Optimization for quantum computer simulation

Naoki Yoshioka[1], Hajime Inaoka[1], Nobuyasu ito[1], Fengping Jin[2], Kristel Michielsen[2,3], and Hans De Raedt[4]

[1]RIKEN Center for Computational Science, [2]Forschungzentrum Jülich, [3]RWTH Aachen University, [4]University of Groningen

Introduction

- Quantum circuits, or quantum logic gates, are studied extensively in these decades.
  - Quantum computers have appeared, such as, IBM Q.
- Simulations of quantum circuits in classical supercomputers [1–3].
  - Useful because it is still limited to obtain large resources of real quantum circuit computers.
- *braket* [4], a software and a C++ template library to simulate quantum cimputers.
  - Propose optimization techniques; a page method and a simple rule of initial permutation of qubits.
- Report performance of our simulations in the K computer up to 45 qubits
  - Using double precision floating point number. Requires 0.5 PB at most.

Quantum gates

- A state vector $|\Phi\rangle$ is described with 2^N complex coefficients $a(n)$;

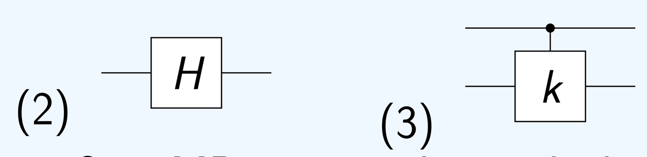
$$|\Phi\rangle = \sum_{n=0}^{2^N-1} a(n)\,|n\rangle\,, \qquad \sum_{n=0}^{2^N-1} |a(n)|^2 = 1 \tag{1}$$

where $|n\rangle \equiv |n_N\rangle \otimes \cdots \otimes |n_1\rangle$, $n = n_N \cdots n_1$, and $n_k \in \{0, 1\}$.

- Quantum gates $\Leftrightarrow$ unitary operators on $|\Phi\rangle$.
- Ex.) Hadamard gate [Eq. (2)] and controlled phase-shift gate [Eq. (3)]

$$H\big(a(0)\,|0\rangle + a(1)\,|1\rangle\big) = \frac{a(0)+a(1)}{\sqrt{2}}\,|0\rangle + \frac{a(0)-a(1)}{\sqrt{2}}\,|1\rangle\,, \tag{2}$$

$$R(2\pi/2^k)\big(a(00)\,|00\rangle + a(01)\,|01\rangle + a(10)\,|10\rangle + a(11)\,|11\rangle\big) = a(00)\,|00\rangle + a(01)\,|01\rangle + a(10)\,|10\rangle + e^{2\pi i/2^k} a(11)\,|11\rangle\,. \tag{3}$$



- Easily parallelized by using OpenMP or any other multithreading libraries.

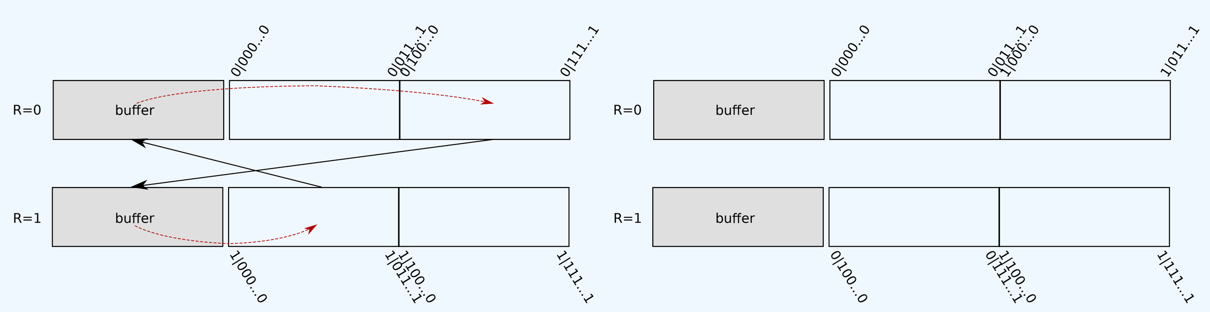
Permutation of qubits [1, 2]

- MPI parallelization is required because we need 2^{N+4} bytes to keep $a(n)$.
- Divide N bits to M global bits and $L = N - M$ local bits:

$$|n\rangle = |n_N \cdots n_{N-M+1}|\, n_L \cdots n_1\rangle\,. \tag{4}$$

- Global bits $\Leftrightarrow$ MPI rank $r = n_N \cdots n_{N-M+1}$.
- Swap data by using `MPI_Sendrecv` when operating a global qubit

$$a(0\,|\,1\,n_{L-1}\cdots) \leftrightarrow a(1\,|\,0\,n_{L-1}\cdots). \tag{5}$$



Note that $M = 1$ is assumed in this example.

References

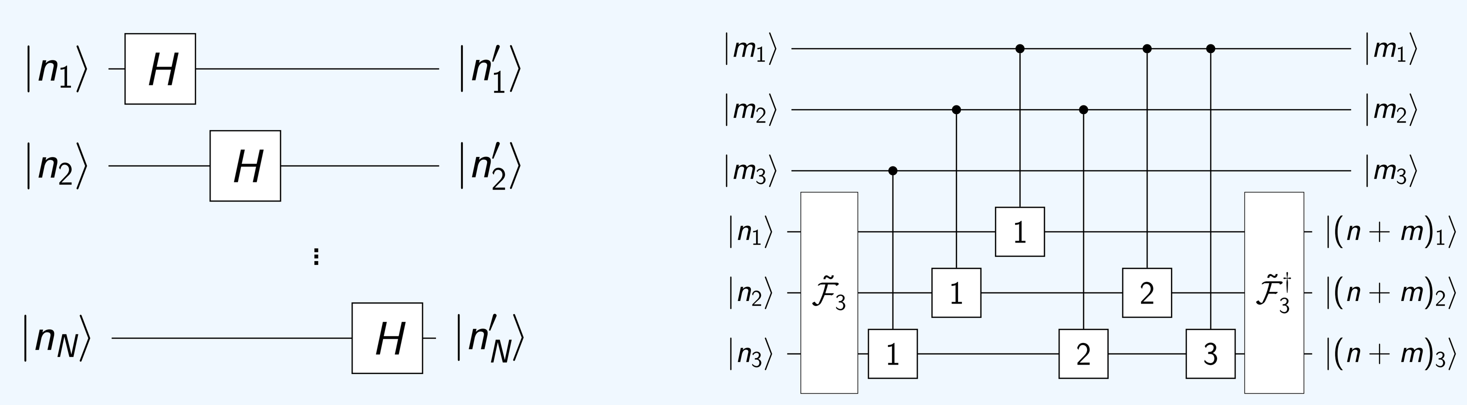
[1] K. De Raedt *et al.*, Computer Physics Communications **176**, 121 (2007).
[2] H. De Raedt *et al.*, Computer Physics Communications **237**, 47 (2019).
[3] T. Häner and Damian s. Steiger, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC17*, 33 (2017).
[4] https://github.com/naoki-yoshioka/braket
[5] T. G. Draper, arXiv:quant-ph/0008033.

Quantum circuits examined in this study

Left Hadamard gates.
Right Quantum adder [5] of two registers $n+m$. $\tilde{\mathcal{F}}_N$ is the quantum Fourier transform

$$\mathcal{F}_N\,|n\rangle = \frac{1}{2^{N/2}} \sum_{k=0}^{2^N-1} e^{2\pi i\, nk/2^N}\,|k\rangle = |\phi_N(n)\rangle \otimes \cdots \otimes |\phi_1(n)\rangle\,, \tag{6}$$

$$\tilde{\mathcal{F}}_N\,|n\rangle = |\phi_1(n)\rangle \otimes \cdots \otimes |\phi_N(n)\rangle\,, \tag{7}$$



Elapsed time of simulations

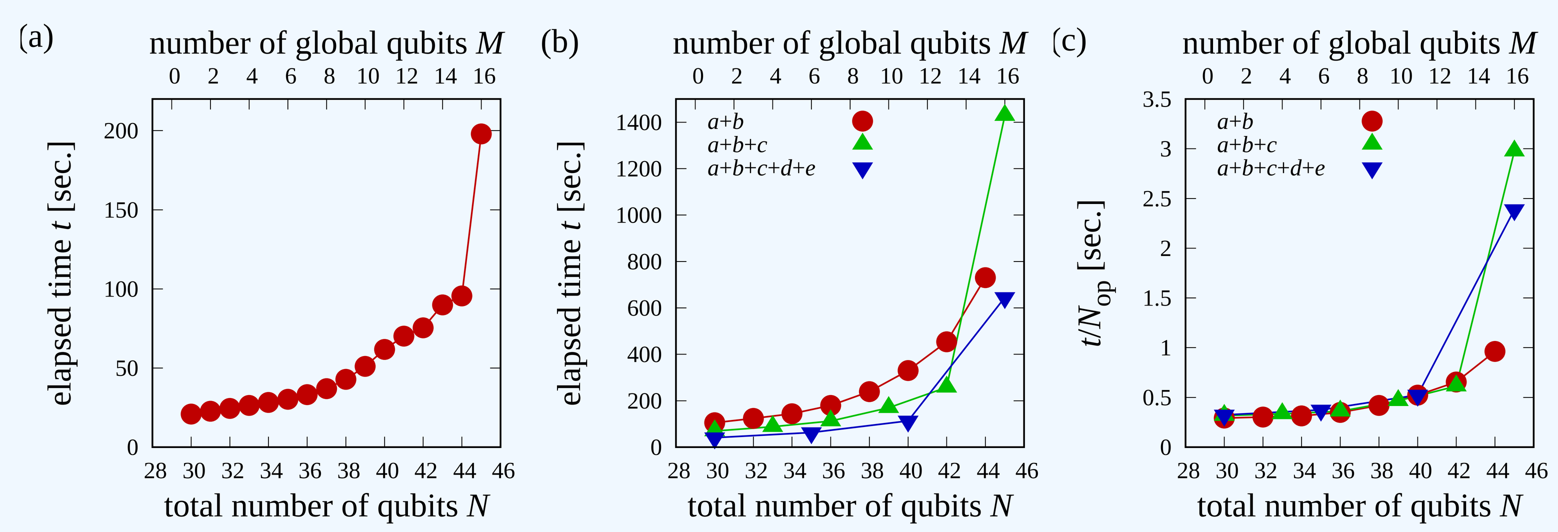
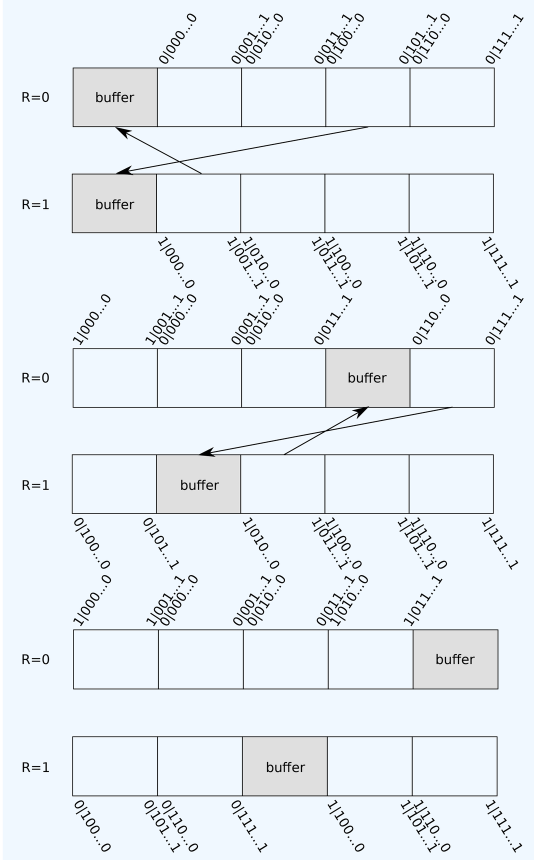


Figure: The elapsed time of simulation of quantum circuits; (a) the Hadamard gates and (b) the quantum adder (2, 3, and 5 registers). (c) The elapsed time of simulation of the quantum adder per gate operation as a function of the number of qubits.

- Elapsed time for $N = 45$ becomes large because of irregular patterns of MPI communications in K computer.

Page method



- Memory throughput: 64 GB/sec $\Rightarrow$ 0.06–0.09 sec. for each gate operation.
- Decrease intranode data transfer (see left figs.)

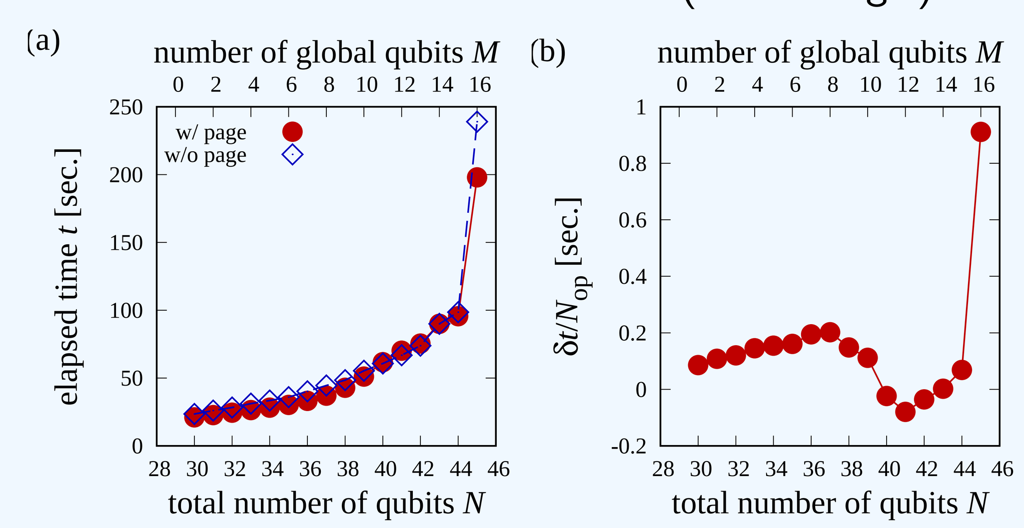
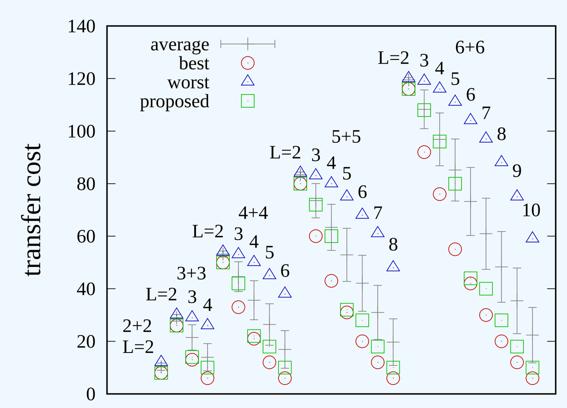


Figure: (a) The elapsed time of simulation of the Hadamard gates. The solid circles and the empty diamonds are corresponding to the results of simulations with page method, t_{w}, or without the method, t_{wo}, respectively. (b) $\delta t \equiv t_{\mathrm{wo}} - t_{\mathrm{w}}$ per gate operation.

Initial permutation of qubits

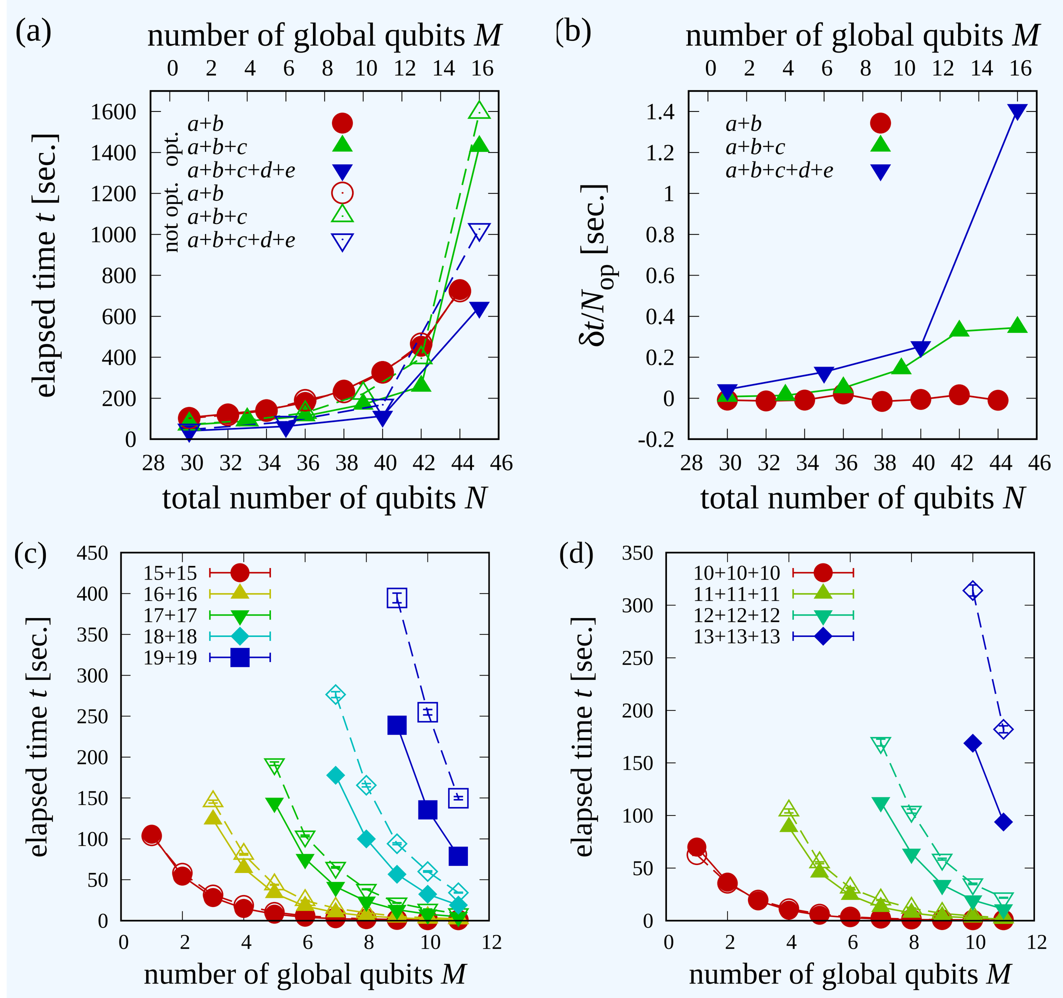
- Initial permutation of qubits is important to decrease data transfers.
  - Ex.) Quantum adder of two registers $n_3n_2n_1 + m_3m_2m_1$.
  - 6 data transfers for $|m_3m_2|n_3n_1n_2m_1\rangle$.
  - 26 data transfers for $|n_3n_2|n_1m_1m_3m_2\rangle$.
- Propose a simple method:
  - Sort by ascending order of the number of gates operating on each qubit.
  - Ex.) $|m_3m_2|m_1n_1n_2n_3\rangle$ for quantum adder $n_3n_2n_1 + m_3m_2m_1$.

Transfer cost



- Transfer cost: the number of pages to be transferred in each quantum gate operation.
  - 2 if one global qubit and one local qubit are swapped.
  - 4 if two global qubits and two local qubits are swapped.
- Brute-force calculations of transfer costs for quantum adder of two registers.
  - Transfer costs for the proposed method are much smaller than the averaged ones and close to the best ones in the case of the large number of local qubits L.
  - The proposed method works very well.

Initial permutation of qubits (cont.)



Elapsed time of simulations of the quantum adder.

(a) Weak scaling
- Solid lines: proposed method
- Empty dotted lines: results of initial permutation as a reverse of identity permutation such as $|m_1m_2|m_3n_1n_2n_3\rangle$.

(b) $\delta t \equiv t_{\text{no opt.}} - t_{\text{opt.}}$
- Improved if the number of registers is large.

(c) Strong scaling for two registers simulations
(d) and for three registers simulations
- Solid lines: proposed method
- Empty dotted lines: results of random initial permutation averaged over 100 samples.

Summary

- Develop a simulator of quantum computer [4] and test on the K computer.
- Simulations of the Hadamard gates and quantum adder up to 45 qubits.
- Page method.
  - Works well if the patterns of MPI intercommunications are irregular.
- Initial permutation of qubits.
  - Proposed simple method works very well.
- Future plans
  - Automatic search of the faster initial permutation by using genetic algorithm.
  - Reduction of data transfers for some quantumgates represented by diagonal matrices.
  - Support GPU/FPGA computing.