

CCMS Webハンズオン：Moller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

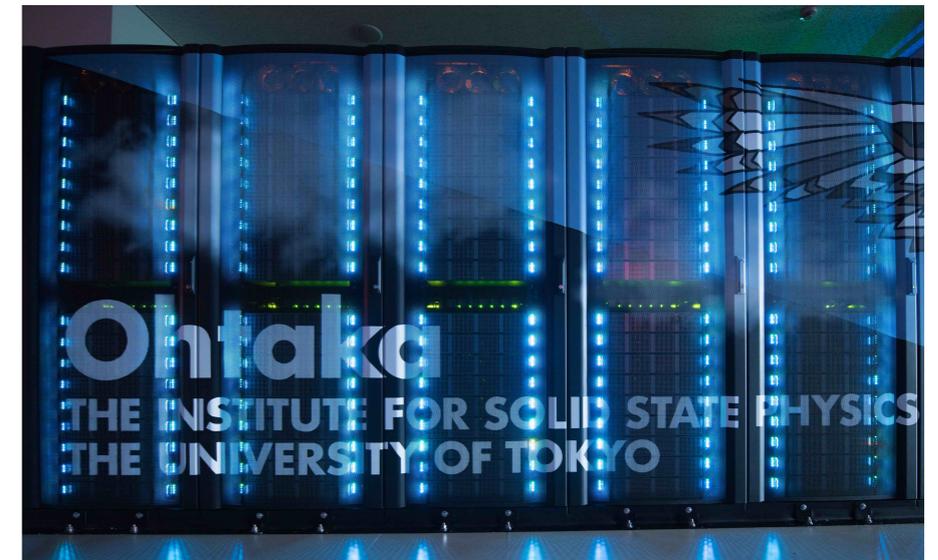
東京大学物性研究所 特任研究員

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Moller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, H-wave, mVMC, DSQSS, DCore, ALPSCore/CT-HYB, TRIQS, TeNeS
 3. 分子動力学関連
LAMMPS
 4. その他
K ω (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

事前に送付されたアカウント情報・パスワードをもとに、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moller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/oneapi_compiler_classic-2023.0.0--openmpi-4.1.5/moller
 - チュートリアルファイルの場所
/home/issp/materiapps/oneapi_compiler_classic-2023.0.0—openmpi-4.1.5/moller/tutorial
 - $H\Phi$ で $S=1/2$ Heisenberg鎖のサイズ依存性を求めるチュートリアル
 - single : Lを自分で変更して実行
 - sequential : L_xxx (xxxはサイズ)の入力ファイルをあらかじめ用意して逐次実行
 - moller : L_xxx (xxxはサイズ)の入力ファイルをあらかじめ用意して一斉実行
- の3つが用意されています。

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/oneapi_compiler_classic-2023.0.0--  
  openmpi-4.1.5/moller/tutorial .  
$ cd ./tutorial/moller
```

2. mollerを実行

```
$ source /home/issp/materiapps/oneapi_compiler_classic-2023.0.0--  
openmpi-4.1.5/moller/mollervars.sh  
$ moller input.yaml > job.sh
```

3. 計算実行 (30秒程度で終了)

```
$ sbatch job.sh list.dat
```

4. 結果確認

```
$ moller_status input.yaml
```

```
| job      | hphi      |  
|-----|-----|  
| L_10    | o         |  
| L_12    | o         |  
| L_14    | o         |  
| L_16    | o         |
```

...

2-4. ohtakaでのソフトウェア実行 (4)

input.yamlの中身 (一部)

```
name: HPhi
description: AFH chain

platform:
  system: ohtaka
  queue: i8cpu          ← キューの指定
  node: 8              ← ノードの個数の指定
  elapsed: 00:30:00    ← 最大計算時間 (時間 : 分 : 秒)

prologue:
  code: |
    echo "start"
    date
    module purge
    module load oneapi_compiler/2023.0.0 openmpi/4.1.5-oneapi-2023.0.0-classic

    ulimit -s unlimited

  ...
```

講習会では専用キュー : ccms8cpu を使用。

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

- XXXXXXXXXXX
- 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

3. 演習時間～サンプルの紹介 (1)

- ISSPデータリポジトリにサンプル例あり

ポータルサイトで
データリポジトリの検索 🔍



- ✓ 論文データや研究・計算データをアップロード
- ✓ ダウンロードしたデータの利活用
- ✓ GitLabによりGUIとCUIの両方に対応
- ex.) 事前にアップロードしてarXiv・投稿論文にURLを記載



☆フロントエンドサーバ

- CPU 3 cores
- Memory 4.5 GB
- Storage 100 GB

☆GitLabサーバ

- CPU 6 cores
- Memory 9 GB

☆データ領域 4 TB
(将来的に拡張予定)

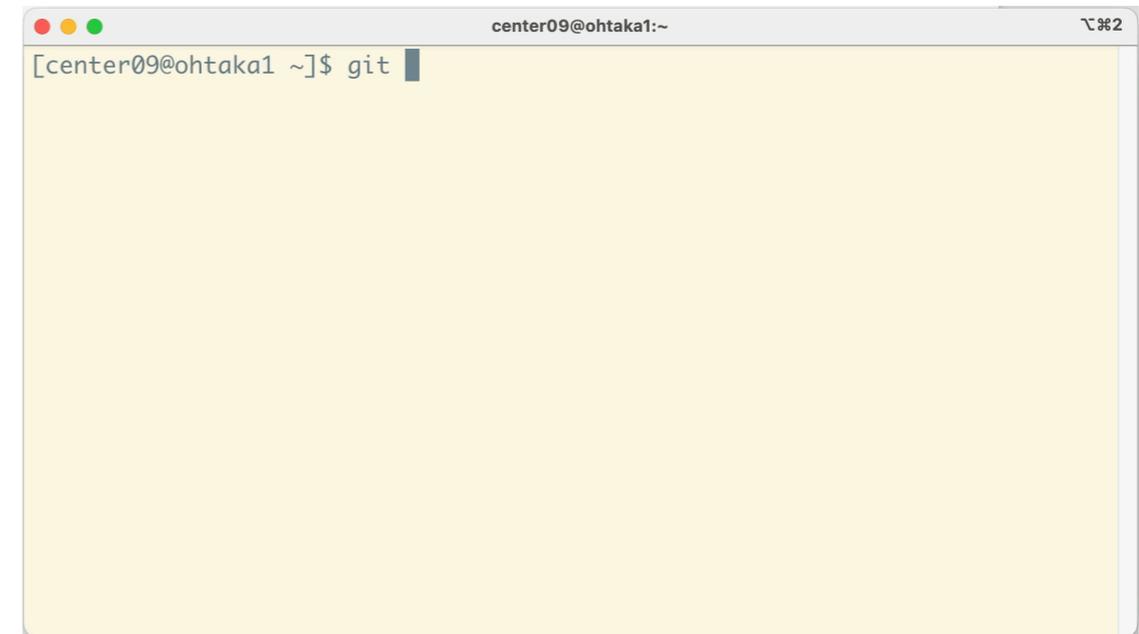
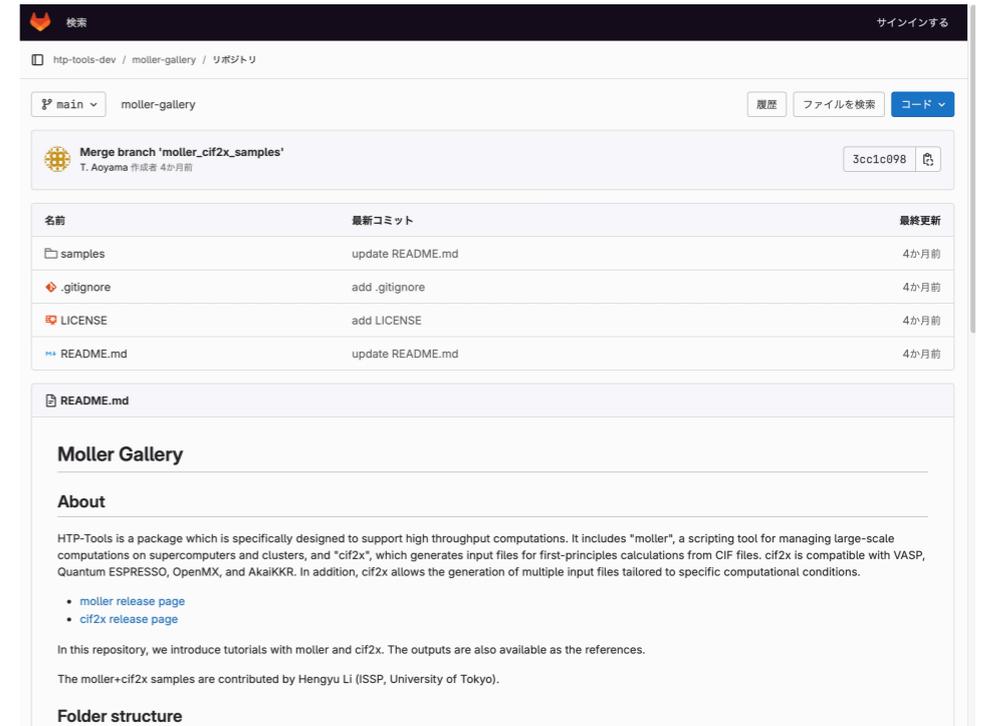
ref.) https://mdcl.issp.u-tokyo.ac.jp/scc/guide/application_proposal/issp-datarepo

- ポータルサイト : <https://datarepo.mdcl.issp.u-tokyo.ac.jp>
- データリポジトリ(GitLab) : <https://isspns-gitlab.issp.u-tokyo.ac.jp/explore>

「ISSP data repository」 -> ポータルサイト -> 「Moller Gallery」 -> 「URL」 をクリック

4. 演習時間～サンプルの紹介 (2)

- データー式のダウンロード方法
 - リンクをコピー：「コード」 → 「HTTPSでクローン」を選択する。
- ohtaka上でクローン
\$ git clone https://isspns-gitlab.issp.u-tokyo.ac.jp/http-tools-dev/moller-gallery.git



最新バージョン(2024/10/18現在)は以下のディレクトリにプリインストール済

/home/issp/materiapps/oneapi_compiler_classic-2023.0.0—openmpi-4.1.5/moller/gallery

3. 演習時間～サンプルの紹介 (3)

- フォルダ構成 (各ディレクトリにREADMEあり)

- LICENSE
- README.md
- samples
 - moller ← **はじめにトライ!**
 - simple ← bashを実行する例
 - dsqss ← dsqss (量子モンテカルロのプログラム)を実行する例
 - AFH-chain
 - hphi ← hphi (厳密対角化のプログラム)を実行する例
 - AFH-chain
 - 2S_1
 - 2S_2
 - qe ← Quantum Espresso (第一原理計算のプログラム)を実行する例
 - moller+cif2x ← **mollerが終わった後にトライ!**
 - Basic_usage
 - cutoff_energy
 - kpoints
 - DFT+U