Full-State Quantum Circuit Simulation by Using Data Compression

Xin-Chuan Wu[1], Sheng Di[2], Emma Maitreyee Dasgupta[1], Franck Cappello[2], Hal Finkel[3], Yuri Alexeev[3], Frederic T. Chong[1]

[1]Department of Computer Science, University of Chicago

[2]Mathematics and Computer Science Division, Argonne National Laboratory

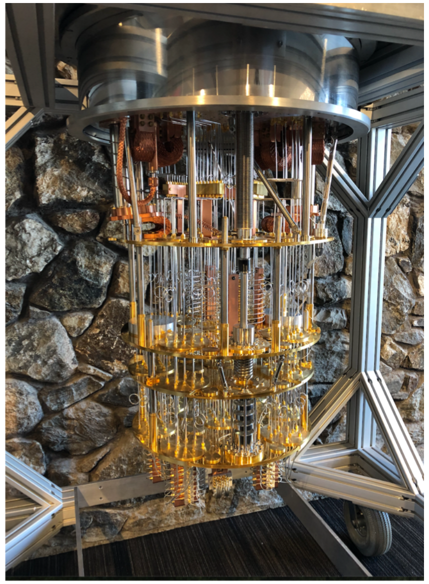
[3]Argonne Leadership Computing Facility, Argonne National Laboratory

Nov. 21

Why Quantum Circuit Simulation?

- Quantum systems: extremely sensitive to environmental effects
  - IBM Q 20 Tokyo

Average measurements	
Frequency (GHz)	4.97
T1 (μs)	79.72
T2 (μs)	52.27
Gate error (10^{-3})	1.71
Readout error (10^{-2})	7.51



- Simulation of quantum circuits
  - Validate quantum circuits
  - Quantify the circuit fidelity on real quantum machines
  - Assess performance of new quantum algorithms
  - Debug quantum program

Quantum Software Debugging

- Statistical assertions for validating patterns and finding bugs in quantum programs [ISCA'19]

- On a real quantum computer
  - Performing measurement for each assertion

- On a quantum circuit simulation
  - Running the quantum program without restarting

Quantum Circuit Simulation by Using Data Compression

- How to perform quantum circuit simulation?
- Our main idea and implementation
- Evaluation

What is Quantum Circuit Simulation?

- Quantum circuit simulation: quantum state amplitudes.
  - Using classical computing systems to simulate quantum computers

- 1-qubit system
  - $|\Psi\rangle = a_0|0\rangle + a_1|1\rangle$
- 2-qubit system
  - $|\Psi\rangle = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle$
- n-qubit system
  - $|\Psi\rangle = a_0|0\ldots000\rangle + a_1|0\ldots001\rangle + \ldots + a_{2^n-1}|1\ldots111\rangle$
  - For n-qubit systems, 2^n amplitudes

$$\longrightarrow \quad |\Psi\rangle = \begin{pmatrix} a_{0\ldots000} \\ a_{0\ldots001} \\ \ldots \\ a_{1\ldots111} \end{pmatrix}$$

- Simulation: $|\Psi_{t+1}\rangle = A_t\,|\Psi_t\rangle$, for $t = 0, \ldots, d$ at each layer
  - A_t is a unitary matrix
  - d is the depth of the circuit

Challenges of Quantum Circuit Simulation

- For n-qubit systems: 2^n amplitudes
  - Double-precision complex number: 16 Bytes
  - State vector size: 2^{n+4} Bytes
  - People believe it is difficult to classically simulate a 50-qubit quantum computer
    - 50-qubit system simulation: 16PB (2^{54} Bytes)
- List of supercomputers and the max size they can simulate

System	Memory (PB)	Max Qubits
Summit	2.8	47
Sierra	1.38	46
Sunway TaihuLight	1.3	46
Theta	0.8	45

Full-State Simulation

- Schrödinger Algorithm
- Keep the full state vector in memory
- Space: 2^{n+4} Bytes
- Circuit depth: High

Year	Reference	Qubits
2016	qHiPSTER: the quantum high performance software testing environment	42
2017	0.5 petabyte simulation of a 45-qubit quantum circuit	45
2018	Quantum supremacy circuit simulation on Sunway taihuLight	49

Partial State Simulation

- Feynman paths integral
  - Calculate one amplitude by following all the paths from the final state to the initial state.
- Tensor network contraction
  - The time and space cost for contracting such tensor networks is exponential with the treewidth of the underlying graphs.
- Circuit depth: Low

Year	Reference	Qubits	Amplitudes	Fidelity
2017	Breaking the 49-qubit barrier in the simulation of quantum circuits	49	All	100%
2017	Simulation of low-depth quantum circuits as complex undirected graphical models	56	1	-
2018	Classical simulation of intermediate-size quantum circuits	144	1	5%
2018	Quantum supremacy is both closer and farther than it appears	56	1	0.5%

Approximate simulation

Quantum Circuit Simulation by Using Data Compression

- How to perform quantum circuit simulation?
- Our main idea and implementation
- Evaluation

Our Simulation

- Goal: For general circuits, increase the simulation size
  - Full state simulation
  - Trade time for space complexity

- A new method for Schrödinger algorithm simulation
  - Applying data compression to state vectors

- Data compression
  - Lossless
  - Lossy – approximate simulation

Main Contributions of Our Work

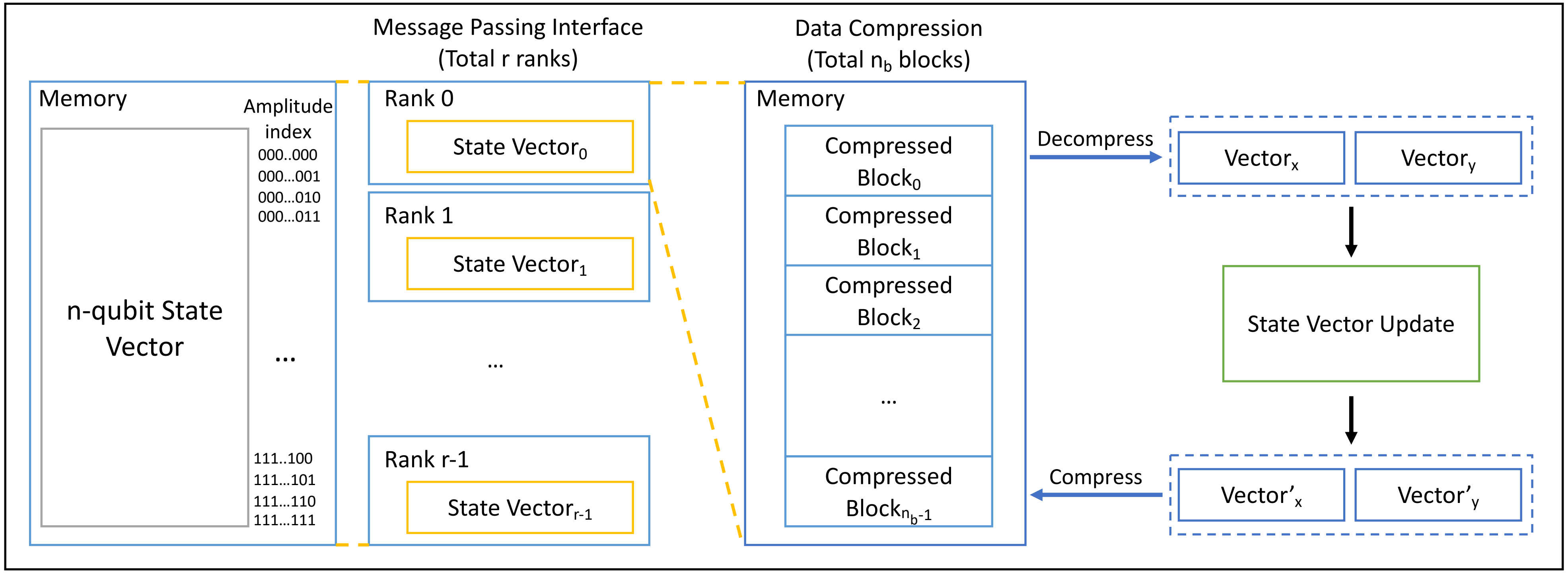
- We provide one more option in the set of tools to scale quantum circuit simulation.

- We present a new technique to reduce memory requirements of full-state simulations by using data compression.

- We implement our general quantum circuit simulation framework on the Theta supercomputer at Argonne National Laboratory.

Simulation Overview

- A gate operation:
  - Decompress two corresponding blocks to update, and then compress the blocks
  - Move to the next two corresponding blocks, repeat until all blocks have been updated

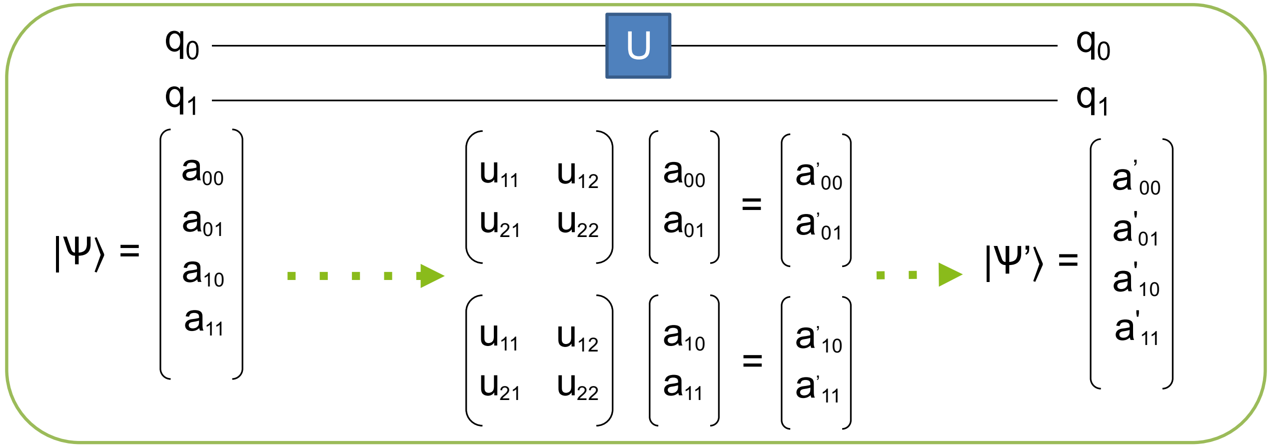


Gate Operation

- $|\Psi_{t+1}\rangle = A_t\, |\Psi_t\rangle$

  $$A = I \otimes I \otimes \ldots \otimes U \otimes \ldots \otimes I \otimes I$$

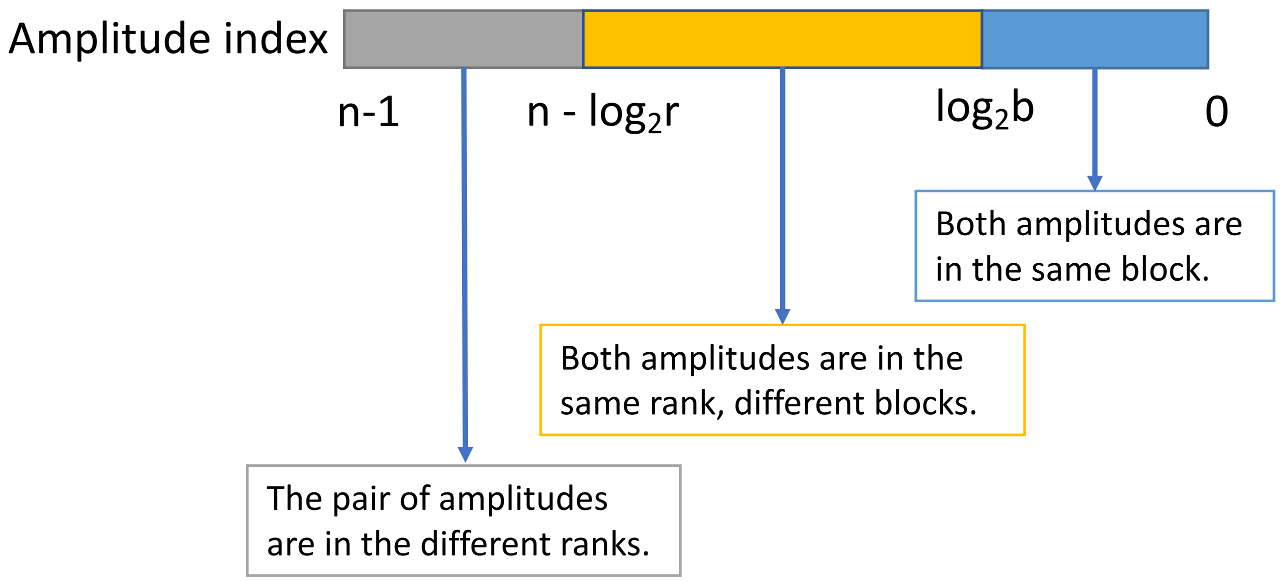
  $$U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$$



- We do not need to construct the entire A.
  - For example, applying a single-qubit gate to the first qubit is equivalent to applying U to every pair of amplitudes, whose indices have 0 and 1 in the first bit, while all other bits remain the same.

Single-Qubit Gate

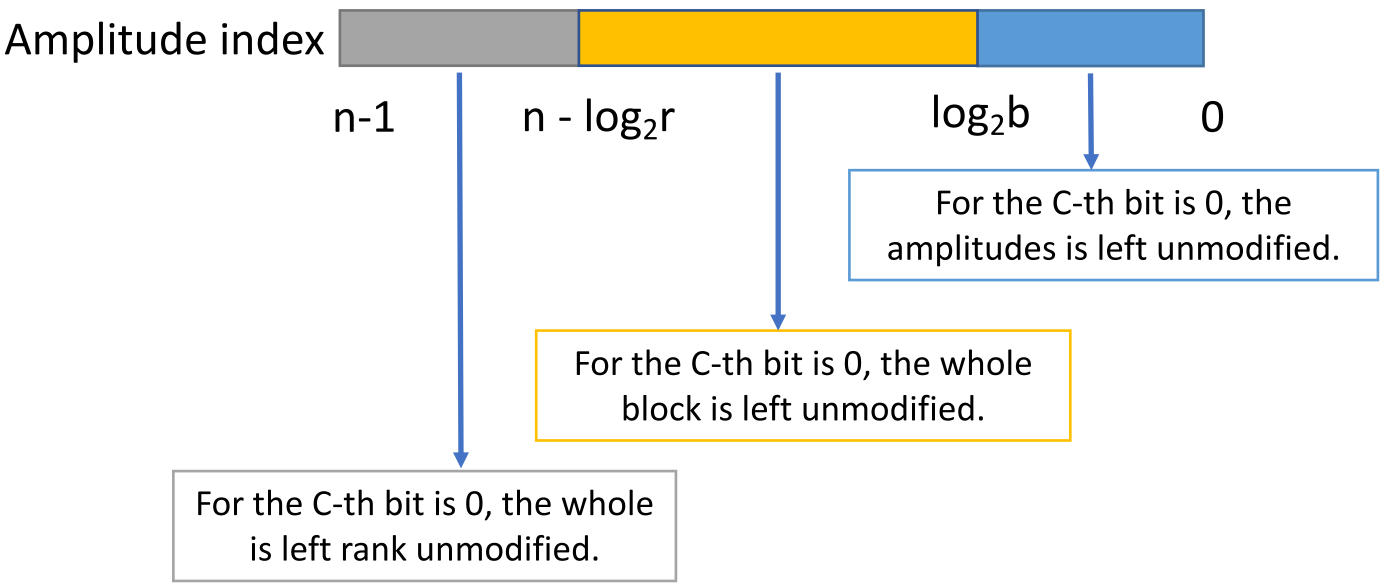
- *n* qubits, *r* ranks, and each block contains *b* amplitudes
- Get the pair of blocks whose indices have 0 and 1 in the target position



n: # of qubits
r: # of ranks
b: # of amplitudes in a block

Two-Qubit Gate

- In a control-U gate, control qubit position: C-th qubit
- If the C-th qubit is 1, apply U to k-th qubit; otherwise left unmodified.



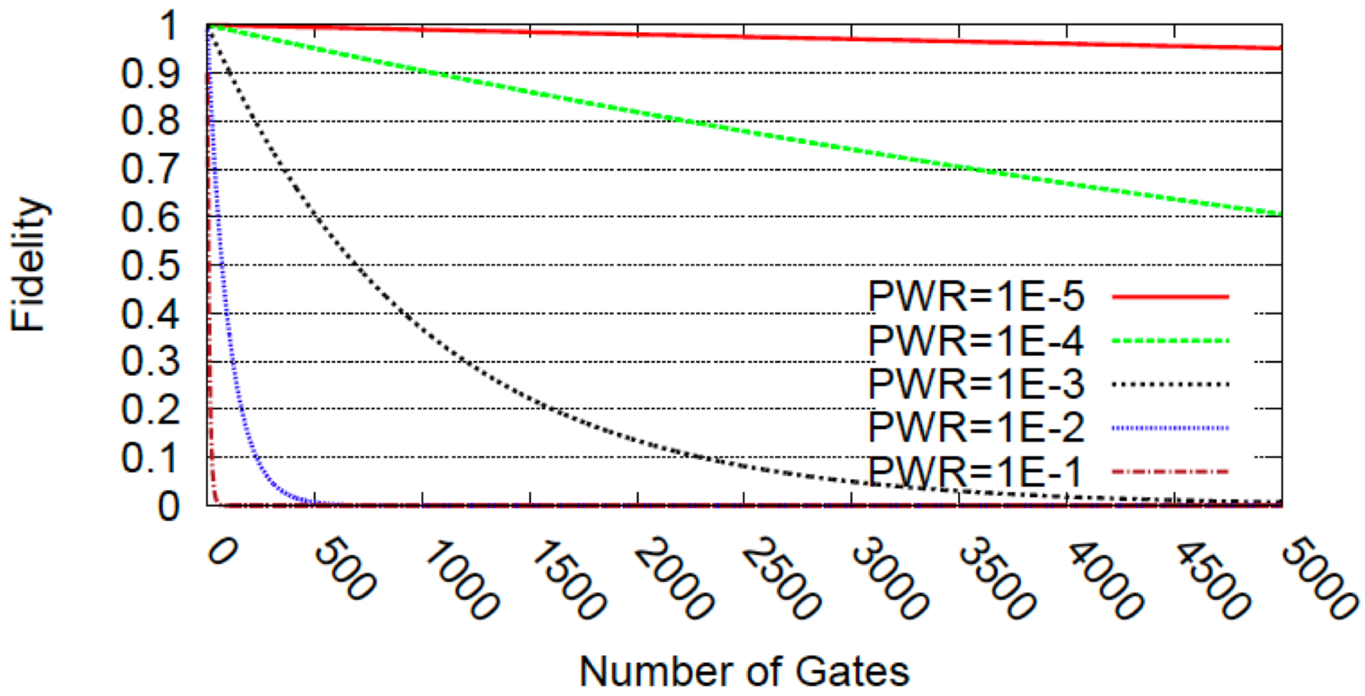
n: # of qubits
r: # of ranks
b: # of amplitudes in a block

Compression Techniques

- Lossless: Zstd
- Lossy: SZ
  - Allowing user-controlled loss of accuracy
  - Set the error bound, denoted δ
  - The decompressed data D_i' must be in the range $[D_i (1 - \delta), D_i (1 + \delta)]$
    - where D_i' is referred as the decompressed value and D_i is the original data value.
  - SZ can compress 1-D dataset efficiently.

Estimated Fidelity

- Simulation starting with lossless compression
- Larger error → higher compression ratio, lower fidelity



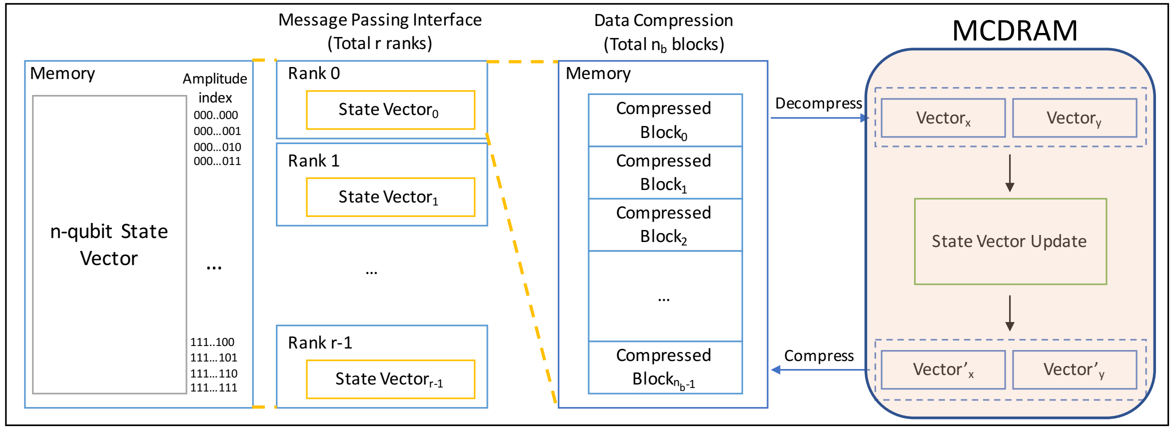
Optimizations

- MCDRAM memory configuration
- SW compressed block record
- Simulation checkpoint

MCDRAM Memory Configuration

- Multi-Channel DRAM
  - High bandwidth (~ 4x more than DDR4)
  - Low capacity (up to 16GB)
  - Packaged with the Knights Landing Silicon (KNL)

- Decompress state vectors to MCDRAM



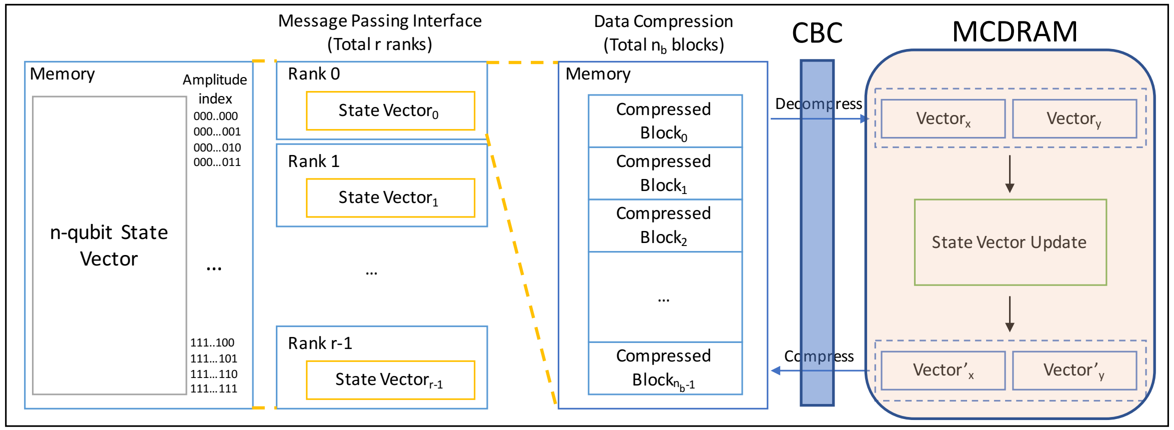
SW Compressed Block Record

- Quantum circuits may have repeated amplitudes.
- A record line

OP	CB_1	CB_2	CB'_1	CB'_2

- 64 lines per rank
  - $OP_t == OP \&\& CB_x == CB_1 \&\& CB_y == CB_2$

  $\rightarrow CB'_x = CB'_1, CB'_y = CB'_2$



Simulation Checkpoint

- Our simulation is allowed to dump full state vectors at any time steps
  - Supercomputing systems usually have a 24-hour wall-time limit
  - Compressed format
    - Reduce disk I/O time
  - Software debugging
    - Recover a state vector without re-run the circuit

Quantum Circuit Simulation by Using Data Compression

- How to perform quantum circuit simulation?
- Our main idea and implementation
- Evaluation

Evaluation: Benchmarks

- Grover
  - Database search algorithm
  - 61, 59, and 47 qubits
- Random circuit sampling
  - Proposed by Google to show the quantum supremacy
  - 45 qubits, 42 qubits, 36 qubits, and 35 qubits
- QAOA
  - Quantum approximate optimization algorithm
  - 43 qubits and 42 qubits
- QFT
  - Quantum Fourier Transform
  - 36 qubits

Experimental Setup

- Single-node environment
  - JLSE system at Argonne
  - 64-core Intel Xeon Phi processor 7210 KNL
  - 16GB MCDRAM
  - 192GB DDR4 memory

- Multi-node environment
  - Theta supercomputer at Argonne
  - 4,392 nodes
  - 64-core Intel Xeon Phi processor 7230 KNL
  - 16GB MCDRAM
  - 192GB DDR4 memory

Experimental Results

- 61-qubit Grover's algorithm simulation: 32EB → 768TB
- Our approach can simulate deep circuits, like QFT
- Simulate more qubits with the limited memory resource

Benchmark	Grover			Random Circuit Sampling				QAOA			QFT
Number of Qubits (Memory Requirement)	61 (32 EB)	59 (8 EB)	47 (2 PB)	5×9 (512 TB)	6×7 (64 TB)	6×6 (1 TB)	7×5 (512 GB)	45 (512TB)	43 (128 TB)	42 (64 TB)	36 (1 TB)
Number of Gates	314	310	305	227	261	165	208	394	344	336	3258
Number of Nodes	4096	4096	128	1024	128	1	1	1024	256	128	1
Total System Memory (Sys Mem / Req.)	768 TB (0.002%)	768 TB (0.009%)	24 TB (1.17%)	192 TB (37.5%)	24 TB (37.5%)	192 GB (18.75%)	192 GB (37.5%)	192TB (37.5%)	48 TB (37.5%)	24 TB (37.5%)	192 GB (18.75%)
Total Time (Hour)	8.14	3.48	0.49	4.87	8.64	7.96	6.23	13.34	5.83	8.65	78.98
Compression Time	1.87%	4.59%	2.04%	55.79%	40.26%	59.10%	58.57%	50.66%	44.97%	41.02%	57.86%
Decompression Time	1.87%	3.73%	4.08%	31.47%	22.19%	33.78%	30.59%	26.46%	27.64%	25.52%	37.68%
Communication Time	32.7%	20.98%	36.73%	0.12%	0.57%	0.02%	0.03%	3.03%	0.22%	0.23%	2.56%
Computation Time	63.47%	70.70%	57.15%	12.60%	36.97%	7.08%	10.8%	19.84%	27.16%	33.22%	1.9%
Time per Gate (Sec)	93.34	40.49	5.78	64.69	119.22	173.65	107.86	121.91	61.02	92.64	87.27
Simulation Fidelity	0.996	0.996	1	0.987	0.993	0.933	0.985	0.895	0.999	0.999	0.962
Compression Ratio	7.39×10^4	8.26×10^4	1.06×10^4	6.03	9.40	8.16	10.05	5.38	4.85	9.25	21.34

Increasing Simulation Size

- Compression ratio: 4.85x ~ 82,600x
  - Increasing the number of qubits in the simulation: $\log_2(4.85)$ ~ $\log_2(82600)$
  - +2 ~ 16 qubits

- List of supercomputers and the max size they can simulate

System	Memory (PB)	Max Qubits	Max Qubits
Summit	2.8	47	49 - 63
Sierra	1.38	[illegible]	48 - 62
Sunway TaihuLight	1.3	[illegible]	48 - 62
Theta	0.8	45	47 - 61

+2 ~ 16 Qubits

Conclusion

- Full-state simulation with data compression
- New method for Schrödinger-style simulation to trade time for space
    - Data compression
- The compression ratio results show
    - Increase the simulation size by 2 to 16 qubits

Full-State Quantum Circuit Simulation by Using Data Compression

Xin-Chuan Wu[1], Sheng Di[2], Emma Maitreyee Dasgupta[1], Franck Cappello[2], Hal Finkel[3], Yuri Alexeev[3], Frederic T. Chong[1]

[1]Department of Computer Science, University of Chicago

[2]Mathematics and Computer Science Division, Argonne National Laboratory

[3]Argonne Leadership Computing Facility, Argonne National Laboratory

Nov. 21

Thank You

Acknowledgment:

This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

This research was supported by the Exascale Computing Project (ECP), Project Number: 17-SC-20-SC, a collaborative effort of two DOE organizations the Office of Science and the National Nuclear Security Administration, responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering and early testbed platforms, to support the nations exascale computing imperative.

The material was supported by the U.S. Department of Energy, Office of Science, and supported by the National Science Foundation under Grant No. 1619253.

This work is funded in part by EPiQC, an NSF Expedition in Computing, under grant CCF-1730449.

This work is also funded in part by NSF PHY-1818914 and a research gift from Intel.

https://www.epiqc.cs.uchicago.edu/