HΦ講習会ー全国共同利用大規模並列計算システムOCTOPUSを用いたハンズオン

並列化性能の紹介

2019/10/16 Wed.
Kazuyoshi Yoshimi (ISSP)

01 Project for advancement of software usability in materials science

02 Parallelization by vector division

03 Parallelization for full diagonalization

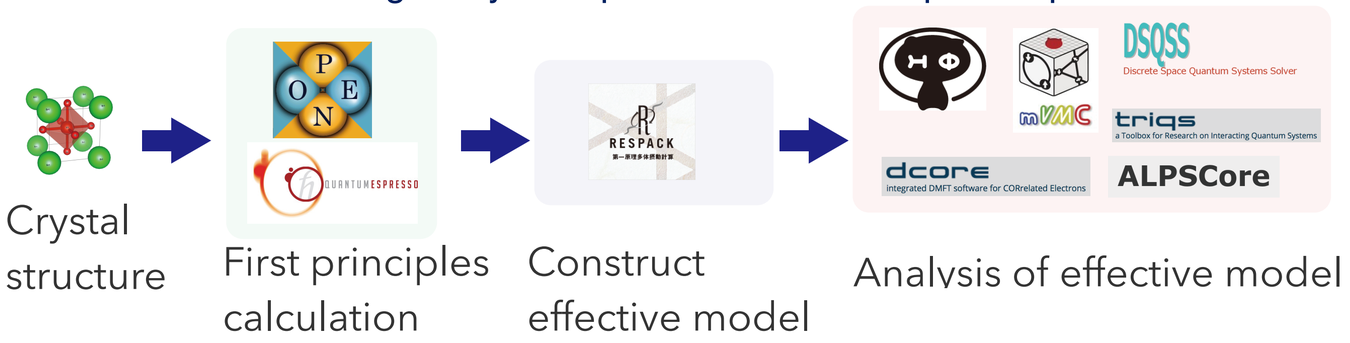
Project for advancement of software usability in materials science

To create an easy-to-use supercomputer environment
by improving the usability of key software in condensed matter physics.

Upgrade 2 software per year with the following team (Workload: 5 man months per software)

- ☆ Coordinator　ISSP staff (1 staff per project)
- ☆ Project manager (sometimes developer)　Kazuyoshi Yoshimi
- ☆ Main developer　Yuichi Motoyama

The following analysis is possible in ISSP supercomputer!



ISSP Super computer

Building communities through supercomputer/projects

ex. Cooperation with mathematical library Kω and HΦ

What is Kω？

Kω

Openness:3 ★★★ Document quality:1 ★☆☆

Kω implements large-scale parallel computing of the shifted Krylov subspace method. Using Kω, dynamical correlation functions can be efficiently calculated. This application includes a mini-application for calculating dynamical correlation functions of quantum lattice models such as the Hubbard model, the Kondo model, and the Heisenberg model in combination with the quantum lattice solver of quantum many-body problems, HΦ.

ref. https://ma.issp.u-tokyo.ac.jp/en/app/546

It is possible to directly examine the properties of matrix vectors by outputting them from HΦ.
→ Solve new problems in large matrices in cooperation with mathematical researchers.

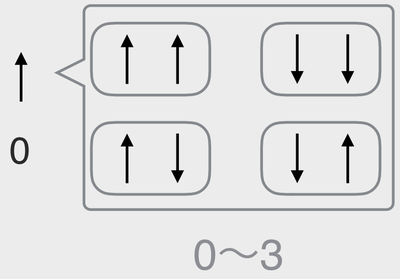
Parallelization (1)

Parallelization by vector division of Hilbert space (MPI)

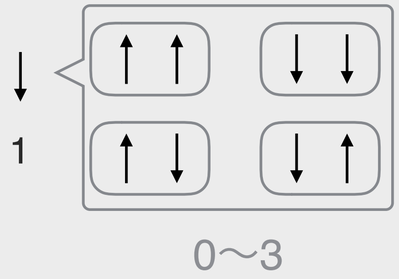
MPI parallelization: The number of processes applied to the subspace of Hilbert space
Thread parallelization: Hamiltonian vector product part using OpenMP

Ex.) MPI parallelization： Spin system (L=3、using 2 processes)

Rank 0: up spin　　Inner process



Rank 1: down spin　　Inner process



How to parallelize with HΦ:
Number of processes required for parallelization

↑, ↓

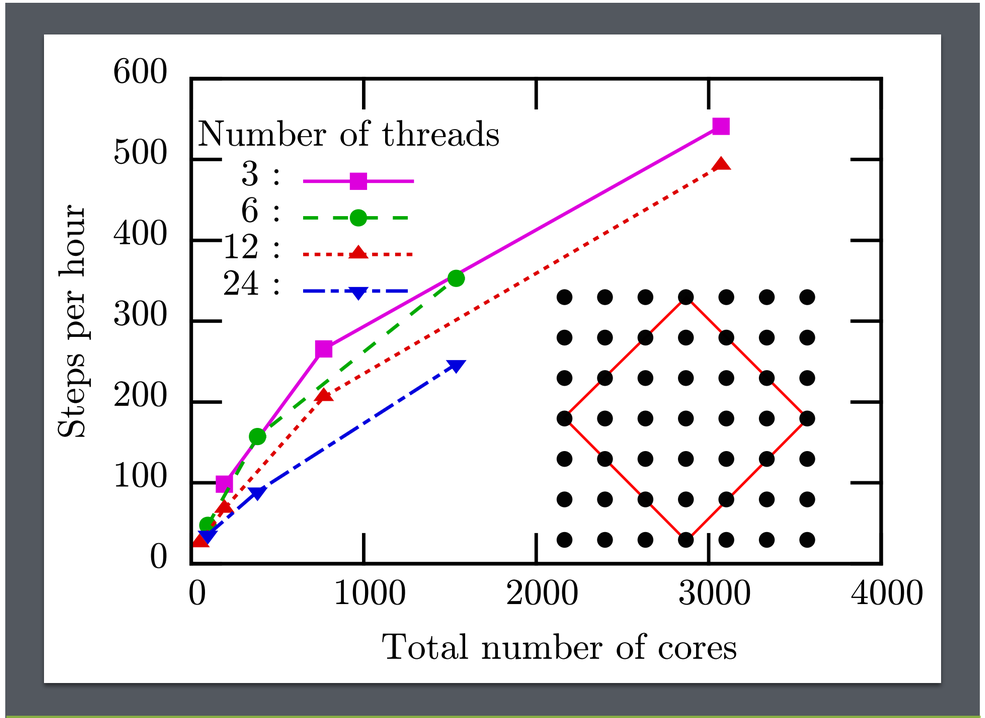
2 states per site
Process number : 2^N

0, ↑, ↓,↑↓

4 states per site
Process number : 4^N

0, ↑, ↓,↑↓

4 states per site (For spin,
0,↑↓ states are excluded)
Process number : 4^N



Total number of cores dependence of TPQ steps per hour

Benchmark result (1)

Parallelization by vector division of Hilbert space

18 site Hubbard model (half-filling, $2S_z=0$, $U/t = 8$)
ISSP super computer system B

- The number of steps increases with increasing the number of cores.
- Basically, flat MPI tends to be faster.

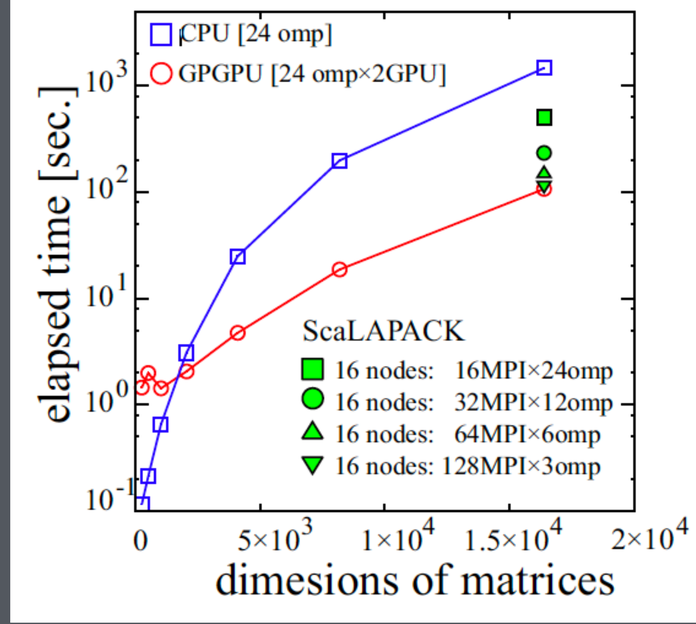
ref.) M. Kawamura, K. Yoshimi, T. Misawa et al, Computer Physics Communications 217, 180 (2017)

Parallelization (2)

Parallelization of full diagonalization

First, Hamiltonian matrix is calculated by single process.
Then, then matrix is diagonalized by the following library

1. LAPACK (single process)
2. ScaLAPACK (using MPI)
3. MAGMA (using GPU)



The dimensions of matrices and
the elapsed time for full diagonalization

Benchmark result (2)

～ Full diagonalization

using ScaLAPACK and MAGMA

- System: ISSP Super computer systemB
  ACC node
  - CPU: Intel Xeon 2.5 GHz (12cores) ×2
  - GPU: Nvidia Tesla K40 × 2
- ScaLAPACK: Faster with more processes.
- MAGMA works well if the matrix size is larger than 10^3.

ref.) “Implementation of GPGPU computing in full diagonalization for HΦ”, T. Misawa and K. Yoshimi,
Activity report 2017/Supercomputer Center, Institute for Solid State Physics, The University of Tokyo, 305-307 (2017).