the underlying physical model will help us determine how much "quantumness" is in the chip, i.e., is the state of the system entangled or not? can this quantumness be used for anything useful?
- by simulating the machine, we'll be able to propose simple modifications that would enhance certain desired properties.
- this in turn, may lead to observation of "quantum speedup", which would constitute as an important discovery in the field.

Studying the D-Wave chip

- however, simulating quantum annealers is hard!

why hard?

- first, we are not sure about the underlying physical model,
- but even if we did, quantum systems are complicated to simulate!

Studying the D-Wave chip

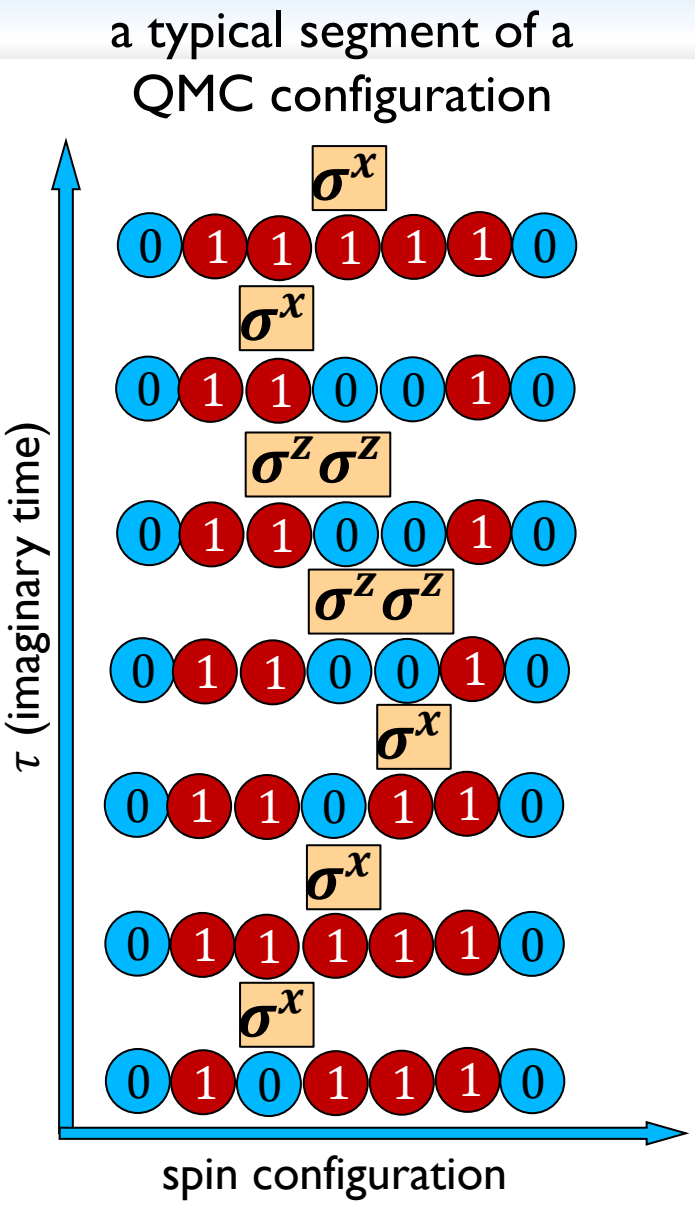
- a quantum bit (or qubit) can be in a superposition of the "0" state and the "1" state: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.
- in principle, the classical description of one qubit requires two infinite-precision complex numbers.
- an *n*-qubit system can be in a superposition of 2^n classical states, e.g., for $n = 3$, $|\psi\rangle = \alpha|000\rangle + \beta|001\rangle + \gamma|010\rangle + \cdots$, i.e., requires 2^n complex numbers.
- this is the "exponential explosion" associated with "fully" simulating quantum systems. each additional bit or spin doubles the resources required for the simulation.
- this is why only very small systems (< 30 quantum bits) can be "fully" simulated.

Studying the D-Wave chip

- on the other hand, it should be clear that such full simulations provide us with much more information than would be accessible in the actual quantum system being simulated.
- the state of the quantum system is not directly accessible.
- with quantum systems, information is obtained only through measurements, each providing 1 bit of data and destroys the state afterwards. this does not happen in classical simulations.
- this is why we have alternatives, e.g., quantum Monte Carlo techniques.

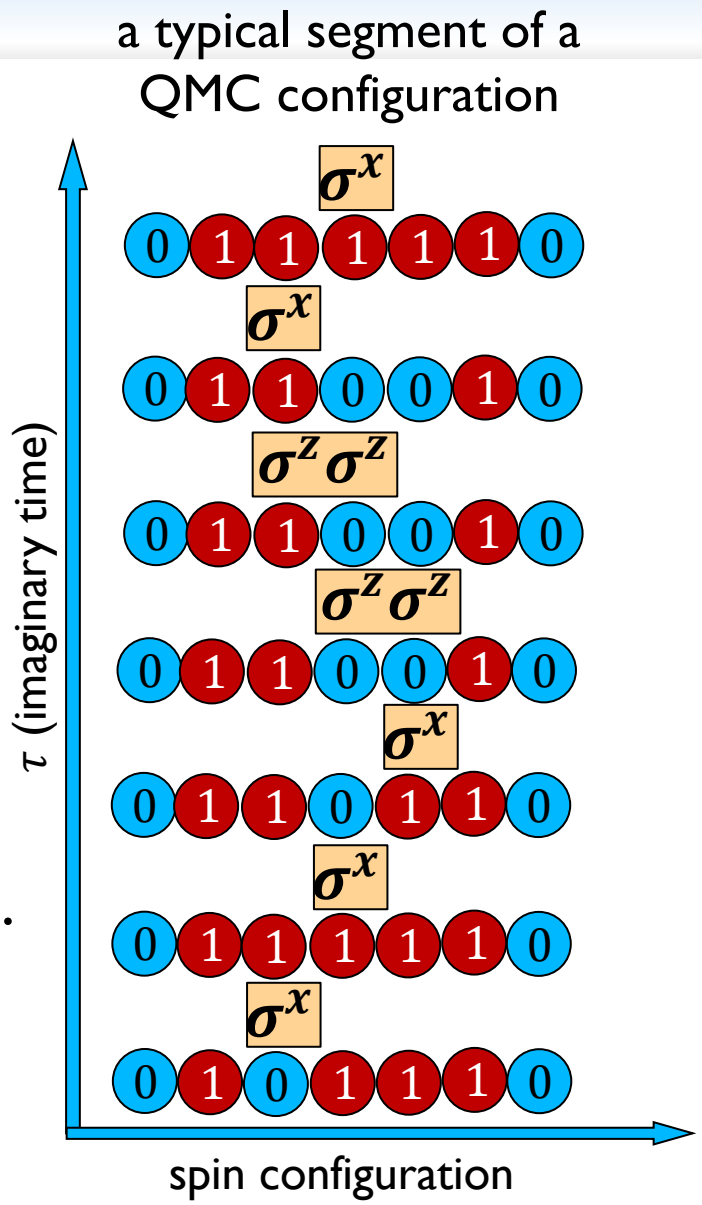
Quantum Monte Carlo

- for large system sizes, we can no longer utilize exact evolution,
- a way to study the various properties of the system is employing quantum Monte Carlo (QMC) techniques, i.e., to simulate a quantum computer.
- with QMC, we (importance-)sample the 2^n states of system. exact-numerical up to statistical errors.
- QMC enables access to the equilibrium properties of the system.
- can only be applied to specific types of quantum systems.



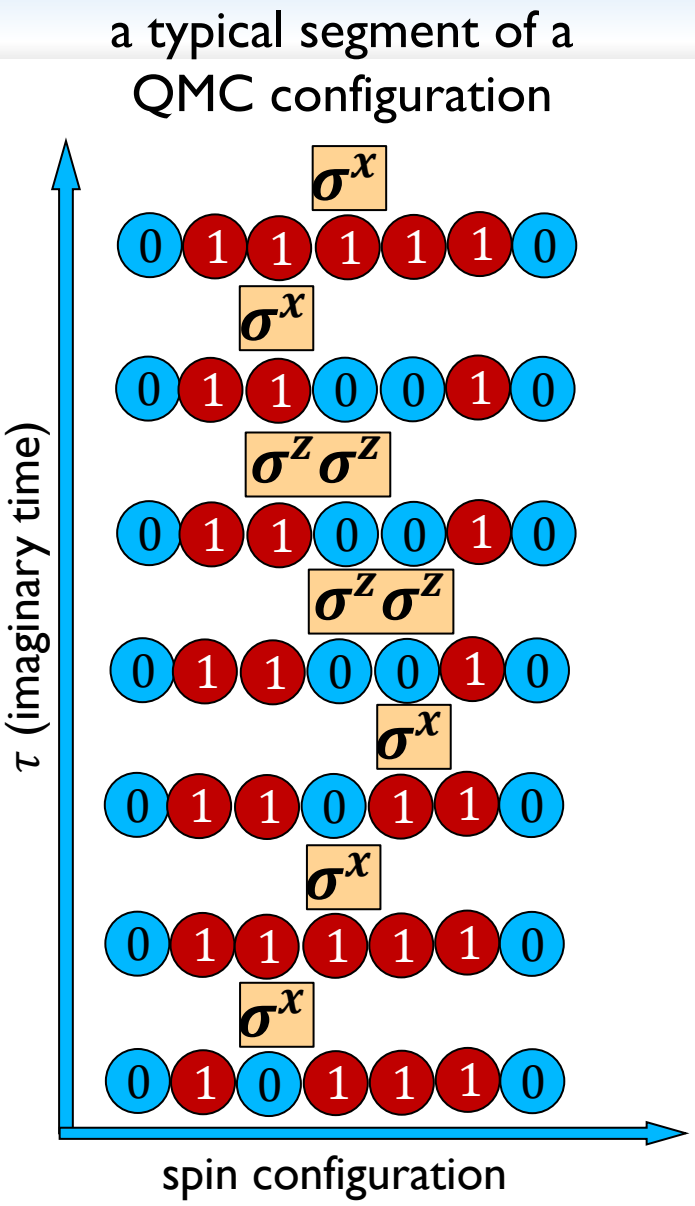
Quantum Monte Carlo

- in QMC, the quantum-computational "explosion" reflects itself in an additional periodic dimension of "imaginary time" $0 \leq \tau < \beta$, where β is the inverse-temperature $(1/T)$.
- simulating low temperatures requires significant computer power (sometimes, goal is zero-temperature).
- memory requirements scale as problem sizes times inverse temperature.



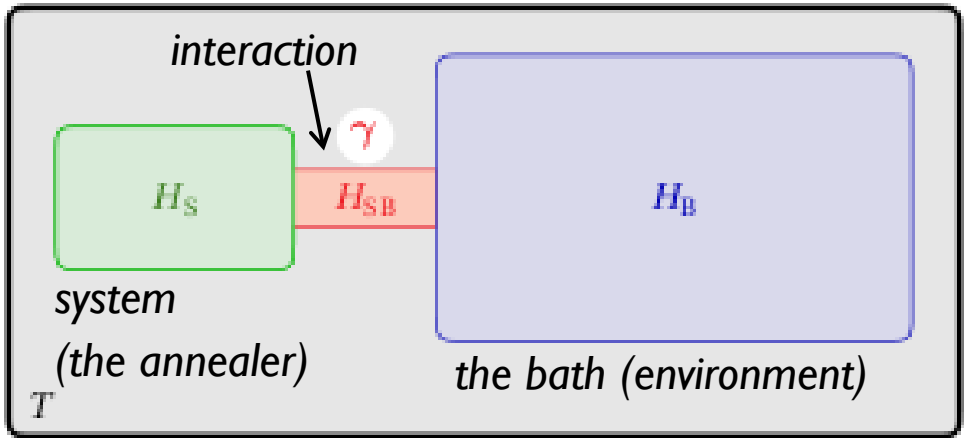
Quantum Monte Carlo

- run times could grow indefinitely if the simulated problems are hard.
- happens when the system undergoes a first order phase transition in which the equilibrium distributions above and below some critical temperature are discontinuous (a "jump").
- luckily, because we are dealing with sampling and statistics, QMC codes are trivially parallelizable.
- the D-Wave annealer can be simulated with QMC.



Interaction with environment

- things get even more complicated, when interaction of the system with the environment (a thermal bath) is taken into account.
- this is crucial for a reliable description of the underlying physics.
- since the environment is infinitely large, approximation methods must be applied (Lindbladian dynamics).
- even then, the task is very challenging. usually involves exact diagonalization of large matrices.
- requires lots of computational resources but gain could be huge.
- understanding the inner machinery may lead to better designs of the chip.
- may lead to observation of quantum speedup.



Benchmarking D-Wave:

determining the usefulness of quantum annealers

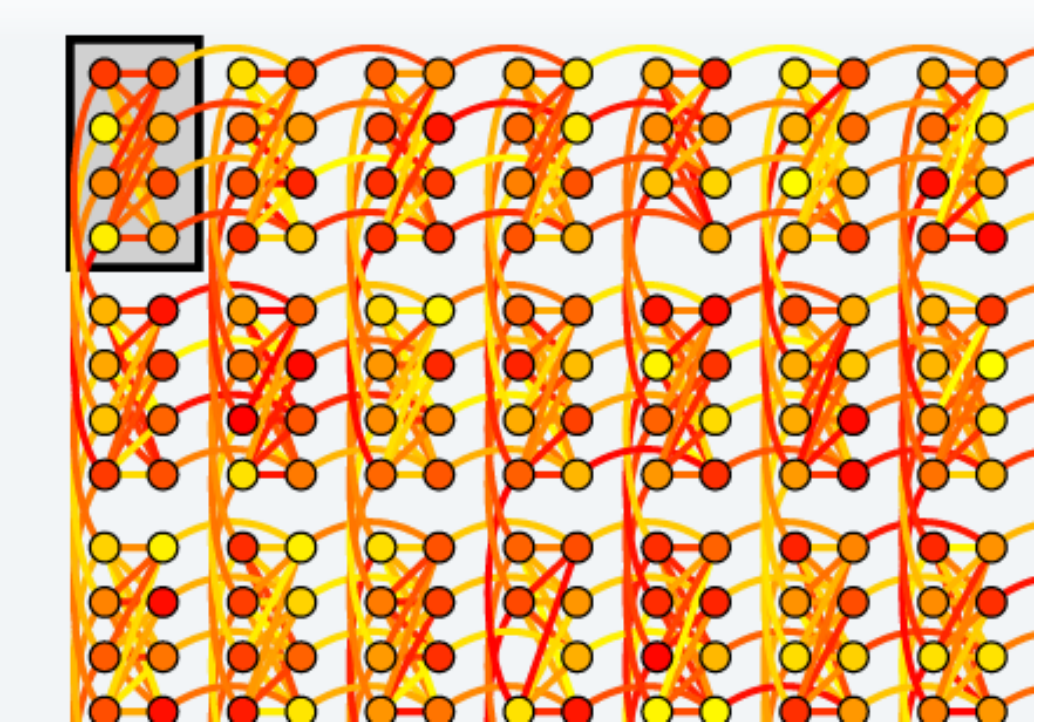
Benchmarking D-Wave

is our quantum annealer useful for anything?

- even if we manage to show that the chip is quantum.
- the question of whether or not this "quantumness" may be used for anything useful, i.e., for solving some problems faster, remains.
- right now, it is almost impractical to test this on real-life problems, because of how small the current chip is.
- next best thing is generating artificial problems, and testing the performance of the D-Wave chip on these against classical state-of-the art optimization algorithms running on cutting-edge hardware (CPUs, GPUs).

Generating hard problems

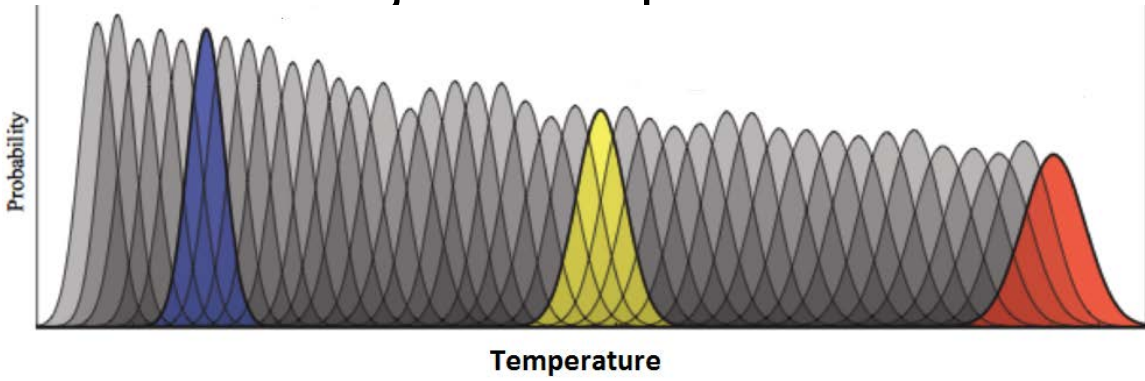
- we can study "artificial" problems by generating random problems.
- some of those are also known to be computationally hard.
- while these will not be directly "useful" for real-life applications, they may help us to find out if there are problems that D-Wave Two can solve faster than traditional resources.
- we can simulate a "modified" chip to look for evidence for speedup.



a randomly-generated problem on D-Wave Two's Chimera architecture

Benchmarking D-Wave

- we must compare the performance of the quantum chip against state-of-the-art algorithms. among the algorithms D-Wave is compared against are:
  - Selby's algorithm: specifically tailored to solve problems defined on D-Wave's Chimera architecture, uses the tree-like features of the graph.
  - simulated (thermal) annealing: may be viewed as a classical version of quantum annealing. it uses gradually decreasing temperature / thermal fluctuations to escape local minima.
  - parallel tempering : running many replicas of the system with various temperatures, allowing simulations of adjacent temperatures to switch configurations.
  - other algorithms…



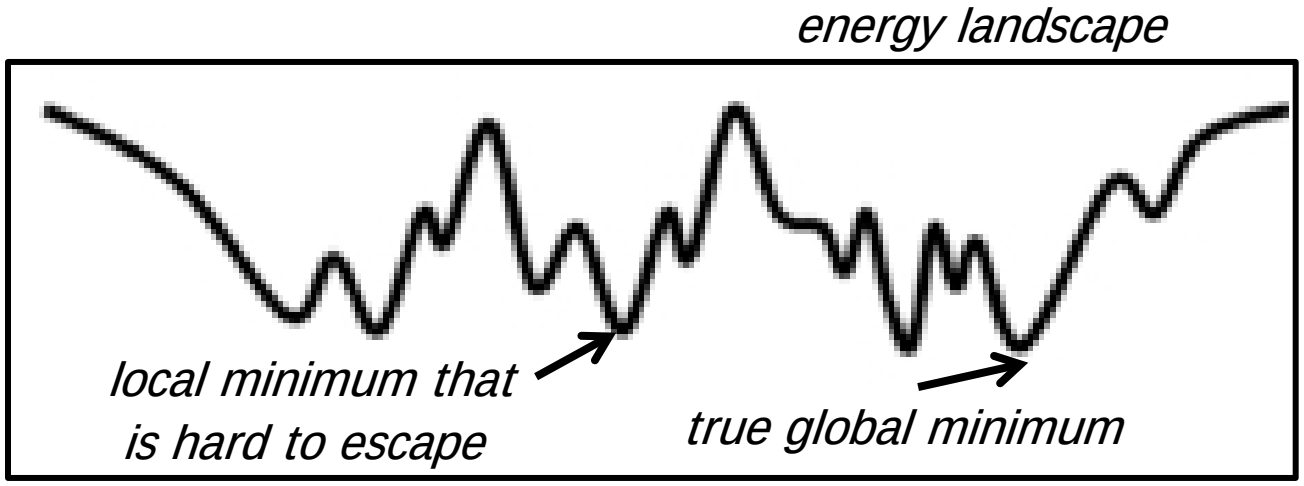
Generating hard problems

- so far, no luck. no quantum speedup has been observed.
- but some (partial) answers are starting to emerge
  - recent results point to the fact that the machine is indeed quantum (Lanting et al., PRX, May 2014).
  - benchmarking tests are so far inconclusive (Rønnow et al., Science, June 2014).
  - theoretical work points to new directions where we can look for more appropriate problems (Katzgraber et al., PRX, April 2014).

better benchmarks are needed, finite-temperature spin-glasses

Generating hard problems

- for some problems, the so-called spin glasses, the energy landscape is so jagged, that classical heuristic solvers tend to get stuck in local minima, failing to solve the optimization problem.
- in the presence of first-order phase transitions, even clever algorithms such as parallel tempering will not do.
- hope is that quantum annealers could use their ability to tunnel through high energy barriers to get to the true global solutions.
- we don't know if such problems that are classically-hard could also be quantum-easy.

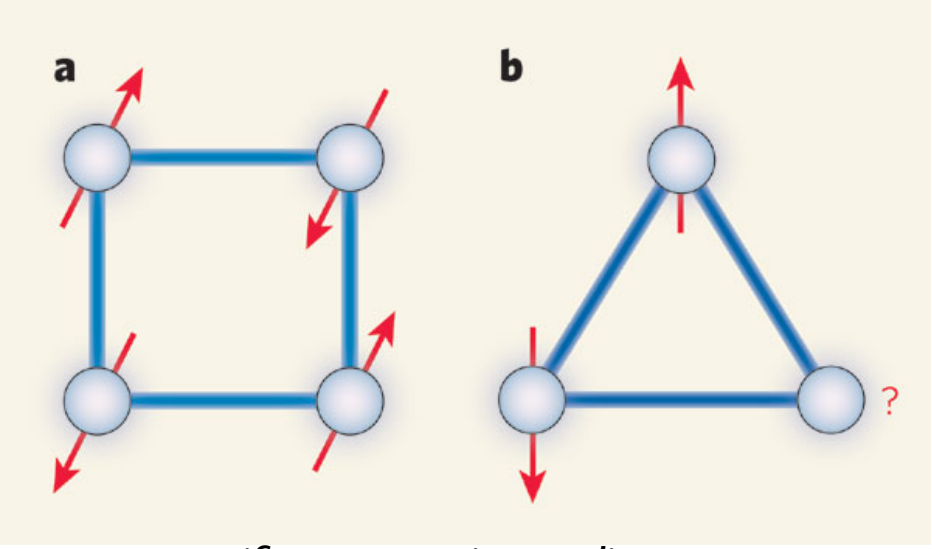


Computational requirements

- benchmarking requires lots of computation resources.
- traditional CPU's as well as GPU's.
- if problems are really hard, even 512-bit problems may run for hours or even days.
- analyzing the performance of classical algorithms on instances of such spin-glasses is extremely resource-demanding.
- hundreds of thousands of instances must be tested to establish the typical behavior of an algorithm on a random problem.
- also, scaling with problem size needs to be tested.
- observation of a different scaling behavior for quantum and for classical solvers would be an important discovery.

Generating hard problems

- these days, we are looking into various types of problems that may potentially reveal quantum speedups.
- we are trying to engineer problems that are expected to be hard.
- specifically, problems with a lot of frustration.
- these are known to be hard for classical algorithms.
- we don't know how D-Wave will do.
- so far, preliminary results are somewhat promising.



antiferromagnetic couplings

Conclusions and outlook

Conclusions

- heavy "classical" computing is extremely important for the simulation and benchmarking of quantum computers (annealers).
- while computationally demanding, computation is expected to reveal many sought-after so-far-unknown properties of the quantum system as well as ideas for improving the hardware, such as different architectures, effects of reduction of noise and temperature, etc.
- applications of state-of-the-art classical optimizers will reveal "intrinsically-hard" optimization problems, whose solutions may be reached faster by quantum computers.
- will help us understand the difference between quantum-hard and classical-hard.

Outlook

- not sure what the quantum Moore's law is going to be.
- so far, quantum annealer sizes double each year.
- for the corresponding classical simulations, each added qubit *doubles* the resource requirements.
- a x10 increase in supercomputing resources a few years from now, may certainly help us "go up in size" as far as simulations go.
- super-optimistic view: judging by annealer sizes, this would be around the time where we hope quantum computers would leave super-computers behind (has not happened so far!).
- therefore, next few years should be the most interesting ones!

Acknowledgements

- we thank OLCF/ORNL for granting us this award.
- we thank Jack Wells and the organizers of this meeting for the invitation to speak here.

Thank you!

Quantum Computational Science

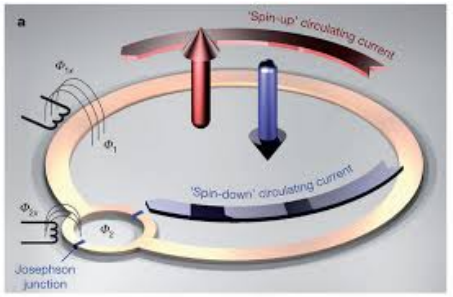
Itay Hen

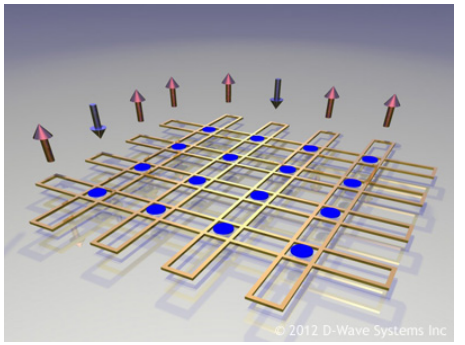
2014 OLCF Users Meeting
Oak Ridge, TN
July 24, 2014

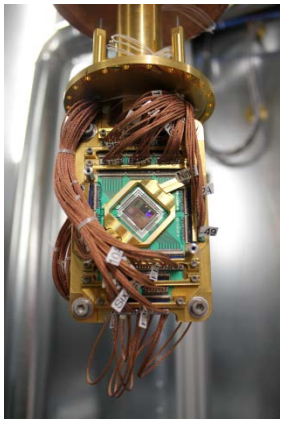
Information Sciences Institute
USC School of Engineering

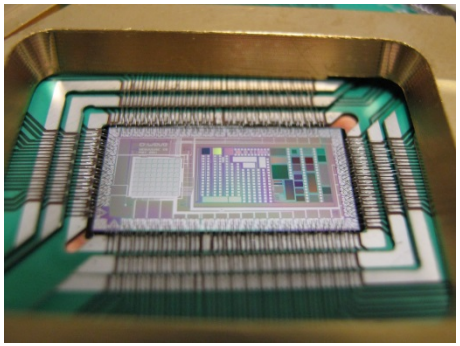
The D-Wave Two Chip

- in the D-Wave Two chip, qubits are superconducting flux circuits.
- these are arranged in arrays of intersecting loops to form the "Chimera" architecture.
- the circuit is magnetically shielded and cooled down to 17mK.



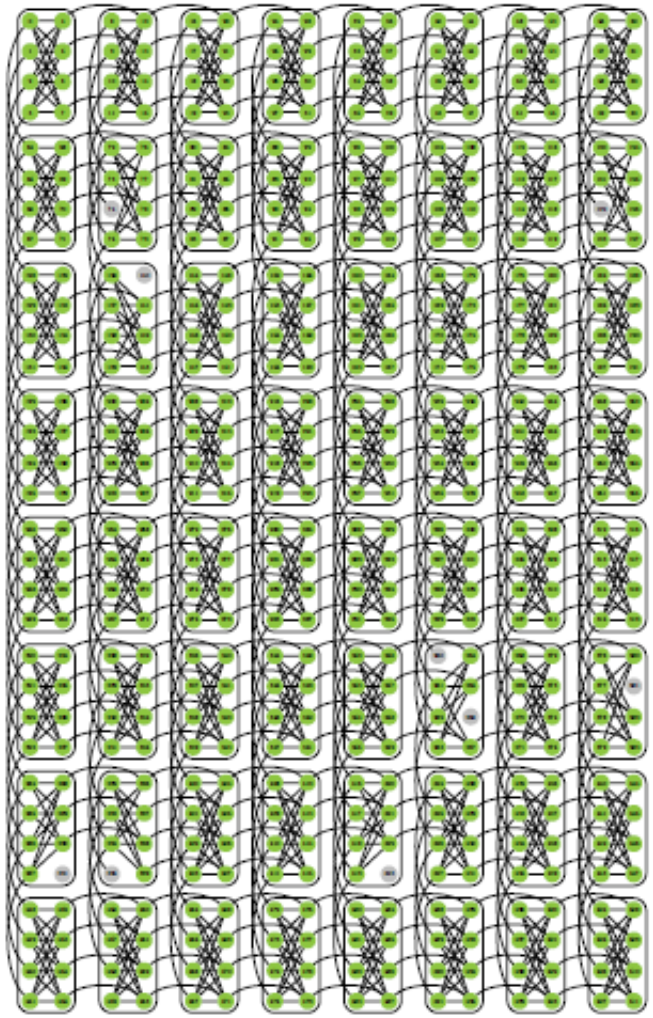






Chip topology: *Chimera* graph

- Motivation for the topology design?
  - Individual addressing of qubits.
  - Scalability & large automorphism.
  - Low decoherence.
- Problem size?
  - $N = 512$ Boolean variables.
- Possible cost functions (Hamiltonians):
  - Any embeddable Hamiltonian $\mathcal{H}(\{S\})$.
  - Potential massive overhead…



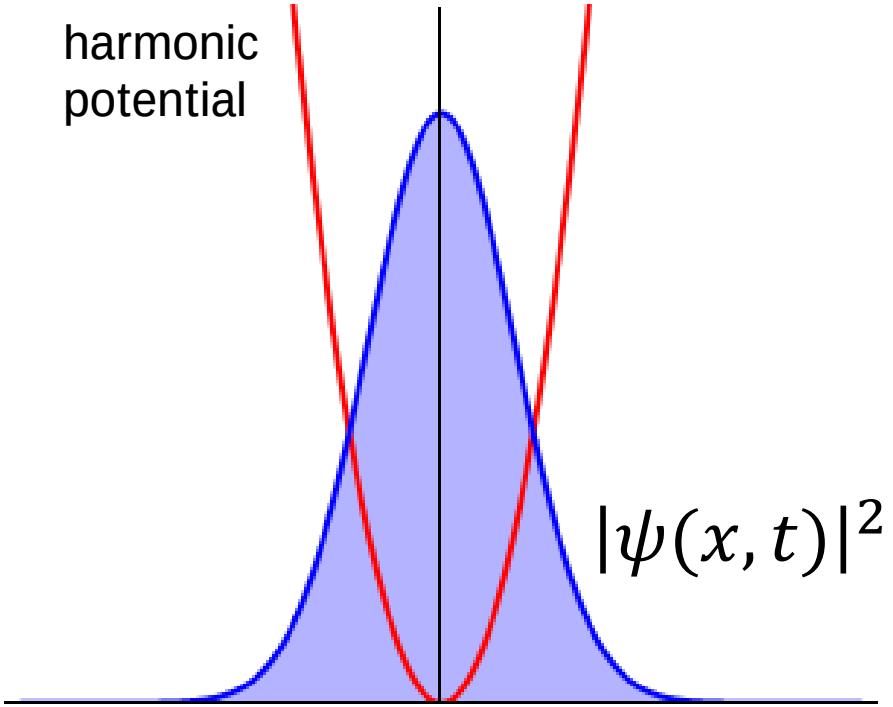
Adiabatic Quantum Computing

The adiabatic theorem of QM

- the adiabatic theorem of QM tells us that a physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.

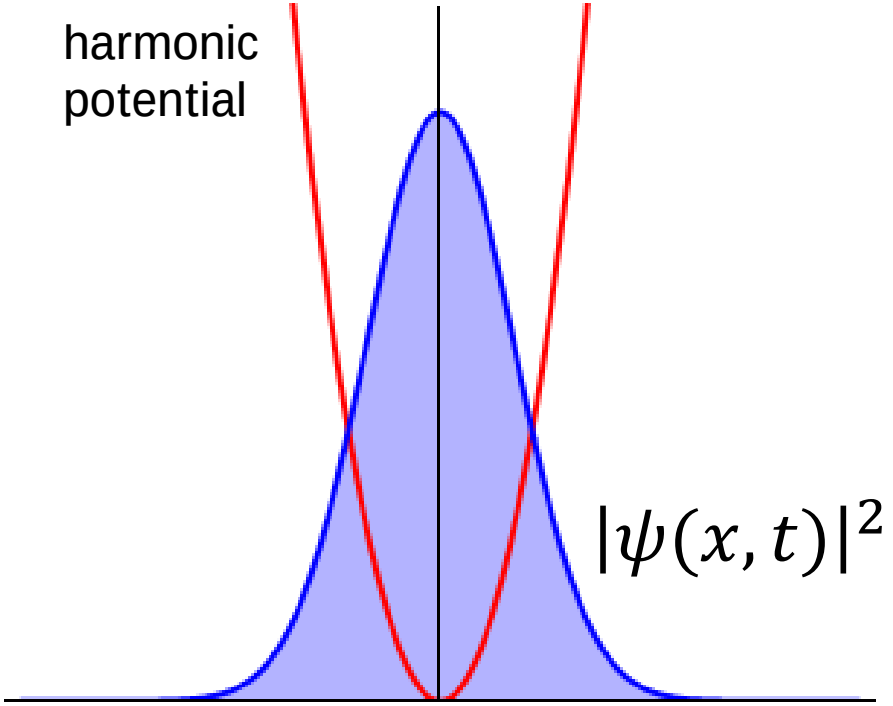
The adiabatic theorem of QM

- the adiabatic theorem of QM tells us that a physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.
- example: change the strength of a harmonic potential of a system in the ground state:
- an abrupt change (a *diabatic* process):



The adiabatic theorem of QM

- the adiabatic theorem of QM tells us that a physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.
- example: change the strength of a harmonic potential of a system in the ground state:
- a gradual slow change (an *adiabatic* process): wave function can “keep up” with the change.



The quantum adiabatic algorithm (QAA)

- the mechanism proposed by Farhi et al., the QAA:

1. take a difficult (classical) optimization problem, generically: find minimizing configuration of the cost function $E = f(x_1, x_2, \ldots, x_n)$ where x_i are binary variables $(0,1)$ and $f(\cdots)$ is given.

2. encode the problem in a "problem" Hamiltonian, $\widehat{H}^{(p)}$, such that its ground state encodes the solution.

3. prepare the system in the ground state of another, easily solvable, "beginning (driver) Hamiltonian" $\widehat{H}^{(b)}$.

4. vary the Hamiltonian slowly and smoothly from $\widehat{H}^{(b)}$ to $\widehat{H}^{(p)}$ until ground state of $\widehat{H}^{(p)}$ is reached.

The quantum adiabatic algorithm (QAA)

- the interpolating Hamiltonian is this:

$$\hat{H}(t) = [1 - s(t)]\hat{H}^{(b)} + s(t)\hat{H}^{(p)}$$

$\hat{H}^{(b)}$ is an easily solvable beginning Hamiltonian, which does not commute with $\hat{H}^{(p)}$

$\hat{H}^{(p)}$ is the problem Hamiltonian whose ground state encodes the solution of the optimization problem

- the parameter s obeys $0 \leq s(t) \leq 1$, with $s(0) = 0$ and $s(\mathcal{T}) = 1$. also: $\hat{H}(0) = \hat{H}^{(b)}$ and $\hat{H}(\mathcal{T}) = \hat{H}^{(p)}$.
- here, t stands for time and $\mathcal{T}$ is the running time, or complexity, of the algorithm.

The quantum adiabatic algorithm (QAA)

- the interpolating Hamiltonian is this:

$$\hat{H}(t) = [1 - s(t)]\hat{H}^{(b)} + s(t)\hat{H}^{(p)}$$

- the adiabatic theorem ensures that if the change in $s(t)$ is slow enough, the system will stay close to the ground state of the instantaneous Hamiltonian throughout the evolution.

- a measurement at the end of the evolution, will give the solution of the original problem with high probability.

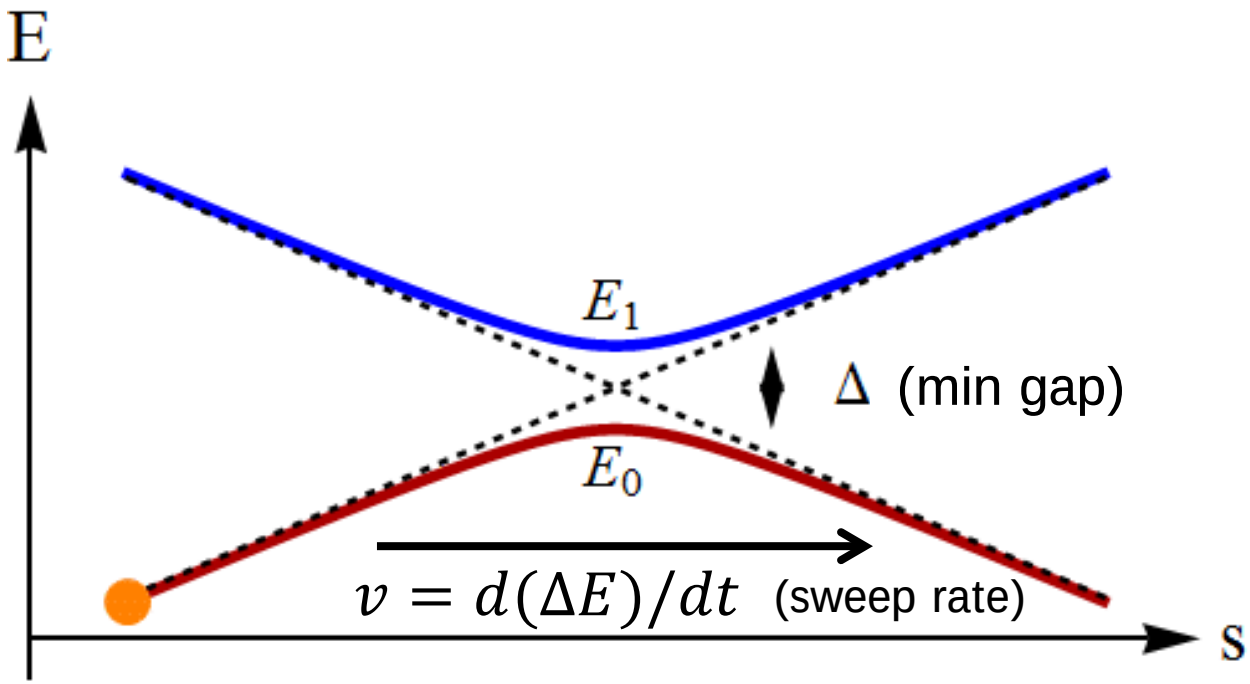
process should be slow,
but how fast can it still be?

Landau-Zener transition

- bottleneck is likely to be where the gap to the first excited state is smallest. there, the probability to "get off track" is maximal.

- generic case: Landau-Zener theory gives the prob. to stay in ground state as:

$$P \approx 1 - e^{-\frac{\pi\Delta^2}{2\hbar v}}$$



- runtime should obey:

$$\mathcal{T} \gg \frac{max_s\left|\langle 0|d\hat{H}/ds|1\rangle\right|}{min_s[gap^2]}$$

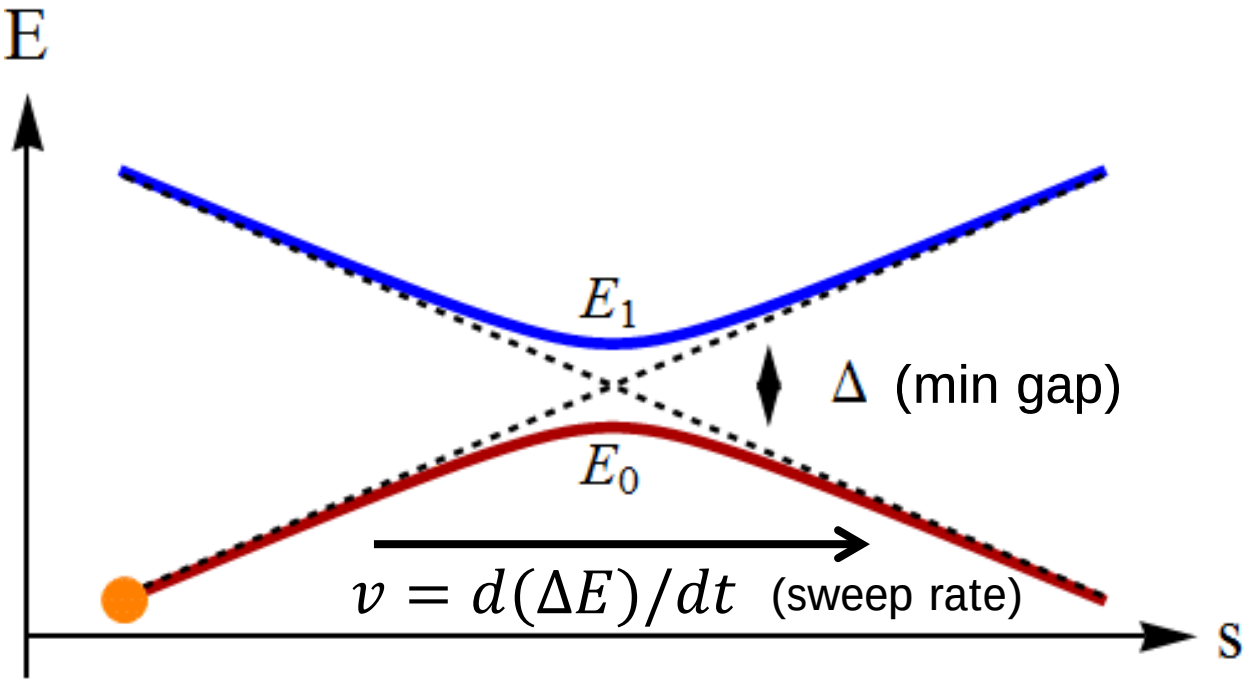
Landau-Zener transition

- ly to be where the gap to the first excited
  here, the probability to "get off track" is

the larger the minimum gap Δ is, and the slower the process is, the better our chances are to stay in the ground state.

ground st s:

$$P \approx 1 - e^{-\frac{\pi\Delta^2}{2\hbar v}}$$

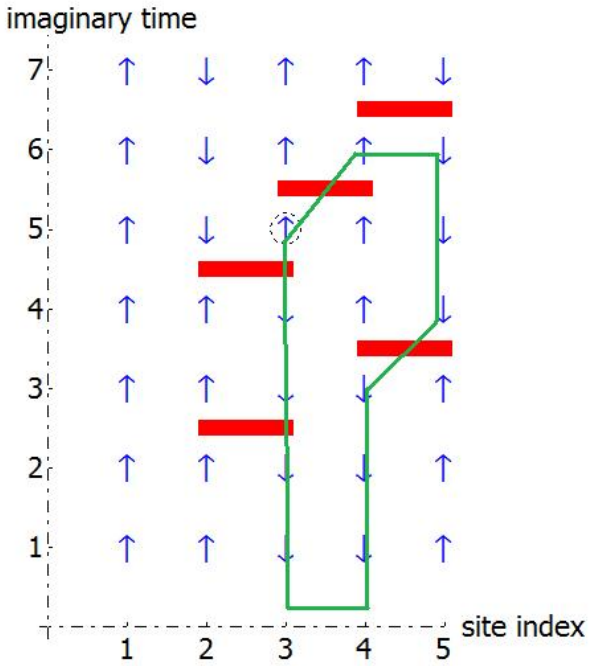


- runtime should obey:

$$\mathcal{T} \gg \frac{max_s\left|\langle 0|d\hat{H}/ds|1\rangle\right|}{min_s[gap^2]}$$

Stochastic series expansion algorithm

- [Sandvik 1994]
- uses both local and global updates
-
-



Stochastic series expansion algorithm

- partition function

$$Z = \mathrm{Tr}\left[e^{-\beta \hat{H}}\right]$$

-

$$\mathrm{Tr}[\cdot] = \sum_{\alpha} \langle \alpha | \cdot | \alpha \rangle$$

$\{|\alpha\rangle\}$

-

$$e^{-\beta \hat{H}} = \sum_{n=0}^{\infty} \frac{\beta^n}{n!} (-\hat{H})^n$$

Stochastic series expansion algorithm

- the partition function then becomes:

$$Z = \sum_{\alpha} \sum_{n=0}^{L} \frac{\beta^n}{n!} \langle \alpha | (-\hat{H})^n | \alpha \rangle$$

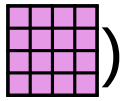
where the Taylor expansion is cut off after *L* terms ($L \gg \langle n \rangle$).

Stochastic series expansion algorithm

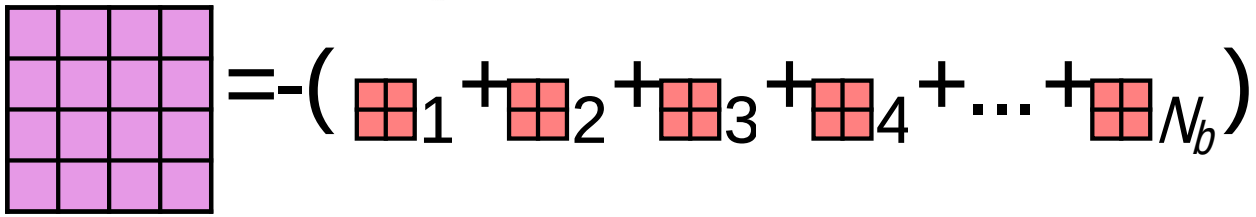
- the partition function then becomes:

$$Z = \sum_{\alpha} \sum_{n=0}^{L} \frac{\beta^n}{n!} \langle \alpha | (-\hat{H})^n | \alpha \rangle$$

where the Taylor expansion is cut off after *L* terms ($L \gg \langle n \rangle$).

- as a next step, we rewrite the Hamiltonian (denoted ) as:

$$\hat{H} = -\sum_b \hat{H}_b$$

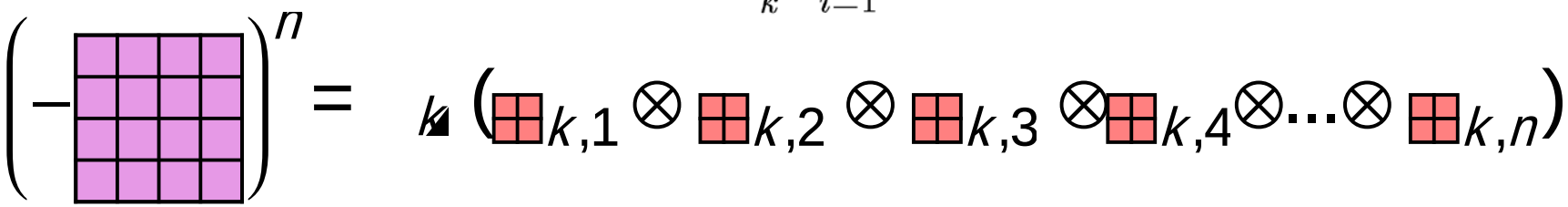


  - where *b* is a “bond” index, which is a pair of two interacting sites.
  - and $\hat{H}_b$ ) is then a “local” two-site operator (a 4x4 matrix).

Stochastic series expansion algorithm

- the terms $(-\hat{H})^n$ inside the partition function are then expanded as:

$$(-\hat{H})^n = \sum_k \prod_{i=1}^{n} \hat{H}_{b_i}^k$$



Stochastic series expansion algorithm

- the terms $(-\hat{H})^n$ inside the partition function are then expanded as:

$$(-\hat{H})^n = \sum_k \prod_{i=1}^{n} \hat{H}_{b_i}^k$$

$$\left(-\right)^n = \sum_k \left(\otimes \otimes \otimes \otimes \cdots \otimes \right)$$

- and the resulting operator products (operator sequences) are “padded” by unit operators in order to fix the length of all sequences at *L*.

$$(-\hat{H})^n = \sum_{k \in S_L} \frac{(L-n)!n!}{L!} \prod_{i=1}^{L} \hat{H}_{b_i}^k$$

$$\left(-\right)^n = \sum_k \frac{(L-n)!n!}{L!} \left(\otimes \otimes \otimes \otimes \otimes \otimes \cdots \otimes \right)$$

Stochastic series expansion algorithm

- the partition function then takes its final form:

$$Z = \sum_{\alpha} \sum_{k \in S_L} \beta^n \frac{(L-n)!}{L!} \langle \alpha | \prod_{i=1}^{L} \hat{H}_{b_i}^{k} | \alpha \rangle$$

all possible operator sequences of length L | weight | configuration

- each configuration in this double sum is a pair (Fock state, operator sequence):

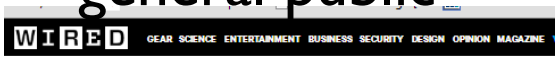
$$\left(|\alpha\rangle, \{\hat{H}_{b_i}\} \right)$$

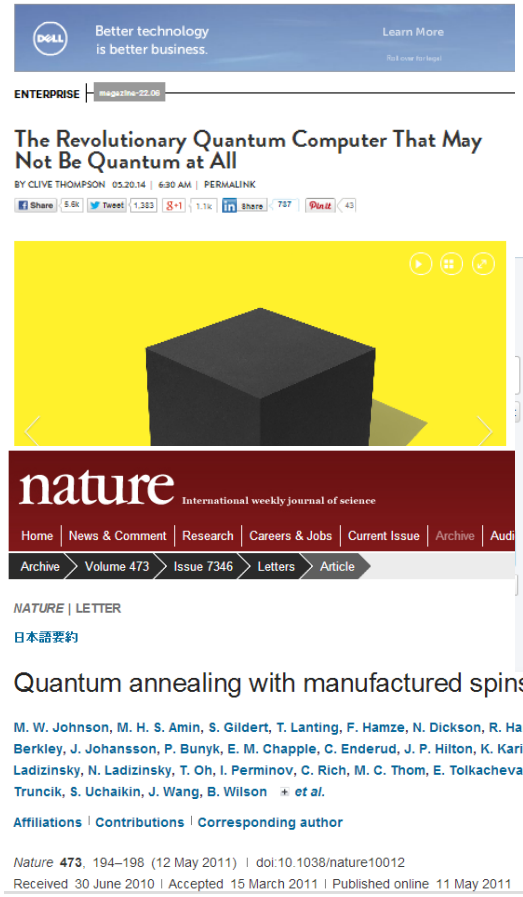
- core of the SSE algorithm is to sample these configurations according to their weights

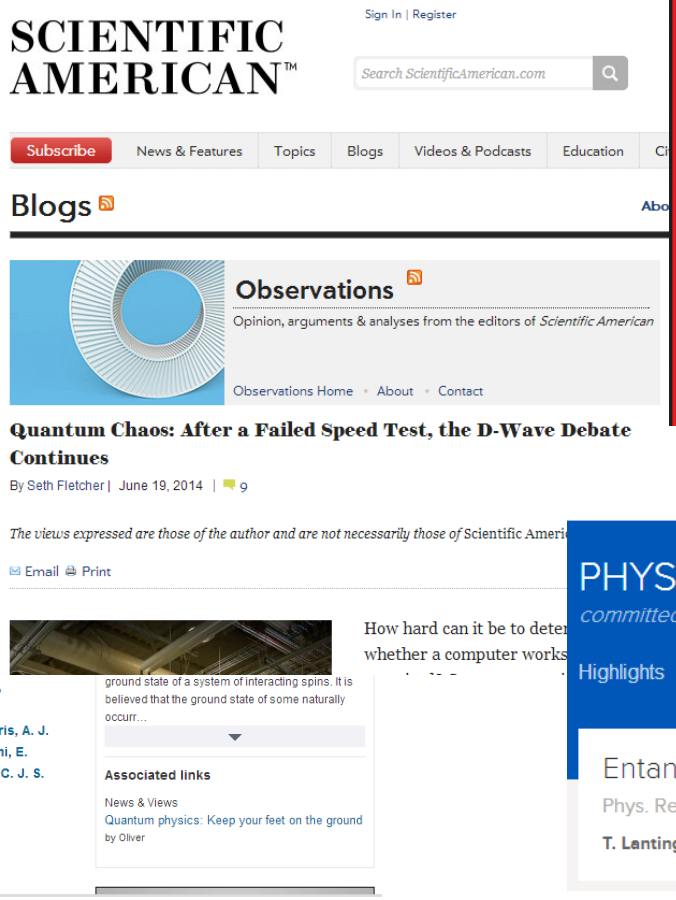
-

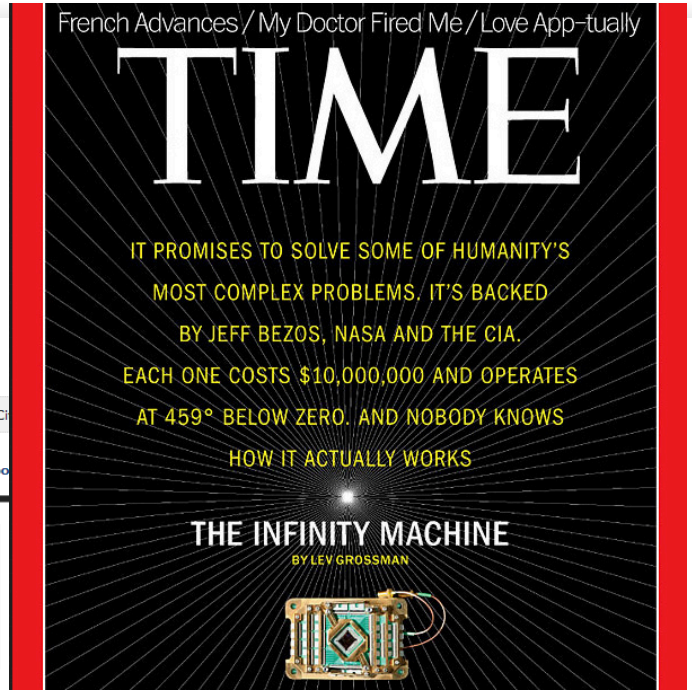
The D-Wave Two Chip

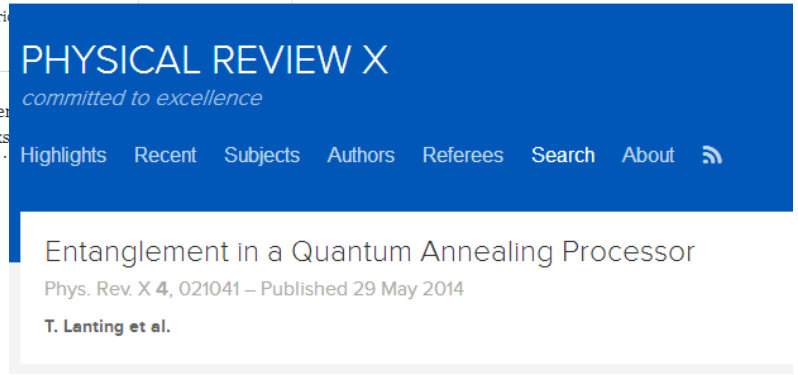
- Lots of interest and attention from scientific community as well as the general public





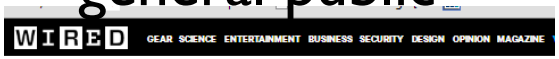


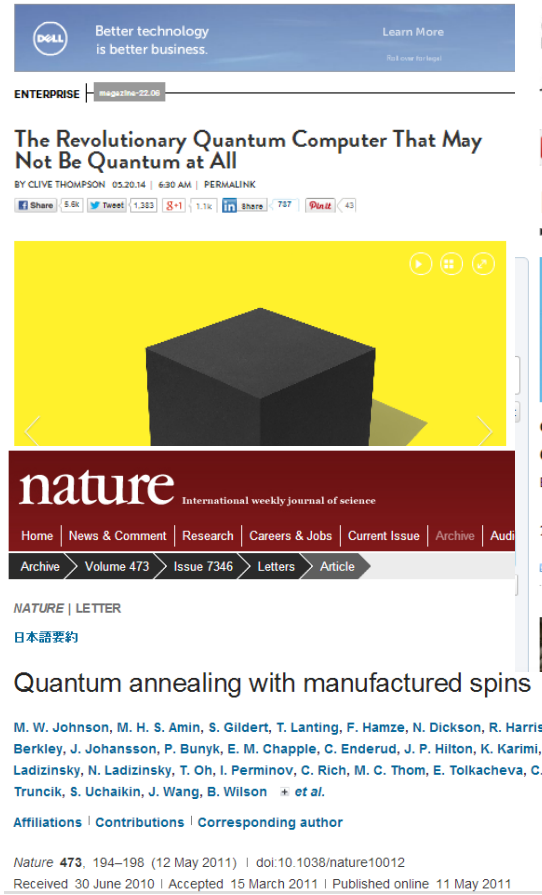


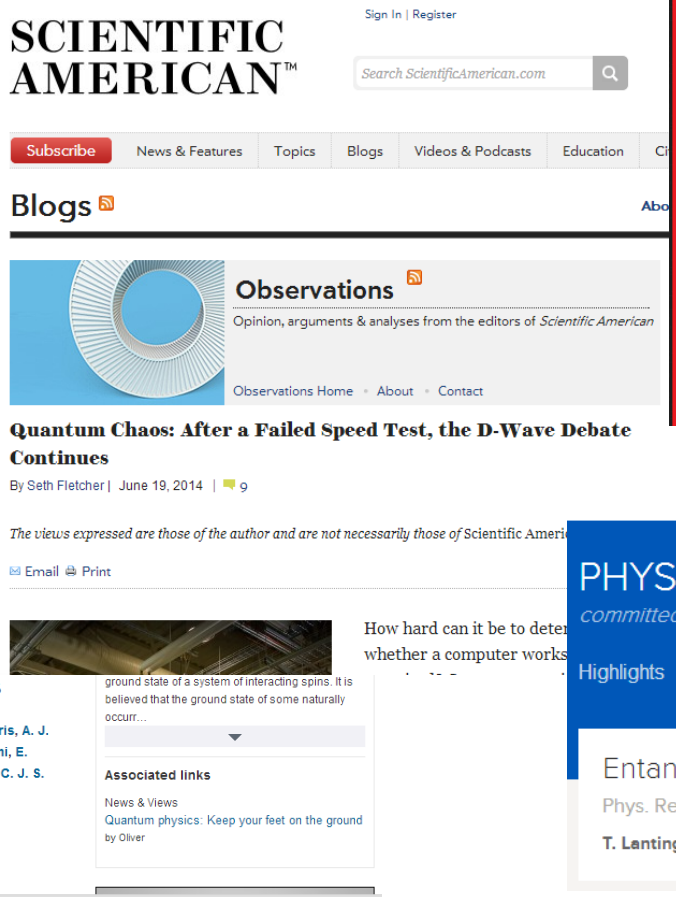


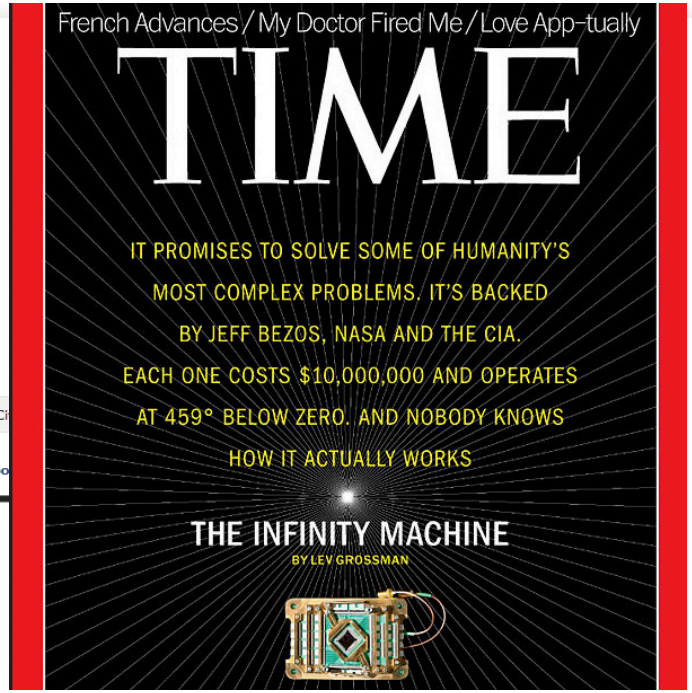
The D-Wave Two Chip

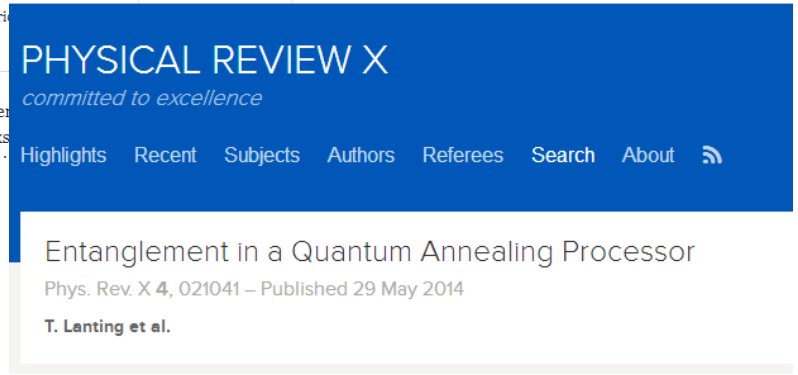
- Lots of interest and attention from scientific community as well as the general public





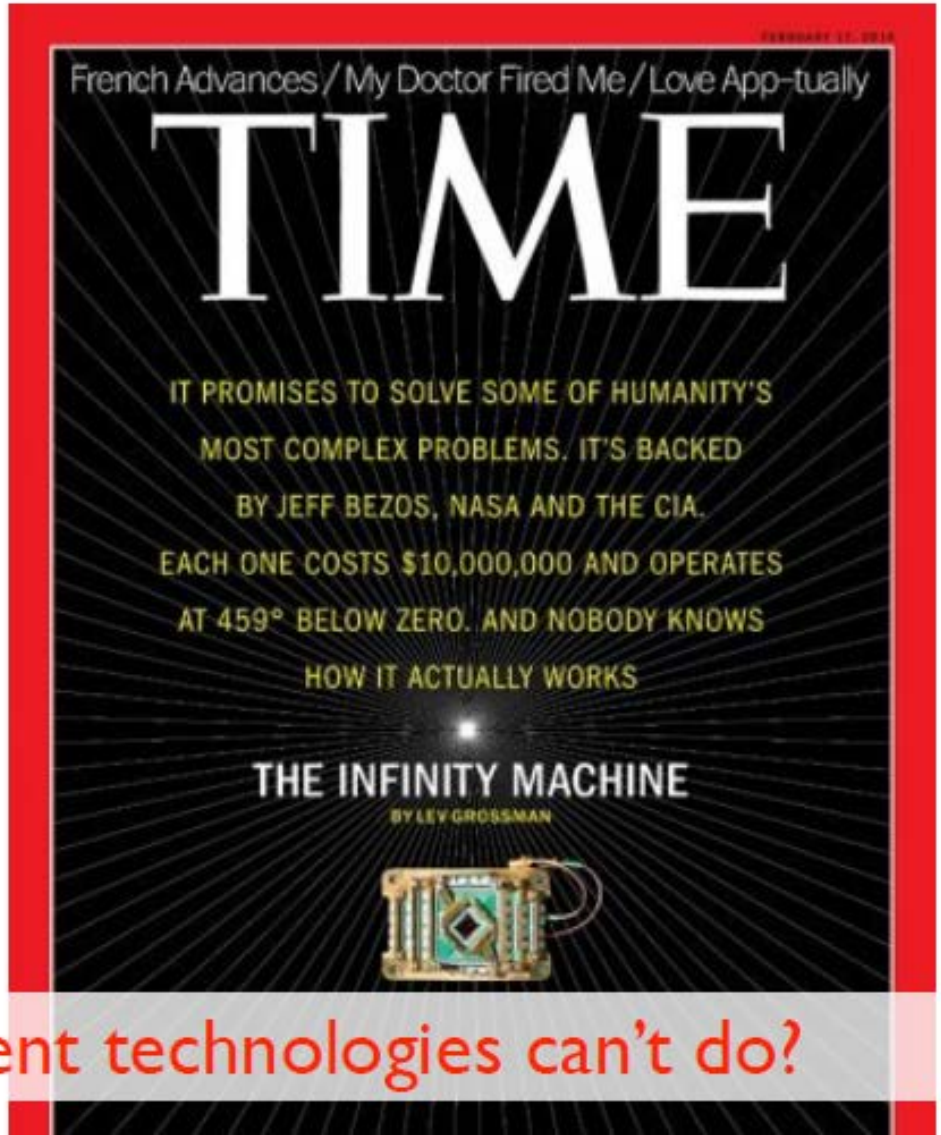






Huge technological feat… Why the fuss?

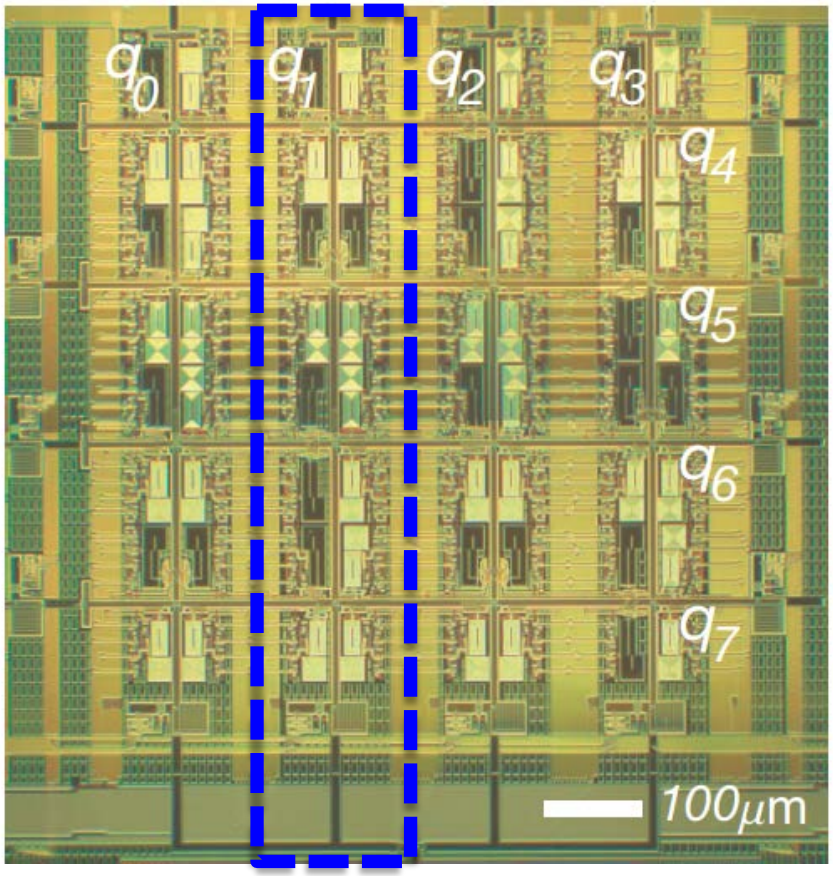
- Wide media coverage:
  - CNN & Bloomberg
  - New York Times
  - Economist
  - Forbes
  - …
- Controversy:
  - Claims of speeds higher than commercial codes.
  - Scientists sceptic …

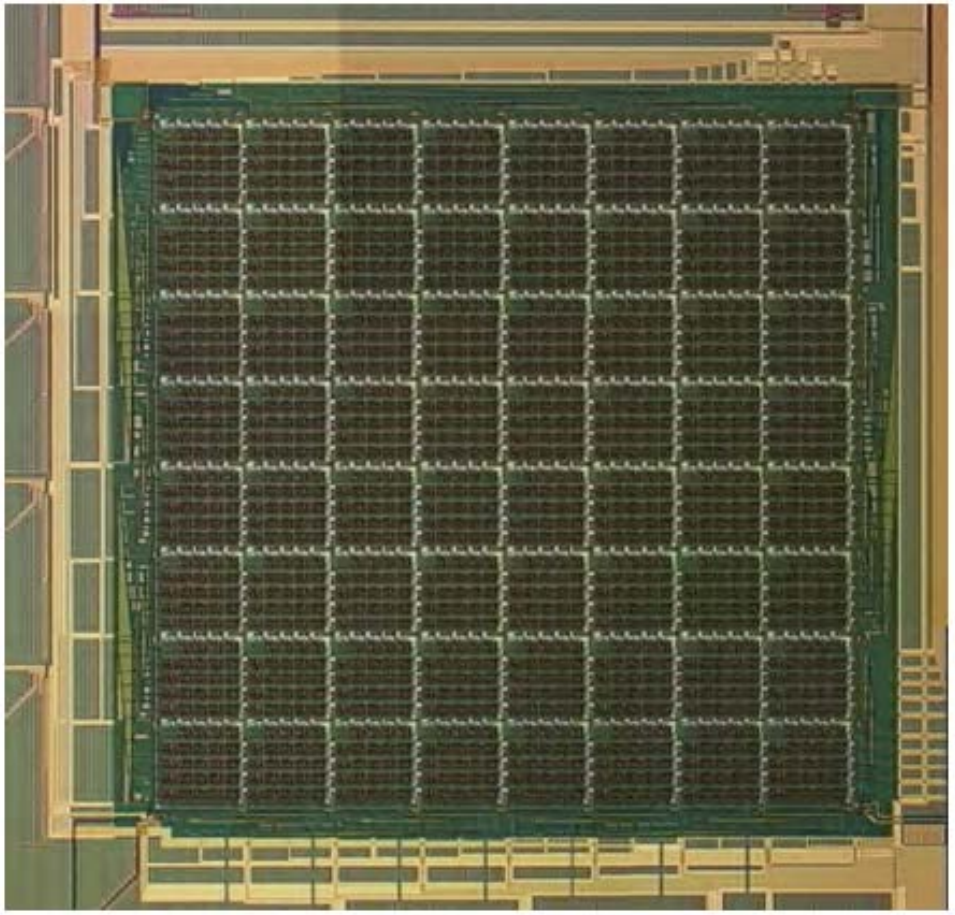


Can it do things current technologies can't do?

D-Wave Two Processor Graph

- 503/512 functional qubits with "Chimera graph" couplings

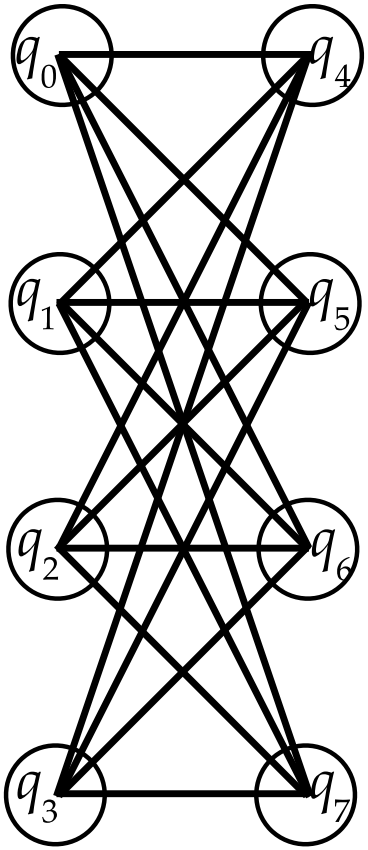




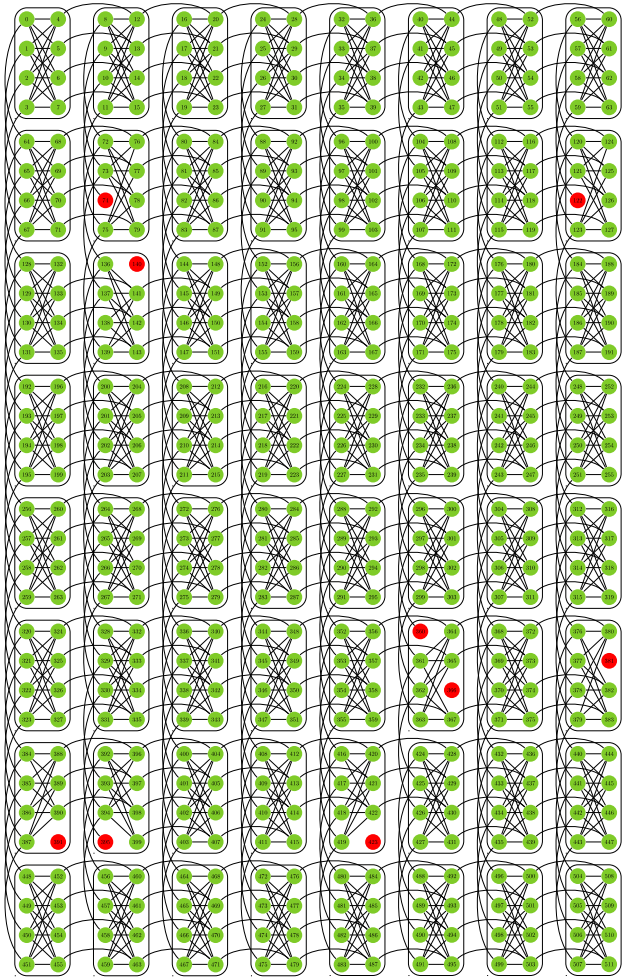
D-Wave Two Processor Graph

- Eight-Qubit Unit Cells and Tiling into 4 x 4 array

unit cell coupling graph



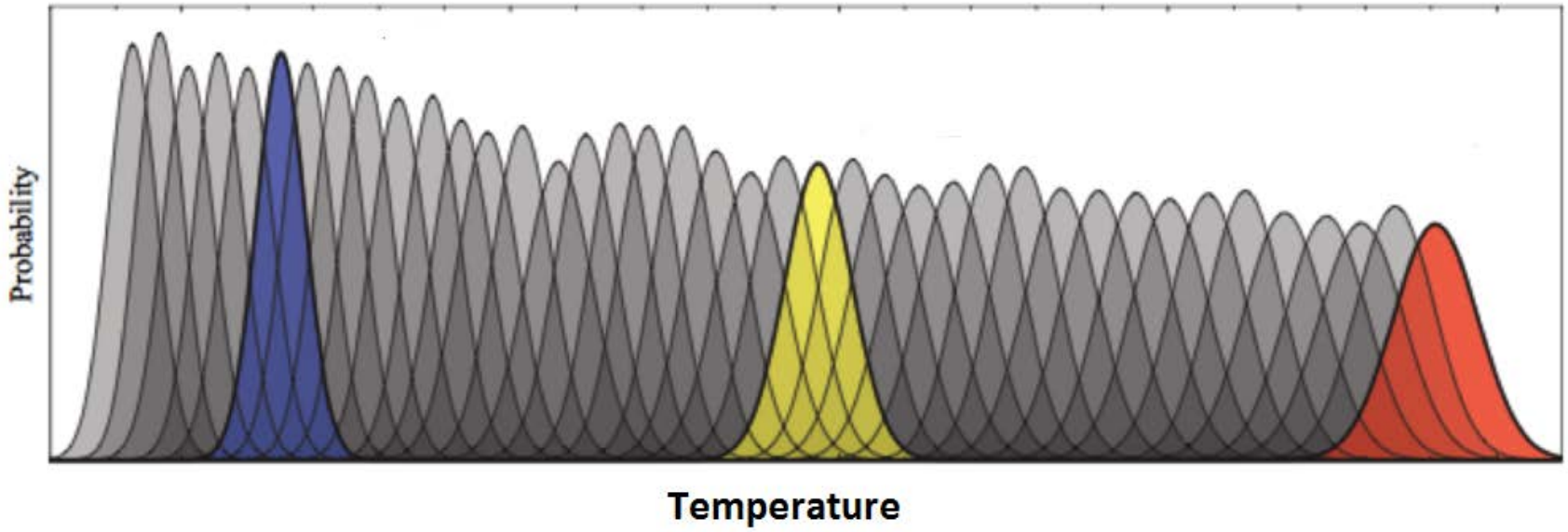
"Chimera" coupling graph of entire chip



Benchmarking D-Wave: Generating hard problems

Parallel Tempering

- for problem for classical solvers
- State of the art optimizers are Parallel Tempering
- in spin glasses it is not useful to work with one replica.
- we need many replicas in parallel.
- different temperatures



Motivation

in what ways quantum computers are more efficient than classical computers?

what problems could be solved more efficiently on a quantum computer?

what is a quantum computer?

Motivation

in what ways quantum computers are more efficient than classical computers?

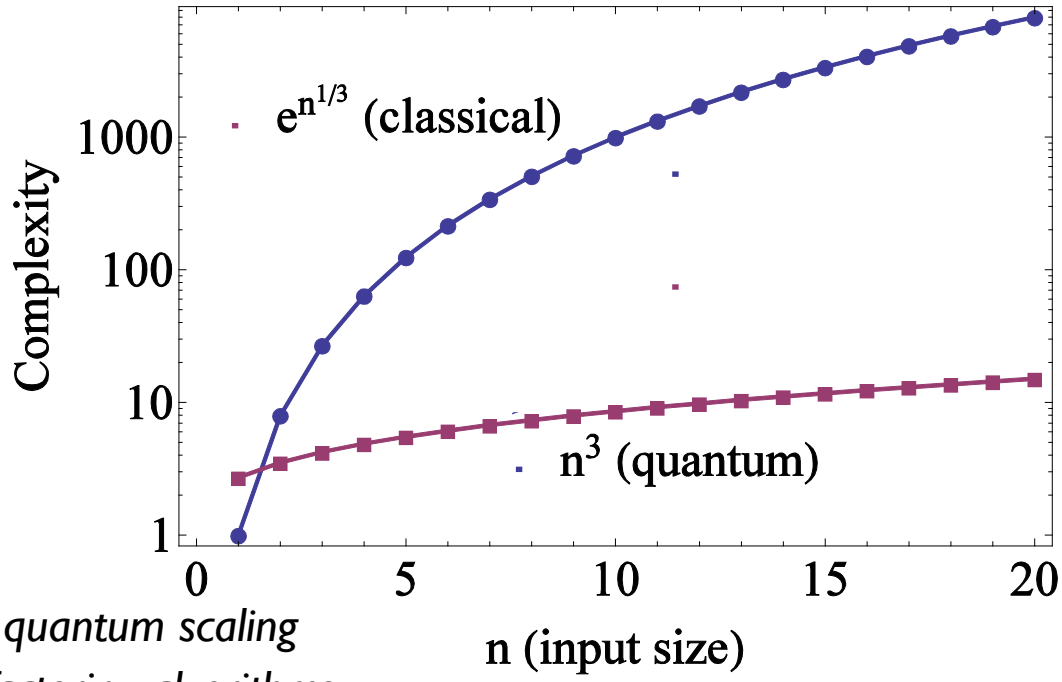
what problems could be solved more efficiently on a quantum computer?

what is a quantum computer?

a quantum computer is a machine that utilizes aspects of quantum physics to perform calculations.

Quantum speedup

- quantum algorithms are expected to be faster in the sense of *scaling with input size* (unlike super computers).
- best known example is Shor's quantum algorithm for integer factorization which scales as n^3 (n is the number of digits of the input number) as opposed to the best known classical algorithm which scales as $e^{c\, n^{1/3}}$, i.e., exponentially slower.
- e.g.: 1000 years on a desktop computer (or, say, 1 year on a supercomputer) vs less than a day on a quantum computer.



classical vs quantum scaling of integer factoring algorithms