QuRE: The Quantum Resource Estimator Toolbox

Martin Suchara (IBM Research)

In collaboration with: Arvin Faruque, Ching-Yi Lai, Gerardo Paz, Fred Chong, and John Kubiatowicz

October 9, 2013

Why Quantum Computer Resource Estimator?

- Building a practical quantum computer is very difficult
- Goal: investigate impact of design choices on the performance of the computer without building one
  - Hardware: speed vs. reliability tradeoff
  - Error correction: choosing good strategies
  - Algorithms: which are efficient?
- This work: flexible configurable estimation tool

Inputs and Outputs of the QuRE Toolbox

Algorithm Specs
- # of logical qubits
- # of logical gates
- Circuit parallelism

Analysis of Error Correction
- Estimate cost of each logical operation as a function of error correction "strength"

Technology Specs
- Gate times and fidelities
- Memory error rates

Automated Resource Estimate
- Find out how strong error correction guarantees target success probability
- Estimate number of physical qubits, running time, physical gate and instruction count, etc.

QuRE Analyzes a Variety of Realistic Scenarios

- 7 quantum algorithms
- 4 quantum error correcting codes
- 12 physical technologies
- This talk
  - Overview of resource estimation methodology and highlights of our results

Overview

I. Properties of quantum technologies and algorithms

II. Estimation methodology – overhead of concatenated error correction codes

III. Estimation methodology – overhead of topological error correction codes

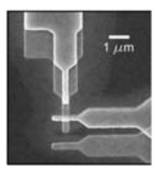
IV. Examples of estimates obtained with QuRE

How Quantum Computers Work

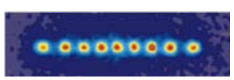
- Quantum instead of binary information
  - Quantum state $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, not just 0 or 1
- Quantum computers must be able to initialize, store, manipulate and measure quantum states
- Operations and memory storage must be reliable

A Number of Competing Candidate Technologies

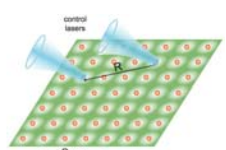
- Superconducting qubits
  - Josephson Junctions between superconducting electrodes



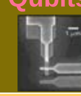
- Ion traps



  - Ions trapped in electromagnetic field, gates performed by applying lasers
- Neutral atoms
  - Ultracold atoms trapped by light waves in an optical lattice



Properties of Quantum Technologies: Gate Times and Errors

	Supercond. Qubits 	Ion Traps 	Neutral Atoms 
Average Gate Time (ns)	25	32,000	19,000
Worst Gate Error	1.00×10^{-5}	3.19×10^{-9}	1.47×10^{-3}
Memory Error	1.00×10^{-5}	2.52×10^{-12}	not available

- Ion traps slower but more reliable than superconductors
- Neutral atoms slower and error prone

The Best Known Quantum Algorithm

- Shor's factoring algorithm
  - Find prime factors of integer N
  - Quantum algorithm runs in polynomial time
  - Can be used to break public-key cryptography (RSA)
  - Algorithm uses quantum Fourier transform and modular exponentiation

Shor's Factoring Algorithm – Logical Gate Count

- Factor a 1024-bit number
- Algorithm needs approximately 1.68 x 10^8 Toffoli gates and 6,144 logical qubits (*Jones et al., 2012*)

Gate	Occurrences	Parallelization Factor
CNOT	1.18 x 10^9	1
Hadamard	3.36 x 10^8	1
T or T†	1.18 x 10^9	2.33
Other gates	negligible	

Overview

I. Properties of quantum technologies and algorithms

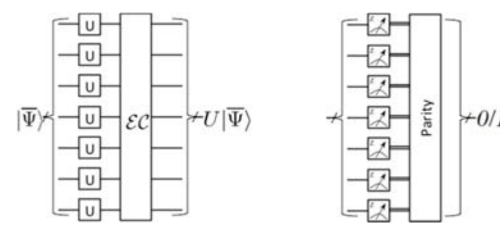
II. Estimation methodology – overhead of concatenated error correction codes

III. Estimation methodology – overhead of topological error correction codes

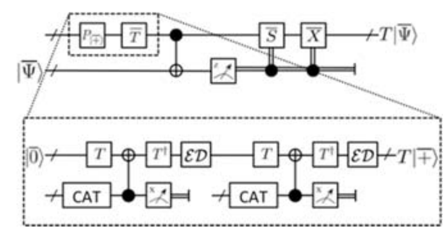
IV. Examples of estimates obtained with QuRE

Steane [[7,1,3]] Concatenated Error Correction Code

- 7 data qubits encode a single logical qubit
- Most operations transversal:

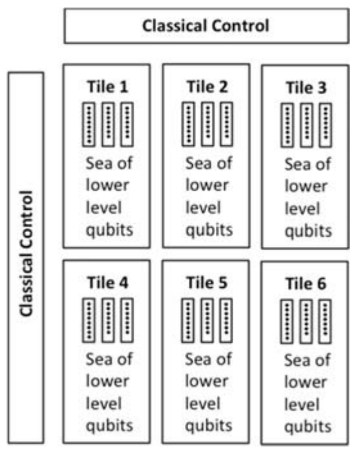


- Non-transversal T gate:



Tiled Qubit Layout for Concatenated Codes

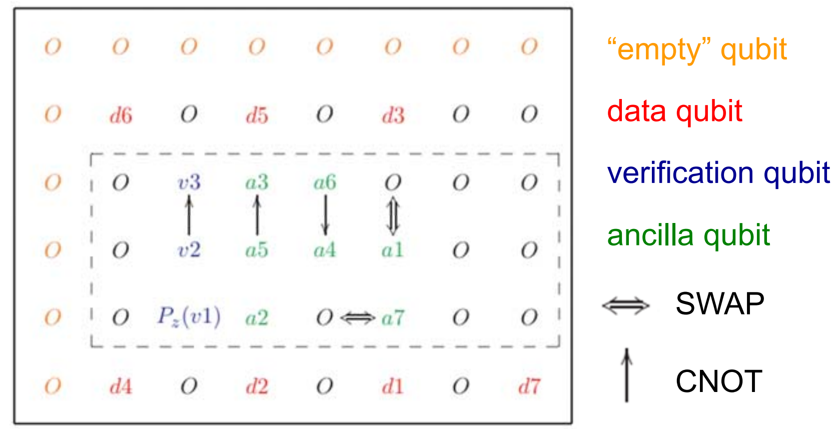
- Each logical qubit is stored in a separate tile



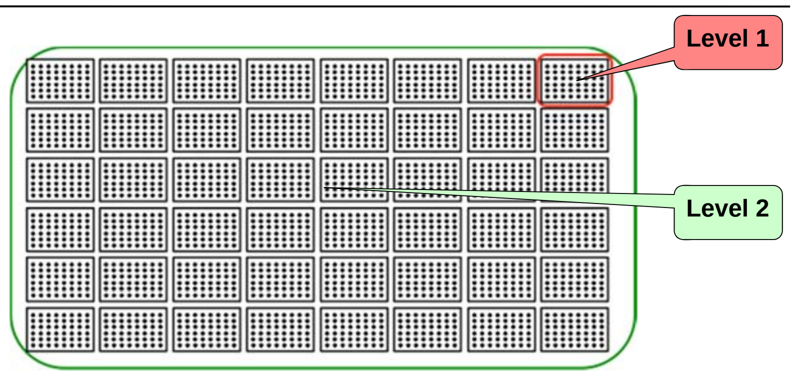
- Tiles arranged in 2-D
- Supported operations:
  - Error correct a tile
  - Apply fault-tolerant operation
- Tiles must contain enough data and ancilla qubits

Optimized Layout in Each Tile

(*Svore et al., 2006*)



Tiles Have a Hierarchical Structure that Allows Code Concatenation



- Sufficient number of concatenations to achieve constant probability of success of computation

Counting the Gates and Computation Time

- For each logical operation (CNOT, error correction, Paulis, S, T, measurement, etc.)
  - Count number of elementary gates
  - Count time taking parallelism into account
- Methodology: recursive equations that follow the concatenated structure

$$ops(X_{(m)}) = 7X_{(m-1)} + \mathcal{EC}_{(m)}$$

$$time(X_{(m)}) = X_{(m-1)} + \mathcal{EC}_{(m)}$$

$$ops(\mathcal{EC}_{(m)}) = 14hCNOT_{(m-1)} + 28vCNOT_{(m-1)} + 7H_{(m-1)} + 18hSWAP_{(m-1)} + 15vSWAP_{(m-1)} + 8P_{|+\rangle(m-1)} + 12P_{|0\rangle(m-1)} + 20\mathcal{M}_{X(m-1)}$$

Overview

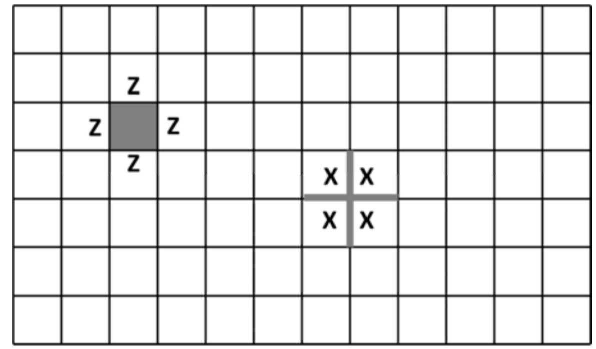
I. Properties of quantum technologies and algorithms

II. Estimation methodology – overhead of concatenated error correction codes

III. Estimation methodology – overhead of topological error correction codes

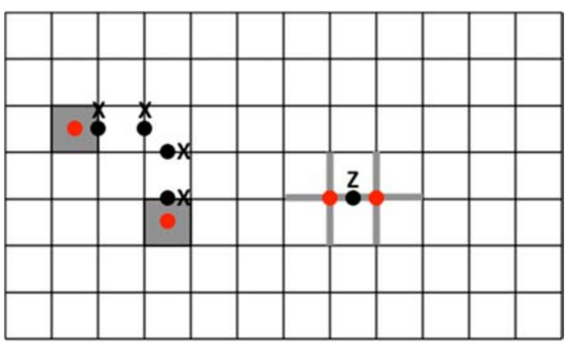
IV. Examples of estimates obtained with QuRE

Topological Quantum Memory



- Physical qubits on links in the lattice
- Measuring the shown "check" operators yields error syndromes

Syndromes Caused by Errors

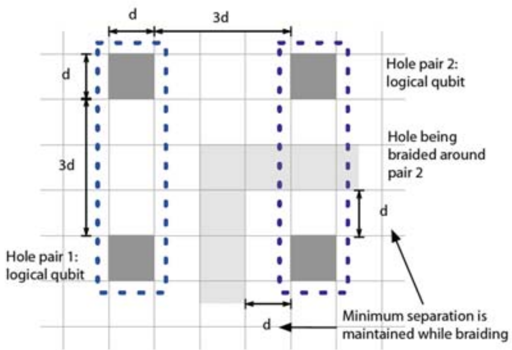


- Guess the most likely error consistent with observed syndromes
- Error correction performed continuously

Tiles Represent Logical Qubits

- Each logical qubit represented by a pair of holes
- CNOT gates performed by moving holes around each other

Code Distance Determines Fault Tolerance and Size of the Tiles



N: number of gates

p: physical error rate

C_1, C_2: constants

$P_{th}\approx 0.1$: error correction threshold

- Distance sufficient for high success probability:

$$\frac{0.5}{N} \geq C_1 \left(C_2 \frac{p}{p_{th}}\right)^{\lfloor \frac{d+1}{2} \rfloor}$$ (*Jones et al., 2012*)

Counting the Qubits and Gates

- Qubit count: multiply number of tiles and size of tile
- Gate count:
  - Calculate total running time T
  - Calculate number of gates required to error correct the entire surface during interval T
  - Estimate the small number of additional gates required by logical operations

Overview

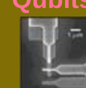
I. Properties of quantum technologies and algorithms

II. Estimation methodology – overhead of concatenated error correction codes

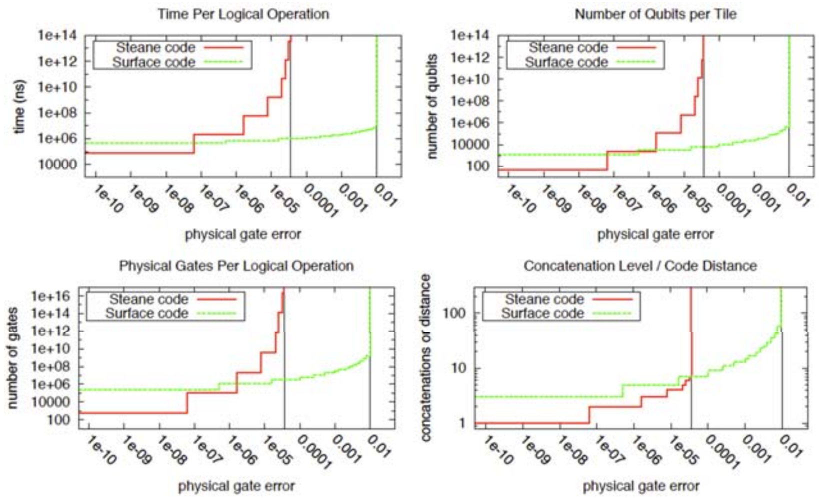
III. Estimation methodology – overhead of topological error correction codes

IV. Examples of estimates obtained with QuRE

Numerical Results – Shor's Factoring Algorithm, Three Technologies

	Neutral Atoms $e = 1 \times 10^{-3}$ $t = 19{,}000$ ns 	Supercond. Qubits $e = 1 \times 10^{-5}$ $t = 25$ ns 	Ion Traps $e = 1 \times 10^{-9}$ $t = 32{,}000$ ns 	
Surface Code	2.6 years	**10.8 hours**	2.2 years	Time
	5.3 x 10^8	**4.6 x 10^7**	1.4 x 10^8	Qubits
	1.0 x 10^{21}	**2.6 x 10^{19}**	5.1 x 10^{19}	Gates
Steane Code	-	5.1 years	**58 days**	Time
	-	2.7 x 10^{12}	**4.6 x 10^5**	Qubits
	-	1.2 x 10^{32}	**4.1 x 10^{18}**	Gates

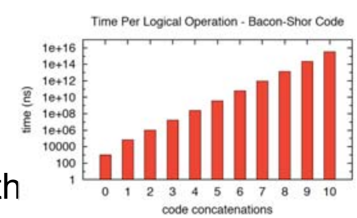
Abstract Technology (1 μs gates) with Varying Physical Error Rate



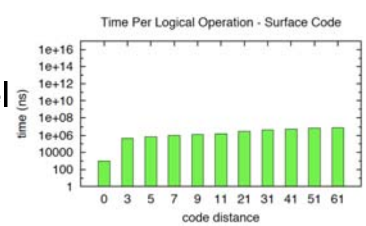
For low error rates concatenated codes outperform topological codes. Why?

The Topological and Concatenated Code Families are Very Different

- Concatenated codes
  - Lightweight with 1-2 levels of concatenation
  - Exponential overhead with additional concatenations

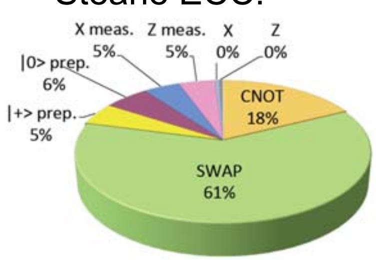


- Topological codes
  - Operations highly parallel
  - Moderate overhead with increasing code distance

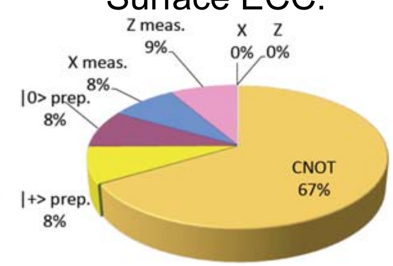


Qualitative Difference in Gate Composition

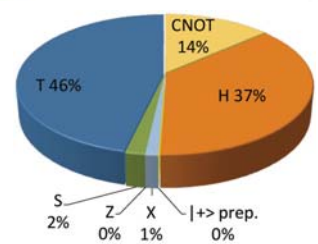
Steane ECC:



Surface ECC:



Logical circuit:



Resource Estimates Useful for Identifying Topics for Future Work

- Low parallelism of studied circuits
  - How to exploit parallelism and move some operations off the critical path?
- Decomposition of arbitrary rotations very costly
  - More efficient techniques?
- Costly T and CNOT gates dominate
  - Circuit transformations to avoid these gates?
  - More efficient offline implementation?

Conclusion

- QuRE is an automated tool that quickly estimates the properties of the future quantum computer
- Reports a number of quantities including gate count, execution time, and number of qubits
- Is easily extendable for new technologies and algorithms
- Allows to identify sources of high overhead and quickly asses the effect of suggested improvements

Thank You!