多体問題の計算科学

Computational Science for Many-Body Problems

#13 Parallelization for many-body problems

15:10-16:40 July 13, 2021

Lecture slide:

https://github.com/compsci-alliance/many-body-problems

1. Parallelization
2. Other numerical methods

Parallelization

(Revisited) Supercomputers in UTokyo

Ohtaka (CPU server)

東京大学 物性研究所
THE INSTITUTE FOR SOLID STATE PHYSICS
THE UNIVERSITY OF TOKYO

ISSP, UTokyo@Kashiwa
Theoretical Peak: 6,881TFlop/s
AMD EPYC 7720 2.0GHz, 64 cores x 2
Memory: 258GiB
1,680 nodes

Oakforest-PACS

JCAHPC@Kashiwa (Utokyo & Tsukuba U)

Theoretical Peak: 24,913.5TFlop/s
Intel Xeon Phi 7250 (1.4GHz, 68 cores, 112GB)
8,208 nodes

Oakbridge-CX (You can use this when you join CSA!)

Intel Xeon
1,368 nodes

(Revisited) “Fugaku” Supercomputer

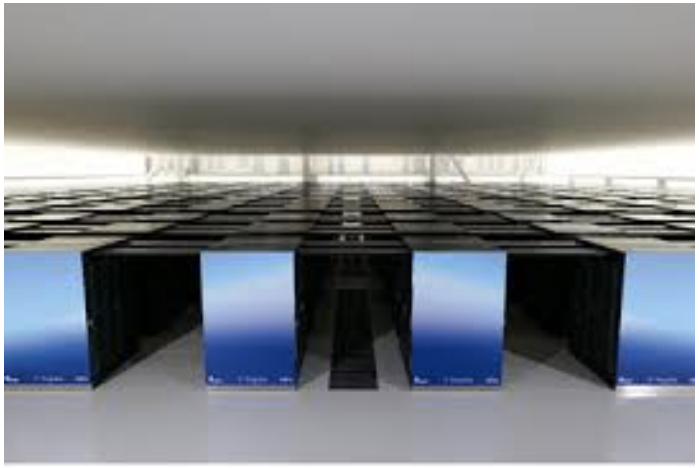
Fugaku 富岳 @RIKEN, Kobe

-158,976 nodes

A64FX

48 cores/node

32 GiB/node



- 1st in Top500 & 9th in Green500 on June 2020
- 1st in Top500 & 10th in Green500 on November 2020

(Revisited) Increasing FLOP/s

Clock rate saturated

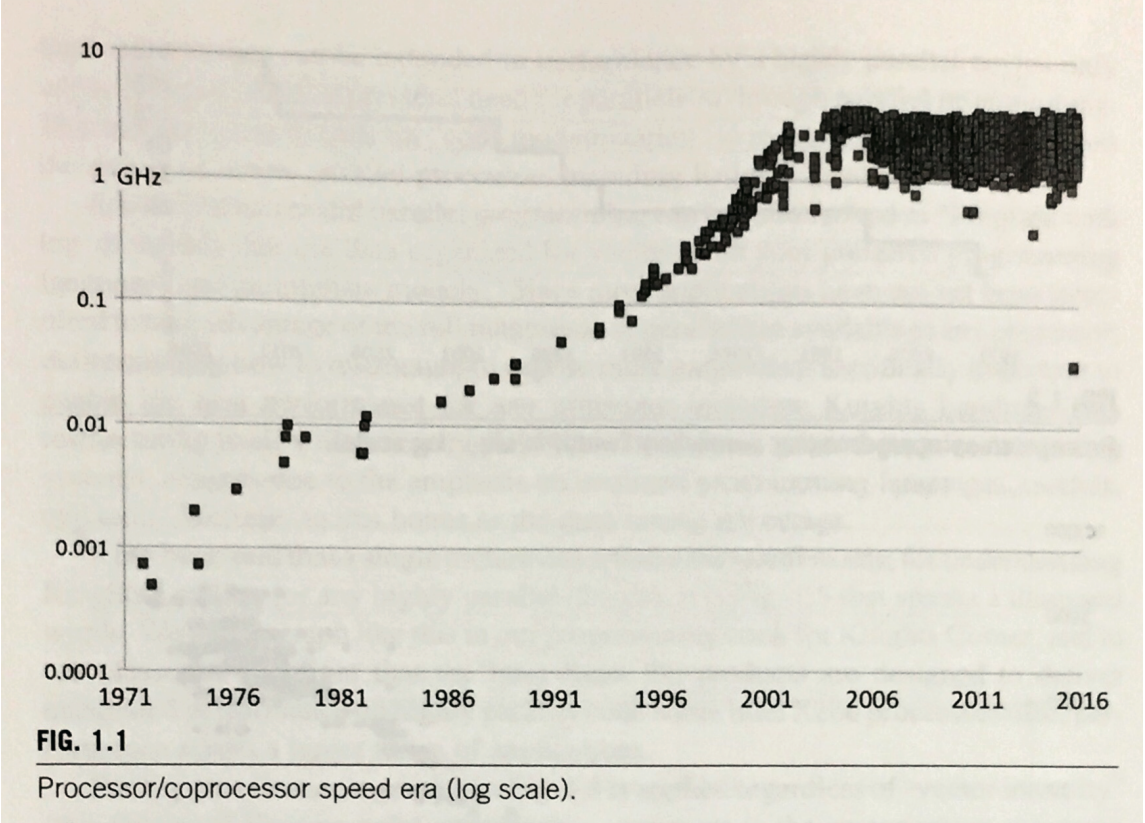


FIG. 1.1

Processor/coprocessor speed era (log scale).

J. Jeffers, J. Reinders, and A. Sodani,
Intel Xeon Phi Processor High Performance Programing

(Revisited) Increasing FLOP/s

Number of cores

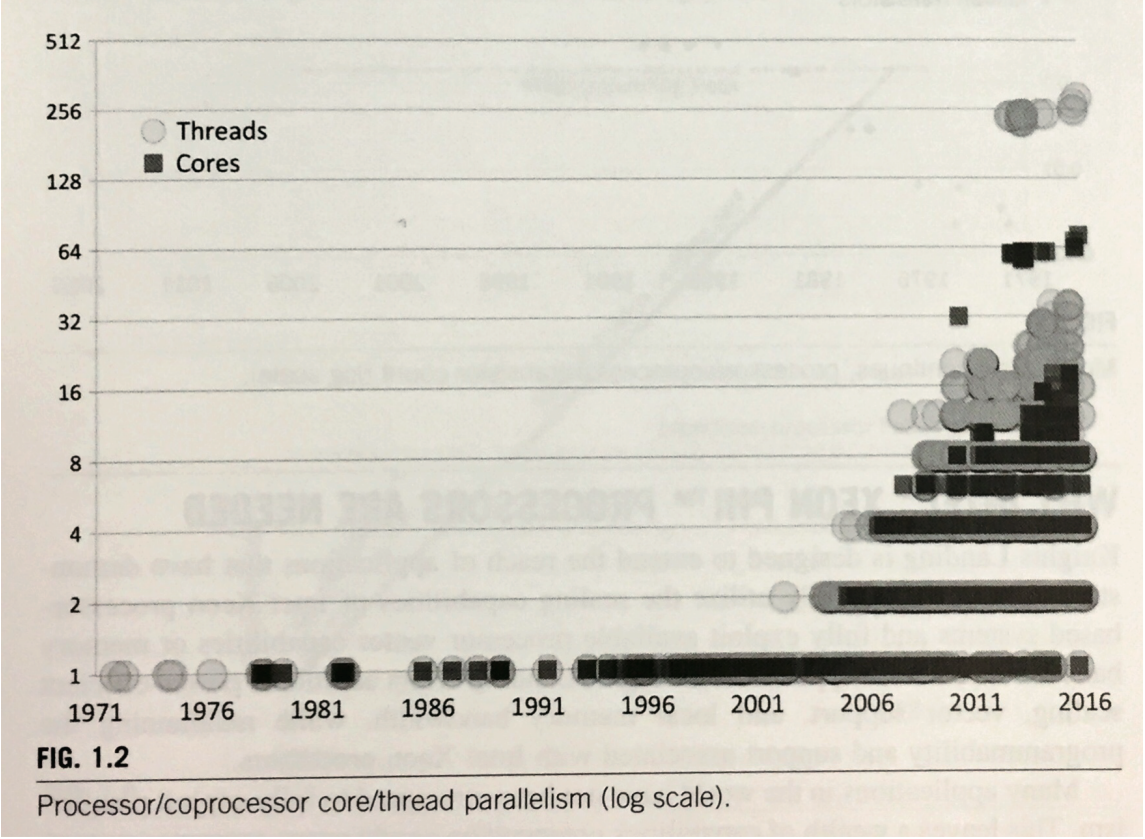


FIG. 1.2

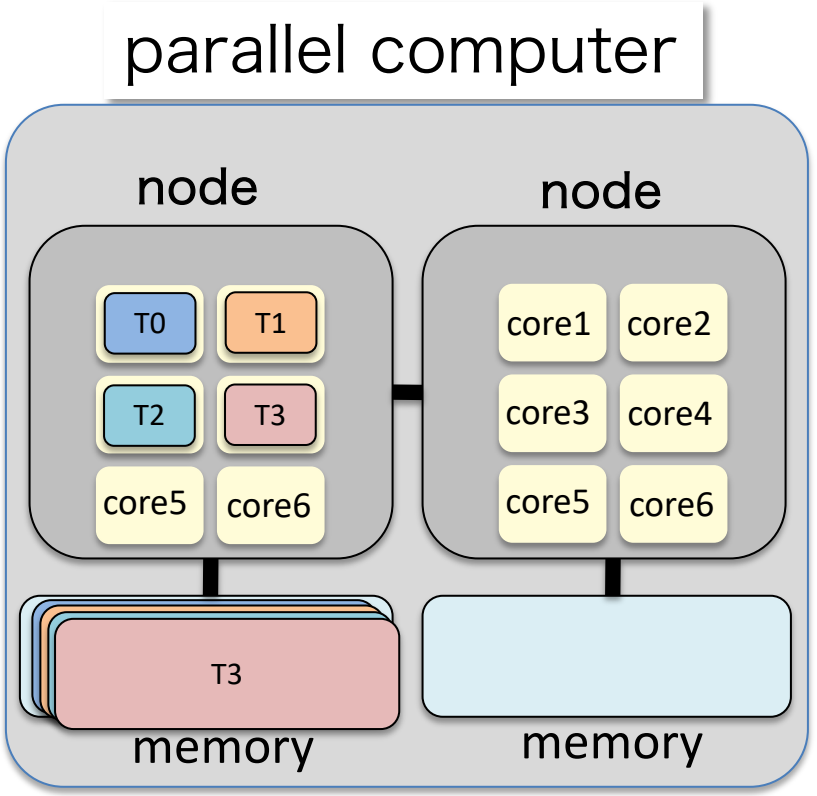
Processor/coprocessor core/thread parallelism (log scale).

J. Jeffers, J. Reinders, and A. Sodani,
Intel Xeon Phi Processor High Performance Programing

Parallelization

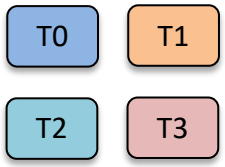
- Shared Memory Parallelization
  Data on the memory of the node
  is shared by the cores in the node

-Thread parallel
 Realized by, for example,
 OpenMP



Simple example in C lang: Divide a loop

```
#pragma omp parallel default(none) private(j) shared(a,b){
#pragma omp for
    for(j = 0; j < N; j++){
        C[j] = a*A[j] + b;
    }
}
```



Dangerous example

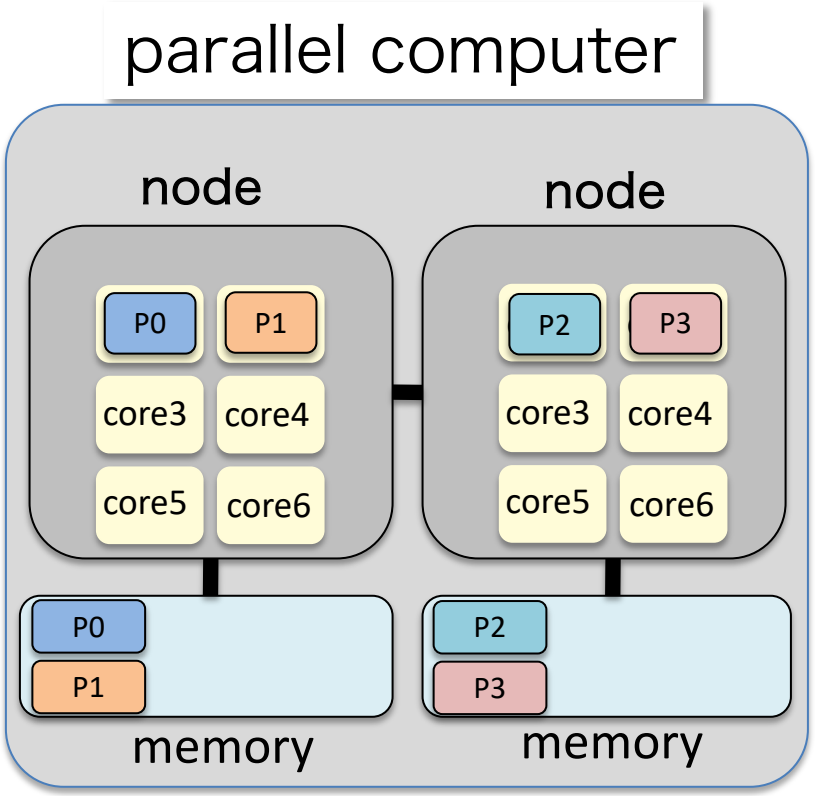
```
#pragma omp parallel default(none) private(j) shared(a,b){
#pragma omp for
    for(j = 0; j < N-1; j++){
        C[j+1] = a*C[j] + b;
    }
}
```

Dependency on former steps

Parallelization

- Distributed Memory Parallelization
  Each *process* can access its own data region in the node.

-Process parallel
Among processes,
data can be transferred
through MPI
(Message Passing Interface)



Declaration of MPI in C lang

```
ierr = MPI_Init(&argc, &argv);
ierr = MPI_Comm_size(MPI_COMM_WORLD, &nproc);
ierr = MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
```

nproc: # of processes
myrank: rank of the process, often used to generate process-dependent random seed for MCMC

A Problem in Massively Parallel MCMC

Burn-in steps in Markov chain Monte Carlo

- Markov chain Monte Carlo (MCMC), for example, Metropolis-Hastings algorithm generates samples following a give probability distribution with *an initial state*
  -A finite sequence of steps in the initial stage of the Markov chain is *often* discarded:
  Equilibration steps, burn-in or warm-up steps

cf.) Textbooks

A Problem in Massively Parallel MCMC

Extreme limit of massively parallel MCMC

- Few or few tens of warm-up steps and a single sample per process:
  1 step annealed importance sampling with trivial prior distribution and no weights

# of warm-up steps (> “auto correlation time”) is constant independent of # of processes

→Total # of MC samples / # of processes will be much smaller than # of warm-up steps

Non Markov chain MC

Yukito Iba, *Population Monte Carlo algorithms*
https://doi.org/10.1527/tjsai.16.279

Population MC

- Diffusion MC
- Population annealing:
  Annealed importance sampling+resampling

K. Hukushima and Y. Iba, AIP Conf. Proc. 690, 200 (2003)

Annealed importance sampling

$\{m_0^k, m_1^k, \ldots, m_n^k\}$ D. A. Hendrix and C. Jarzynski, J. Chem. Phys. 114, 5974 (2001).
R. M. Neal, Statistics and Computing, 11, 125 (2001)

0. Prepare replicas sampled from a prior distribution with β_0
1. Change β_j (j=0,1,⋯), calculate the weight, and anneal the replica and repeat 1.

$$W_j^k = W_{j-1}^k e^{-(\beta_j - \beta_{j-1}) f(m_{j-1}^k)}$$

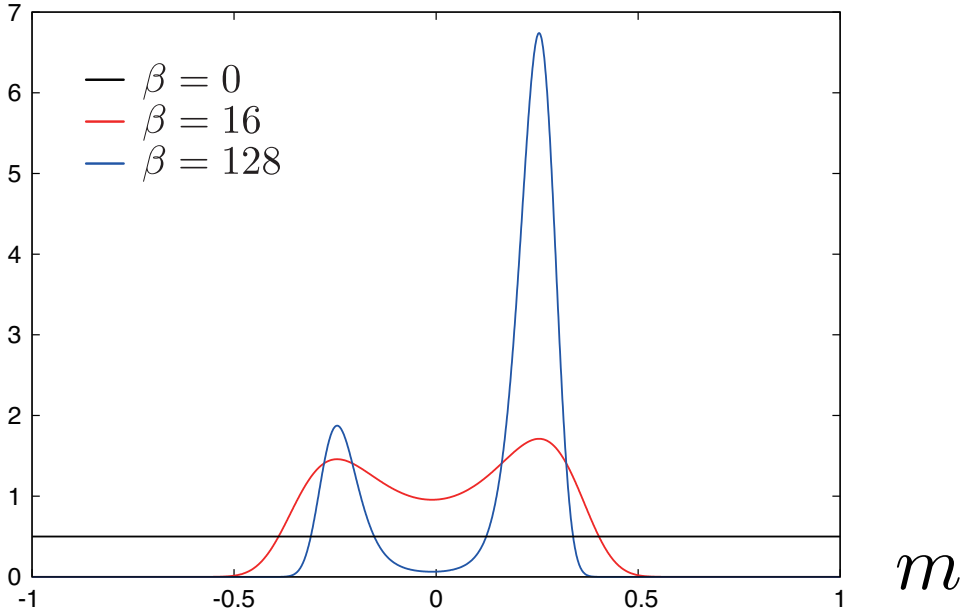
2. When $\beta_n = \beta_t$, take averages over the replicas with the weight

$$\beta_0 \leq \beta_1 \leq \cdots \leq \beta_n = \beta_t$$

$$p(m) \propto e^{-\beta f(m)}$$

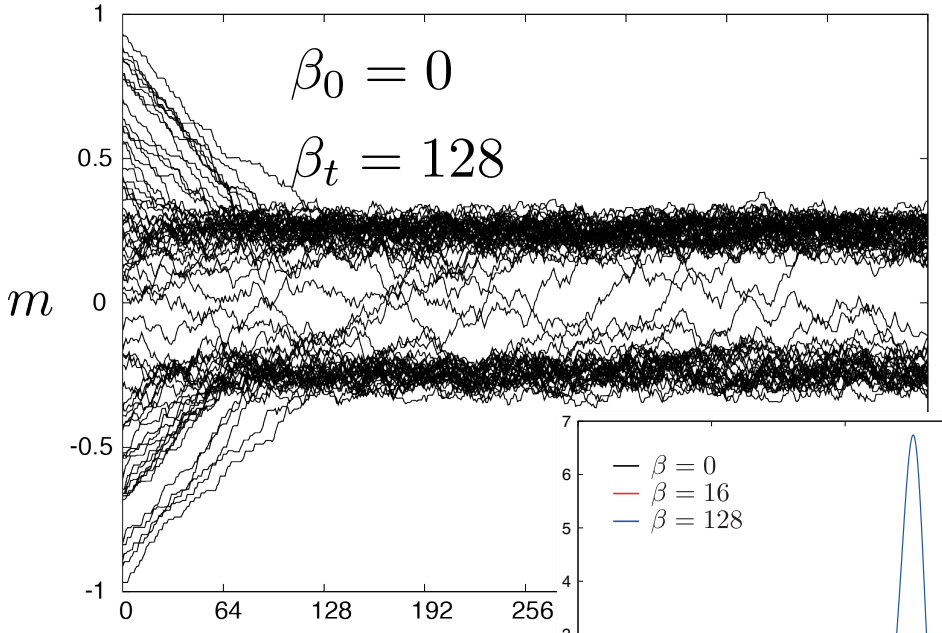
$$f(m) = -hm + rm^2 + vm^4$$

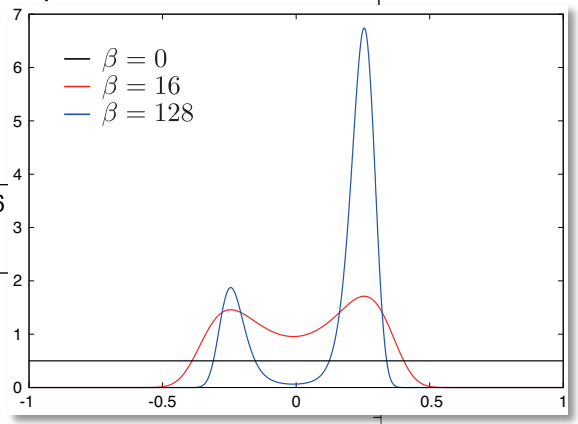
$$(h, r, v) = (0.02, -1, 8)$$

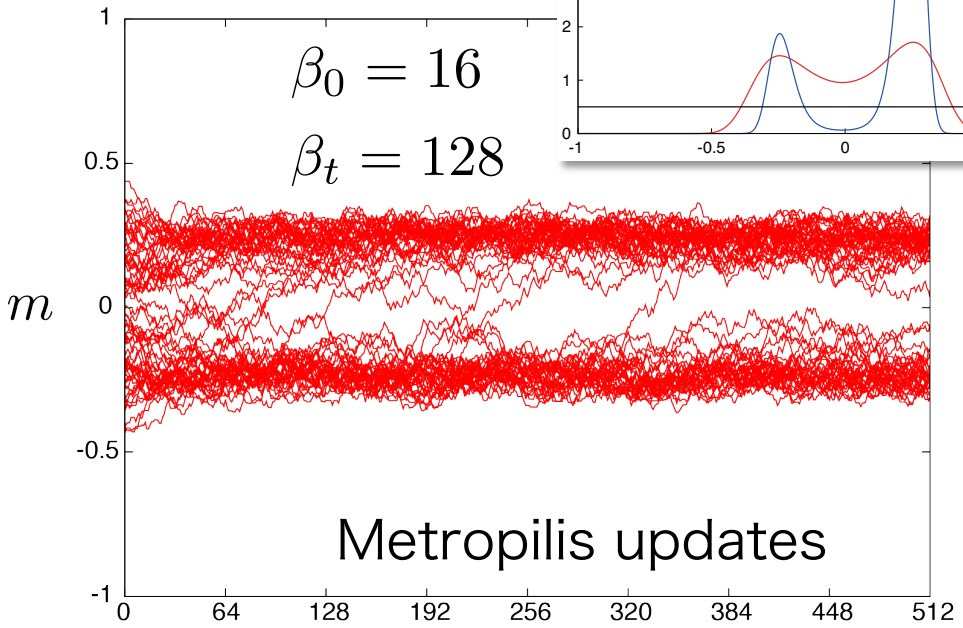


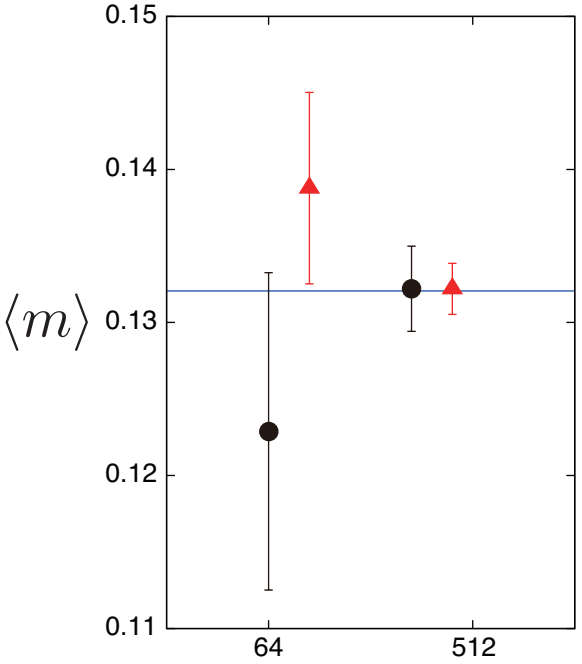
-Support condition $V_0 \supseteq V_1 \supseteq \cdots \supseteq V_n$ $V_j = \{x | f_j(x) \neq 0\}$

One step annealed importance sampling









$$n = 1$$

$$W_1^k = e^{-(\beta_t - \beta_0) f(m_0^k)}$$

$$\langle m \rangle = \frac{\sum_k m_n^k W_n^k}{\sum_k W_n^k}$$

Distributed Wave Functions

Multiplication of Hamiltonian to Wave Function

Example of multiplication

-4 spins

$$(\hat{S}_1^- \hat{S}_3^+ + \hat{S}_1^+ \hat{S}_3^-)|0100\rangle = |0001\rangle$$

$$(\hat{S}_1^- \hat{S}_3^+ + \hat{S}_1^+ \hat{S}_3^-)|0001\rangle = |0100\rangle$$

$$\hat{S}_\ell^+ = S_\ell^x + iS_\ell^y$$
$$\hat{S}_\ell^- = S_\ell^x - iS_\ell^y$$

$$|\phi\rangle = \sum_{I_j \in \{0,1\}} C_{I_0 I_1 I_2 I_3} |I_0 I_1 I_2 I_3\rangle$$

$$(\hat{S}_1^- \hat{S}_3^+ + \hat{S}_1^+ \hat{S}_3^-)|\phi\rangle$$

$$C_{I_0 1 I_2 0} \leftarrow C_{I_0 0 I_2 1}$$

$$C_{I_0 0 I_2 1} \leftarrow C_{I_0 1 I_2 0}$$

Multiplication of Hamiltonian to Wave Function

Example of multiplication

-4 spins

-Parallel: 2 processes
(processes are labeled by their rank)

$|I_0 I_1 I_2 I_3\rangle$

0	$\|0000\rangle$	$\to 0$
1	$\|1000\rangle$	$\to 1$
2	$\|0100\rangle$	$\to 8$
3	$\|1100\rangle$	$\to 9$
4	$\|0010\rangle$	$\to 4$
5	$\|1010\rangle$	$\to 5$
6	$\|0110\rangle$	$\to 12$
7	$\|1110\rangle$	$\to 13$
8	$\|0001\rangle$	$\to 2$
9	$\|1001\rangle$	$\to 3$
10	$\|0101\rangle$	$\to 10$
11	$\|1101\rangle$	$\to 11$
12	$\|0011\rangle$	$\to 6$
13	$\|1011\rangle$	$\to 7$
14	$\|0111\rangle$	$\to 14$
15	$\|1111\rangle$	$\to 15$

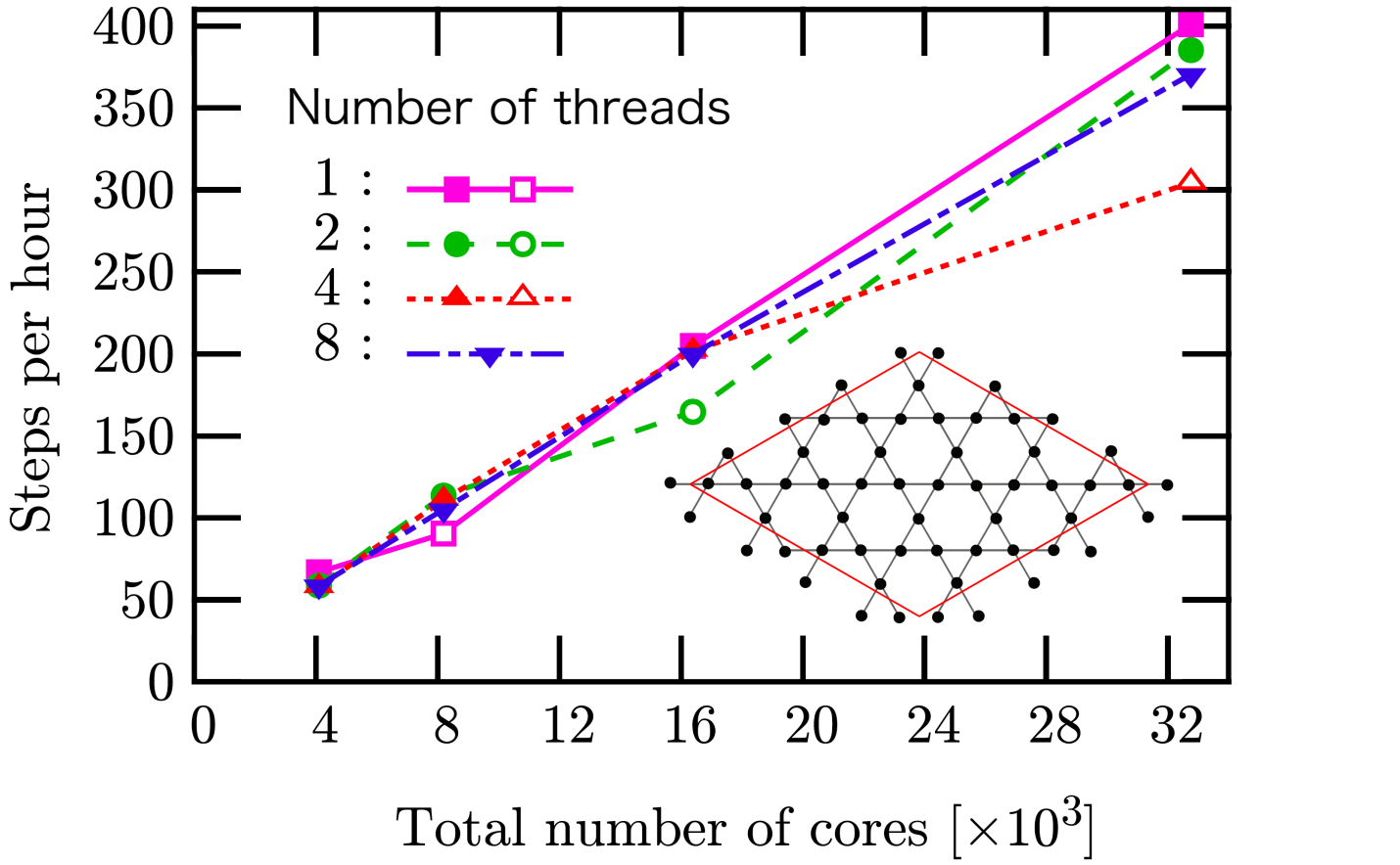
rank 0

rank 1

Speedup by Parallelization

36 spin Heisenberg model by HΦ

@K computer, AICS, RIKEN



Other Numerical Methods

Other Numerical Methods

Ab initio

-Hartree-Fock theory*
-Density functional theory
 Local density approximation*
 Generalized gradient approximation*
 Hybrid functional*
 GW*
-Post Hartree-Fock (Quantum Chemistry)
 Møller-Plesset*/Configuration interaction/Coupled cluster

*Difficulties in describing Mott insulators

-Transcorrelated

Other Numerical Methods

Many-body

-Exact diagonalization

-Quantum Monte Carlo (QMC) method

BSS, continuous time,…

Variational Monte Carlo

Green's function Monte Carlo

Diffusion Monte Carlo (also used in *ab initio* approaches)

-Numerical renormalization group (NRG) method

K. Wilson

-Density matrix renormalization group (DMRG) method

S. White

-Matrix product and tensornetwork method

→ lecture in A term

-Dynamical mean-field theory

Combination

(Revisited) 2nd Report Problems

Please choose one of three

Report 2

Problem 1: Monte Carlo for quantum systems

1-1 (compulsory).

-Evaluate statistical errors in a 2-point distribution function $g(r)$ of liquid helium 4 at each distance r.
You may use McMillan's variational Jastrow wave function and variational parameters given in vmc_helium4_2021.ipynb.
-Obtain the relationship between the statistical errors and numbers of Monte Carlo samples (and confirm the error is proportional to $1/N_{MC}^{1/2}$, where N_{MC} is the number of Monte Carlo samples).

1-2 (optional).

-Obtain 2-point distribution functions $g(r)$ of liquid helium 3 for likewise (not likewise) spin pairs at ambient pressure.
You may use variational Jastrow-Slater wave functions and variational parameters obtained in D. Ceperley, G. V. Chester, and M. H. Kalos, Phys. Rev. B 16, 3081 (1977).

Report 2

Problem 2: Krylov subspace method

2-1 (compulsory).

-Implement CG and solve a linear question.

Please select a symmetric postive-definite matrix A with non-zero offdiagonal element.

The vector b should not be the zero vector.

Please choose A and $\boldsymbol{b}$ with N_{H} (linear dimension of A) = 5.

-Illustrate convergence of the solution obtained at each CG step.

You may plot step dependence of 2-norm of residual vectors $\boldsymbol{r}_k$.

-Obtain the solution of $A\boldsymbol{x}=\boldsymbol{b}$ by Lapack and compare with the solution by CG.

2-2 (Optional). Implement LOBCG method.

-Confirm that the code can calculate eigenvalues and eigenvectors for a real symmetric matrix.

-Solve an eigenvalue problem with $N_{\mathrm{H}} > 5$

-Obtain the solution by Lapack and compare with the solution by CG.

Report 2

Problem 3: Open source software

3-1 (compulsory).

-Solve the following problems. You may use HΦ.
Estimate the energy difference between the lowest and 2nd lowest eigenstates of the 1dimensional S=1/2 and S=1 Hesenberg models with periodic boundary conditions.

-Use several L (number of spins) and extrapolate the gap to thermodynamic limit ($L \rightarrow \infty$). Please compare the extrapolated values with other results in the literature. You may use $a+b/L+c/L^2$、or $a+b\exp(-cL)$.

-Illustrate the extrapolations.

3-2 (optional). Reproduce the figures in Sec. 2.2 of the tutorials of HΦ: https://issp-center-dev.github.io/HPhi/manual/develop/tutorial/en/html/finite_temperature/Kitaev.html

-Please choose one out of the three problems

-The code should be included (a jupyter notebook is recommended).

-Deadline: 8/6

Report 1 & 2

Deadline for Report 1:
-2020/7/13
Deadline for Report 2:
-2020/8/6

Please submit your report through ITC-LMS
(web page for the reports will be opened).
If you have any trouble, please contcat us via email.

*If you are interested in QMC, open source software ALF & DSQSS would be worth trying.

Lecture Schedule

Classical

#1 Many-body problems in physics
#2 Why many-body problem is hard to solve
#3 Classical statistical model and numerical simulation
#4 Classical Monte Carlo method and its applications
#5 Molecular dynamics and its application
#6 Extended ensemble method for Monte Carlo methods

Quantum

#7 Quantum lattice models and numerical approaches
#8 Quantum Monte Carlo methods
#9 Applications of quantum Monte Carlo methods
-Path integral & applications
#10 Linear algebra of large and sparse matrices for quantum many-body problems
#11 Krylov subspace methods and their applications to quantum many-body problems
#12 Large sparse matrices and quantum statistical mechanics
#13 Parallelization for many-body problems