

CCMSハンズオン：TeNeS講習会 ～物性研スパコンでの使用方法の解説

吉見 一慶

東京大学物性研究所 物質設計評価施設

ソフトウェア高度化推進チーム

1. 物性研スパコンの説明
2. システムB(ohtaka)でのTeNeSの利用方法
3. TeNeSの実習の前に

1-1. 物性研スパコンのスペック

物性研スパコンにはシステムBとシステムCの2つがあります。

- システムB (ohtaka) Dell PowerEdge C6525/R940 ← 実習ではこちらを利用

- CPUノード×1,680ノード (最大144ノードまで同時に使用可能)

CPU: AMD EPYC 7702, 2.0GHz (64core) ×2

主記憶容量: DDR4-3200, 16GB×16 = 256GB/node

- Fatノード×8ノード (最大2ノードまで同時に使用可能)

CPU: Intel Xeon Platinum 8280, 2.7GHz (28core) ×4

主記憶容量: DDR4-2933, 64GB×48 = 3TB/node

- システムC (kugui) HPE Apollo 2000 Gen10 Plus/HPE Apollo 6500 Gen10 Plus

- CPU: 128ノード (最大16ノードまで同時に使用可能)

CPU: AMD EPYC 7763, 2.45GHz, (64core)×2

主記憶容量: DDR4-3200, 16GB×16 = 256GB/node

- ACC: 8ノード

CPU: AMD EPYC 7763, 2.45GHz, (64core)×1

GPU: NVIDIA A100 40GB for HGX ×4

主記憶容量: DDR4-3200, 32GB×8 = 256GB/node

1-2. 物性研スパコンを使用するには？(1)

以下の手順で申請すれば利用可能です。

1. 研究代表者の登録
2. 研究課題を申請 (B, C, Eクラスは6月,12月の2回)
3. 利用審査
4. 報告書の提出

利用の流れの詳細は下記URLに記載してありますので、ご参照ください。

<http://www.issp.u-tokyo.ac.jp/supercom/visitor/overview>

1-2. 物性研スパコンを使用するには？(2)

小さい計算向けのクラス：Aクラス

Aクラスの概要

- 申請ポイント：100 ポイント以下
(基本的には1ノードを1日使用すると1ポイント消費。
fatノードの場合は4ポイント消費。)
- 申請回数：半期ごとに1回申請が可能。
ただし、A以外のクラスですでに利用している
研究代表者(グループ)の申請は不可。
- 報告書は必要なし。

その他申請クラスの詳細については https://mdcl.issp.u-tokyo.ac.jp/scc/guide/application_proposal/sckyouudou/about をご参照ください。

1-3. 利用可能なソフトウェア

- 物性研スパコンにプリインストールされているソフトウェア
 - ISSPスパコンページの「物性研スパコン」 - 「システムB」 or 「システムC」 - 「ソフトウェア」に記載
 - プリインストールソフトウェア一覧 (各ソフトウェアの詳細はMateriApps参照)
 1. 第一原理計算関連
 - abICS, OpenMX, VASP, QUANTUM ESPRESSO, RESPACK
 2. 量子格子模型ソルバー関連
 - ALPS, HΦ, Hwave, mVMC, DSQSS, DCore, ALPSCore/CT-HYB, TRIQS, TeNeS
 3. 分子動力学関連
 - LAMMPS
 4. その他
 - PHYSBO, 2DMAT, Julia, $K\omega$ (Shifted-Krylov)

赤字は東大物性研ソフトウェア開発・高度化プロジェクトに関連して導入されたソフトウェア

2-1. ohtakaでのソフトウェア実行 (1)

- ・ 事前準備

- ・ ohtakaへのログイン

メールで受け取ったアカウント・パスワードを使用します。
端末を開き以下のコマンドを打ってください(MA LIVE!でも可).

```
$ ssh -Y アカウント名@ohtaka.issp.u-tokyo.ac.jp
```

→ パスワードを入力

2-2. ohtakaでのソフトウェア実行 (2)

- TeNeSはohtakaにプリインストール済。
- 各種ファイルの置き場所 (覚書)
 - TeNeSのインストール場所
/home/issp/materiapps/oneapi_compiler_classic-2023.0.0--openmpi-4.1.5/tenes/
 - 実行ファイルのインストール場所
/home/issp/materiapps/oneapi_compiler_classic-2023.0.0--openmpi-4.1.5/tenes/tenes-2.0.0-0
 - サンプルファイルの場所
/home/issp/materiapps/oneapi_compiler_classic-2023.0.0--openmpi-4.1.5/tenes/tenes-2.0.0-0/sample

2-3. ohtakaでのソフトウェア実行 (3)

1. 計算環境の準備

```
$ source /home/issp/materiapps/oneapi_compiler_classic-2023.0.0--  
openmpi-4.1.5/tenes/tenesvars.sh
```

2. 入力ファイルの準備 (今回はAFH_squareを実行)

```
$ cp -rf $TENES_ROOT/sample/ ./tenes-sample  
$ cd ./tenes-sample/AFH_square
```

3. ジョブ投入スクリプトのコピー

```
$ cp /home/issp/materiapps/oneapi_compiler_classic-2023.0.0--openmpi-4.1.5/  
tenes/sample_jobscript/tenes.sh .
```

(*) 今回はお試し用のキュー(ccms8cpu)を使用します。

<<お試しキューの制限>>

1. ノード数は最大8ノードまで同時使用可能
2. CPU数は1ノードあたり128
3. 実行時間：最大一時間まで → 本日の19時まで使用可能です。

2-4. ohtakaでのソフトウェア実行 (4)

tenes.shの中身 (お試しキュー用に赤字部分は変更されています)

```
#!/bin/sh
#SBATCH -p i8cpu
#SBATCH -N 1
#SBATCH -n 4
#SBATCH -c 32
#SBATCH -t 00:10:00

set -e

source /home/issp/materiapps/oneapi_compiler_classic-2023.0.0—openmpi-4.1.5/
tenes/tenesvars.sh

module list

tenes_simple simple.toml
tenes_std std.toml
srun tenes input.toml
```

- ← キューの指定 (ccms8cpuに変更)
- ← ノードの個数の指定
- ← プロセス数の指定 (MPI)
- ← スレッド数の指定 (OpenMP)
- ← 最大計算時間 (時間 : 分 : 秒)

← 実行環境呼び出し

ジョブの投入方法

```
$ cd ./tenes-sample/AFH_square
$ sbatch tenes.sh
```

ジョブのステータス確認
\$squeue

2-4. ohtakaでのソフトウェア実行 (5)

ファイルをアップロード/取得する方法

- ・ アカウントにログイン
\$ sftp -Y アカウント名@ohtaka.issp.u-tokyo.ac.jp
→ パスワードを入力
- ・ 取得したいファイルがある場所まで移動
- ・ ファイルのアップロード (xxxxはファイル名)
\$ put xxxx
- ・ ファイルの取得 (xxxxはファイル名)
\$ get xxxx

<< 注意事項 >>

スパコンのお試しアカウントは講習会終了後(19時)に削除されます。
必要なファイルがある場合には、
講習会終了までにファイルを取得してください。

2-4. ohtakaでのソフトウェア実行 (6)

おまけ) sampleにあるpythonスクリプトの実行方法

- AFH_square/run.py,
- 04_magnetization/tutorial_magnetization.py
- 04_magnetization/tutorial_magnetization_square.py

1. run.pyを開き、以下のように変更する。

`MPI_cmd = ""` → `MPI_cmd = "srun"`

2. tenes.shを開き、以下のように変更する。

`tenes_simple simple.toml`

`tenes_std std.toml`

`srun tenes input.toml`

→ `python3 xxxx.py`

(xxxxは該当するpythonスクリプト名)

3. `sbatch tenes.sh`を実行