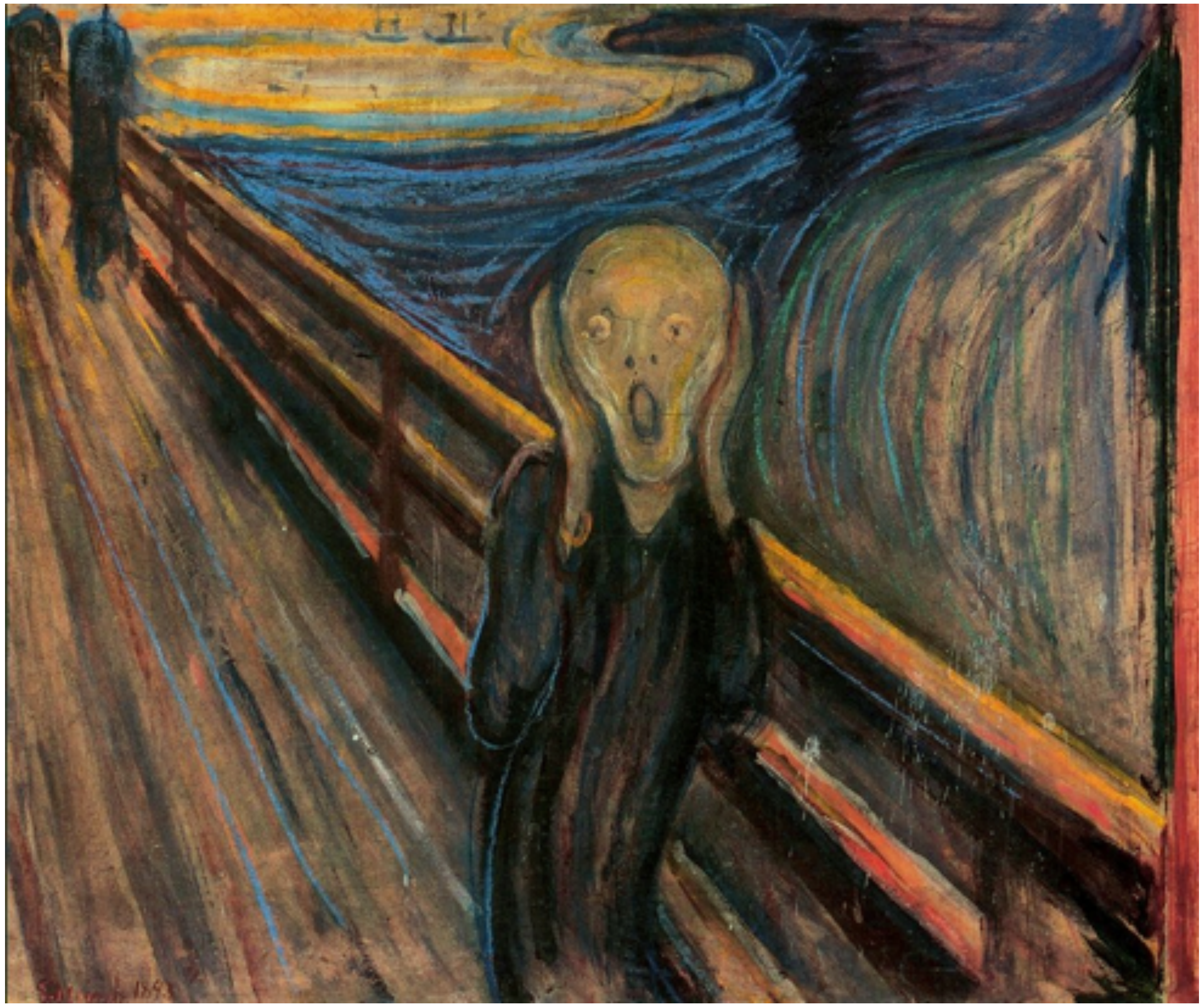
SAND2013-10000P

Screaming Quantum Algorithms



Robin Blume-Kohout

U.S. DEPARTMENT OF ENERGY
Sandia National Laboratories

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Streaming Quantum Algorithms

for Big Data in Small Traps

Robin Blume-Kohout

U.S. DEPARTMENT OF ENERGY
Sandia National Laboratories

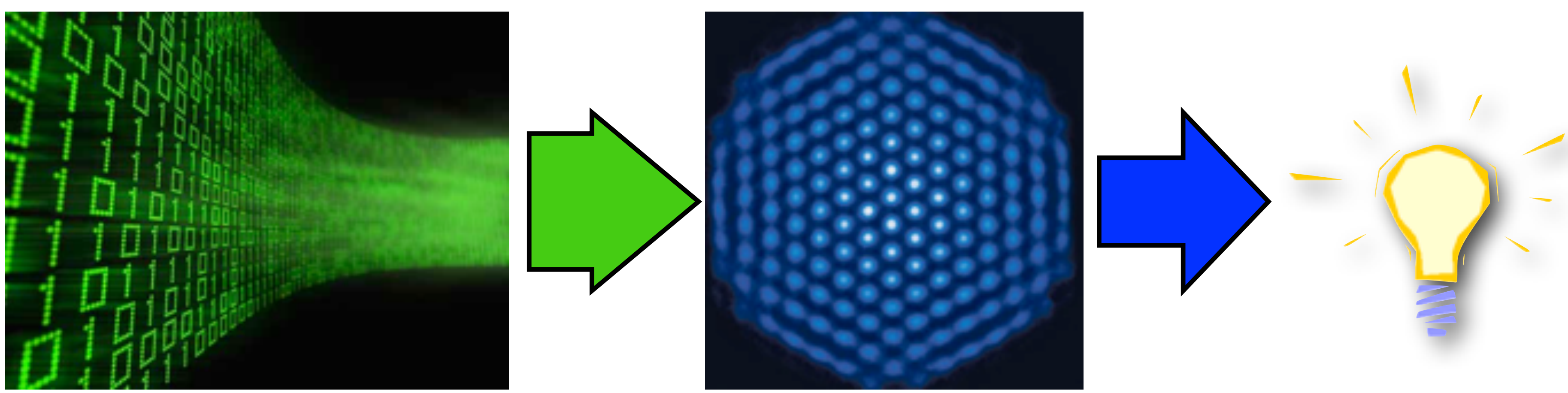
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The Problem

- Quantum computers promise fast solutions to problems. To actually *use* a quantum computer, you need algorithms.
- (Conversely, to run a quantum algorithm, you need a quantum computer).
- Famous quantum algorithms:
  - ☛ Factoring (Shor). Exponential speedup. Breaks RSA.
  - ☛ Search (Grover). Quadratic speedup. Finds solutions.
  - ☛ Simulation. Exponential speedup? Predicts materials.
- But all of these algorithms (except maybe simulation) require loading the whole problem into quantum memory! Which requires kilo/mega/giga-qubits... in the far future...

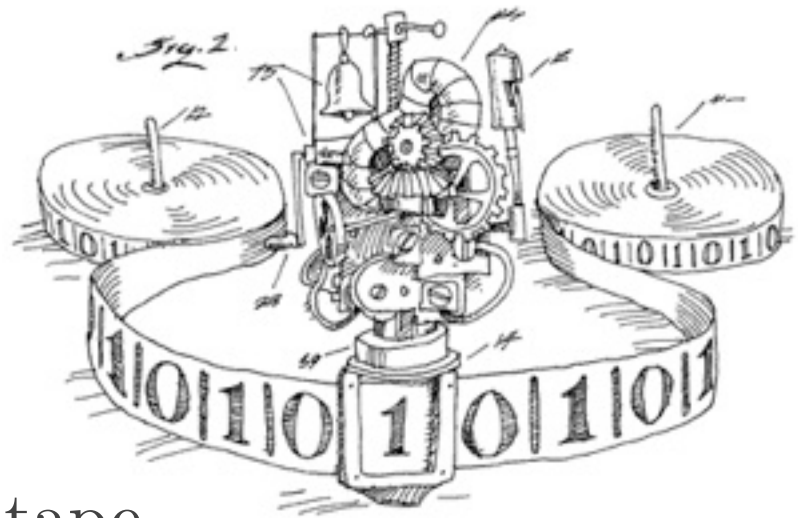
The Solution

- We want to use *small* (30-50 qubit) quantum computers...
  ...to analyze *big* datasets.

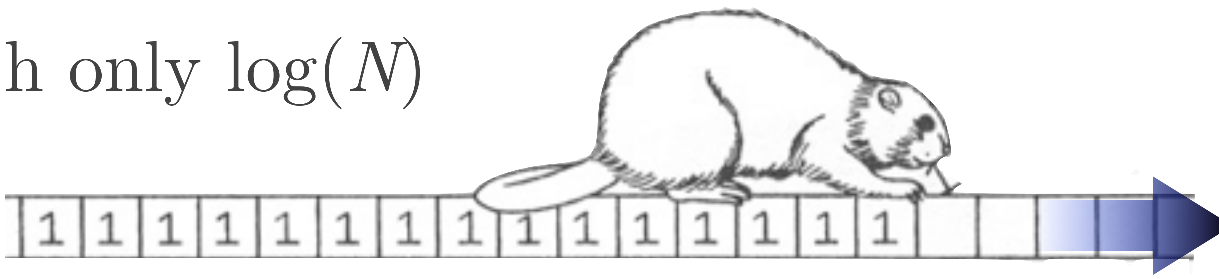


- To do that, we need algorithms that:
  (1) Have *small space complexity* (low memory)
  (2) *Stream* classical input data through the quantum computer
  (3) Can do something better/faster than classical algorithms.

Streaming Algorithms

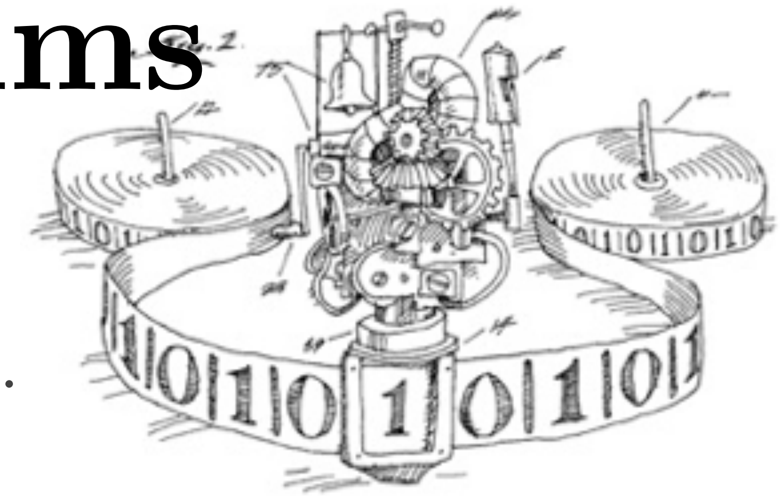


- Most algorithms have access to at *least* enough memory to store the input data.
- But consider a Turing machine with a read-only tape. Algorithms can only use the limited internal memory.
- Turns out, you can do a fair bit with only $\log(N)$ memory! (e.g., all arithmetic).



- In the *streaming data* model, the tape only moves *one way*.
- Applications: *streaming video compression*: ...have you watched a movie lately?
- So what about *quantum* streaming algorithms?

Quantum
Streaming Algorithms



- Really, we're interested in any algorithm that uses very little -- e.g., O(log N) -- quantum space.

- However, streaming algorithms are particularly interesting because:
  (1) It's a natural model for gigantic datasets,
  (2) Reading the tape back and forth can take a *lot* of time.

- So what are we looking for?
  * Things that can be done in quantum logspace, but not classical logspace.
  * Things that can be done *faster* in quantum logspace than classical logspace.
  * Problems where quantum logspace is faster than classical *polynomial* space!
  * <u>Streaming</u> problems that be solved [faster] by quantum computers.

- Current state of knowledge: (1) Quantum *can* do some remarkable things.
  (2) We don't know of any killer apps yet.

Is this a pipe dream?

- No! We already know that
  - (1) Quantum streaming, low-memory algorithms exist, and
  - (2) They can do some things that classical algorithms can't, and
  - (3) Some of those things are potentially relevant!
- Case #1: Streaming Entanglement Concentration / Data Compression

arxiv/0910.5952

Streaming universal distortion-free entanglement concentration

Robin Blume-Kohout,* Sarah Croke,† and Daniel Gottesman‡
Perimeter Institute

This paper presents a streaming (sequential) protocol for universal entanglement concentration at the Shannon bound. Alice and Bob begin with N identical (but unknown) two-qubit pure states, each containing E ebits of entanglement. They each run a reversible algorithm on their qubits, and end up with Y perfect EPR pairs, where $Y = NE \pm O(\sqrt{N})$. Our protocol is streaming, so the N input systems are fed in one at a time, and perfect EPR pairs start popping out almost immediately. It matches the optimal block protocol exactly at each stage, so the average yield after n inputs is $\langle Y \rangle = nE - O(\log n)$. So, somewhat surprisingly, there is no tradeoff between yield and lag – our protocol optimizes both. In contrast, the optimal N-qubit block protocol achieves the same yield, but since no EPR pairs are produced until the entire input block is read, its lag is $O(N)$. Finally, our algorithm runs in $O(\log N)$ space, so a lot of entanglement can be efficiently concentrated using a very small (e.g., current or near-future technology) quantum processor. Along the way, we find an optimal streaming protocol for extracting randomness from classical i.i.d. sources and a more space-efficient implementation of the Schur transform.

Is this a pipe dream?

- No! We already know that
  - (1) Quantum streaming, low-memory algorithms exist, and
  - (2) They can do some things that classical algorithms can't, and
  - (3) Some of those things are potentially relevant!

- Case #2: Francois Le Gall's "Disjointness Tester"

Theory of Computing Systems
August 2009, Volume 45, Issue 2, pp 188-202

Exponential Separation of Quantum and Classical Online Space Complexity

François Le Gall

✦**Input:** Two m-bit strings $\boldsymbol{x}$ and $\boldsymbol{y}$ are fed in, $m^{1/2}$ times in a row.

✦**Problem**: Is there any index i for which $x_i = y_i = 1$?

✦**Classically**: The computer must have at least $m^{1/2}$ bits of memory!

✦**Quantumly**: Problem can be solved in only $\log(m)$ qubits of memory!

Is this a pipe dream?

- No! We already know that
  (1) Quantum streaming, low-memory algorithms exist, and
  (2) They can do some things that classical algorithms can't, and
  (3) Some of those things are potentially relevant!

- Entanglement concentration:
  + does something that no classical machine can
  + operates on streaming data in quantum logspace
  - only useful for *quantum* input data.

- Le Gall's disjointness tester:
  + exponential separation between classical & quantum *space* (memory)
  + operates on very long *classical* strings
  - only a quadratic improvement in *time*
  - contrived problem

- So are there *useful* quantum streaming algorithms? Open question!

General plan of research

- **Stage 1: Determine what's known to be possible/impossible.**
  - Apply known results to:
    (i) the tasks we care about (big data),
    (ii) the resources we intend to develop (trapped ions)
  - Identify critical open questions.

- **Stage 2: Work to answer broad open questions.**
  - What is possible?
  - How many qubits/gates/resources are necessary to achieve specific (useful!) levels of performance?
  - Where are niche applications for small quantum computers?

- **Stage 3: Seek constructive algorithms and/or impossibility theorems for specific tasks of interest.**

U.S. DEPARTMENT OF ENERGY Sandia National Laboratories
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

What do we hope for?

- **Questions for the near term:**
  - For what types of problem can quantum computers provide *any* advantage in the streaming model?
  - For what problems is there *known* to be no advantage?
  - Could there be problems that a quantum logspace computer could solve faster than *any* classical computer?
  - Can we do streaming period-finding (the core of Shor)?

- **Example long-term "goals":**
  - Find a non-contrived problem that a 50-qubit quantum computer could solve substantially faster.
  - Show how to do streaming period-finding, and find a practical application for it.
  - Prove lower bounds on the quantum memory required to speed up any problem.
  - Find algorithms that work w/out [much] error correction

U.S. DEPARTMENT OF ENERGY
Sandia National Laboratories
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.