

CCMS Webハンズオン：2DMAT講習会 ～物性研スパコンを用いた使用方法の説明

吉見 一慶

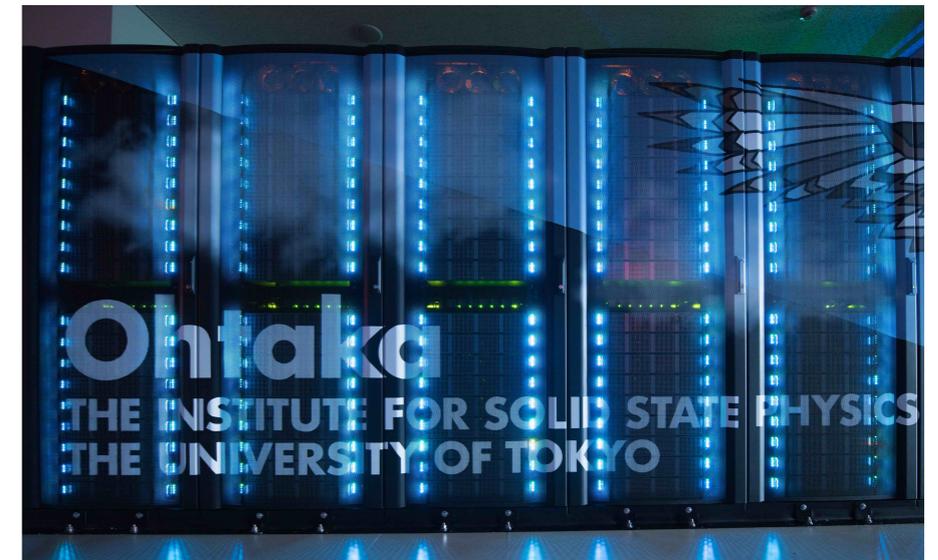
東京大学物性研究所 特任研究員

ソフトウェア高度化推進チーム

PASUMS

Project for advancement of
software usability in materials science

1. ohtakaの説明
2. ohtakaでの2DMATの利用方法



1-1. ohtakaの性能

- Fat ノード (2 ノードまで使用可能)

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

主記憶: DDR4-2933, 64GB×48 = 3TB/node

- CPU ノード(144 ノードまで使用可能)

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

主記憶: DDR4-3200, 16GB×16 = 256GB/node

1-2. ohtaka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. ohtaka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

■ 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 <http://www.issp.u-tokyo.ac.jp/supercom/visitor/about-class> をご参照ください。

1-2. ohtakaを使用するには？(3)

100ポイントでどの程度計算可能？

- CPUノードを1ノード1日利用：1ポイント消費

- Fat ノードを 1ノード 1 日利用：4ポイント消費

(ポイント消費のルールは ISSP スパコン Webページの「利用案内」 - 「ポイント消費制」に記載)

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

1-3. ohtakaで利用可能なソフトウェア

- ・ システムB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「利用案内」 - 「インストール済みアプリケーション」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

OpenMX, VASP, Quantum ESPRESSO, RESPACK

2. 量子格子模型ソルバー関連

ALPS, HΦ, mVMC, DSQSS, DCore, ALPSCore/CT-HYB, TRIQS, TeNeS

3. 分子動力学関連

LAMMPS

4. その他

Kw(Shifted-Krylov), 2DMAT, PHYSBO

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 (プロジェクトの詳細は 東大物性研スパコンページに記載！)

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)

- システムB ohtakaに2DMATはプリインストール済。
- 各種ファイルの置き場所 (覚書)
 - サンプルスクリプトと入力ファイルの場所
/home/issp/materiapps/intel/py2dmat
 - 順問題ソルバーの置き場所
 - sim_trhepd_rheed
/home/issp/materiapps/intel/sim_trhepd_rheed/
 - sxrd_calc
/home/issp/materiapps/intel/sxrd_calc/

2-3. ohtakaでのソフトウェア実行 (3)

1. 入力ファイルの準備

```
$ cp -rf /home/issp/materiapps/intel/py2dmat/py2dmat-2.1.0-1/sample .  
$ cd cd ./sample/sim-trhepd-rheed/mapper/
```

2. ジョブスクリプトのコピー (サンプルスクリプトを既に用意してあるのでそれを使用)

```
$ cp /home/issp/materiapps/intel/py2dmat/sample_jobscript/  
py2dmat_trhepd_rheed.sh .
```

3. 2DMATを実行

```
$sbatch py2dmat_trhepd_rheed.sh
```

4. 結果確認 (ColorMapができていることを確認。ref_ColorMap.txtが参照用の回答)

```
$cat ColorMap.txt  
6.000000 6.000000 0.047852  
6.000000 5.750000 0.055011  
6.000000 5.500000 0.053190  
6.000000 5.250000 0.038905  
6.000000 5.000000 0.047674  
6.000000 4.750000 0.065919  
6.000000 4.500000 0.053675
```

...

2-4. ohtakaでのソフトウェア実行 (4)

py2dmat_trhepd_rheed.shの中身

```
#!/bin/sh
#SBATCH -p i8cpu
#SBATCH -N 1
#SBATCH -n 128
#SBATCH -c 1
#SBATCH -t 00:10:00
set -e
source /home/issp/materiapps/intel/py2dmat/py2dmatvars.sh
module list
bulk.exe
srun py2dmat input.toml
```

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

← バルク計算の実行 (bulk.Pが出力される)

ohtaka: 128 cpu/ノード
128 * ノード数
= プロセス数 * スレッド数

2-4. ohtakaでのソフトウェア実行 (4)

py2dmat_trhepd_rheed.shの中身

```
#!/bin/sh
#SBATCH -p ccms8cpu      ← キューの指定
#SBATCH -N 1              ← ノードの個数の指定
#SBATCH -n 128           ← プロセス数の指定
#SBATCH -c 1             ← スレッド数の指定
#SBATCH -t 00:10:00      ← 最大計算時間 (時間 : 分 : 秒)
set -e
source /home/issp/materiapps/intel/py2dmat/py2dmatvars.sh
module list
bulk.exe                 ← バルク計算の実行 (bulk.Pが出力される)
srun py2dmat input.toml
```

ohtaka: 128 cpu/ノード
128 * ノード数
= プロセス数 * スレッド数

講習会では専用キュー : ccms8cpu のみ使用可能。

投げたジョブの状況の確認は

`$squeue`

でできます。

(注) ccms8cpuキューはこの講習会中のみ使用可能です。

2-4. ohtakaでのソフトウェア実行 (5)

sxrdを利用するには？ (sample/sxrdにサンプルあり)

```
$ cp /home/issp/materiapps/intel/py2dmat/sample_jobscript/py2dmat.sh .  
py2dmat.shの中身
```

```
#!/bin/sh  
#SBATCH -p i8cpu  
#SBATCH -N 1  
#SBATCH -n 128  
#SBATCH -c 1  
#SBATCH -t 00:10:00  
set -e  
source /home/issp/materiapps/intel/py2dmat/py2dmatvars.sh  
module list  
srun py2dmat input.toml
```

← この部分をccms8cpuに書き換えてください。

(*)前処理にあったbulk.exeが消えている。

ジョブの実行方法

```
$sbatch py2dmat.sh
```

→ ジョブが終了するとColorMap.txtが生成されます。

どんな場合にスパコンを使うと便利？

アルゴリズムごとのMPI実装に関する簡単な紹介

(ohtakaでは1ノード128cpuなので128プロセスでのMPI並列可能)

1. **mapperモードで細かく一気に計算したい場合**

プロセス数を多くして刻みを細かくする。



2. **exchangeで温度分割点を増やして細かく計算したい場合**

プロセス数ごとに温度分割が行われるので、プロセス数を増やす。

3. **bayseで候補点数がたくさんある場合**

全候補点での獲得関数の評価をMPI並列しているので高速化可能。

2-5. (補) ファイルの送受信方法

scp, sftpなどが利用可能です。以下、scpを利用する場合の例を記載します。

1. 自分のPCからohtakaにコピー

```
$scp path_to_file user_name@ohtaka.issp.u-tokyo.ac.jp:path_to_copy
```

- `path_to_file`: コピーしたいファイルのパス
- `user_name`: ohtakaのユーザ(アカウント)名
- `path_to_copy`: コピーするファイルの置き場所

パスワードが聞かれるので、スパコンのログイン用パスワードを入れる。

2. ohtakaから自分のPCにコピー (ひっくり返す)

```
$scp user_name@ohtaka.issp.u-tokyo.ac.jp:path_to_file path_to_copy
```

なお、ディレクトリごとコピーしたい場合には、"-r"オプションをつければOKです。

```
$scp -r user_name@ohtaka.issp.u-tokyo.ac.jp:path_to_file path_to_copy
```

2-5. (補) ohtakaでの利用回数測定

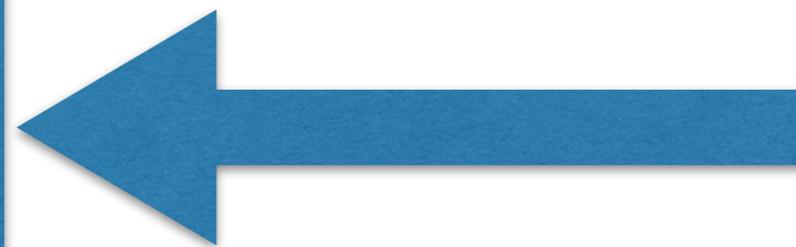
対象ソフトウェア：ソフトウェア高度化対象プログラム

プリインストールソフト

(計測用*)

システムB

- ・XXXXXXXXXX
- ・並列数
- ・高度化ソフトA



ユーザー

- ・ユーザーID
- ・並列数
- ・高度化ソフトA

ユーザーIDを暗号化

個人情報は見えない

(*) 利用率を計測しないソフトの選択