ハイスループット計算ツール moller の紹介

2024年度 第4回物性アプリオープンフォーラム
2024年10月7日 @物性研A612+Zoom

東京大学物性研究所 附属物質設計評価施設
青山 龍美

Outline

- 物性研スパコンとソフトウェア開発高度化プロジェクト
  - スパコンシステムについて
  - ソフトウェア開発高度化について
- ハイスループット計算ツール HTP-tools (cif2x + moller)
  - moller とは？
  - moller を使う
  - moller をインストールする
- まとめと展望

物性研スパコン

- 物性研ではスーパーコンピュータの計算資源提供による計算物性科学コミュニティ支援を行なっている
  - 1995年よりスパコン全国共同利用を実施
    国内の大学・研究機関に所属する研究者の物性研究を目的とした申請課題を受け付け、審査により資源配分する
  - 物質科学スーパーコンピュータ共用事業(SCCMS)枠でのハードウェア共用を実施
  - 最近は各年度 約330課題・1000ユーザが利用
- 約5年ごとにハードウェアを更新
  - ベクトル計算機、スカラー型並列計算機、アクセラレータ搭載型などのバリエーション
  - 計算機アーキテクチャの複雑化により、オリジナルコードの開発・高度化のコストが上昇

「スーパーコンピュータによる計算物性研究の革新と未来」(物性研)より

物性研スパコン

現行システム

	システムB (ohtaka)	システムC (kugui)
	2020年10月～	2022年6月～
機種名	Dell PowerEdge C6525/R940	HPE Apollo 2000/6500 Gen10 plus
CPUノード	AMD EPYC 7702 (64core) x2 256GB memory 1680 nodes	AMD EPYC 7763 (64core) x2 256GB memory 128 nodes
FATノード/ ACCノード	Xeon Platinum 8280 (28core) x4 3TB memory 8 nodes	AMD EPYC 7763 (64core) x1 256GB memory NVIDIA A100 x4 8 nodes
ネットワーク	InfiniBand HDR100 Fat-tree	InfiniBand HDR200 Fat-tree
総演算性能	**6.881 PFlops**	**0.973 PFlops**

PASUMS

物性研ソフトウェア開発高度化プロジェクト

- **P**roject for **A**dvancement of **S**oftware **U**sability in **M**aterials **S**cience
  - 並列計算の高度化・複雑化に対応するためのプログラム開発支援
  - 物性研究分野で特に重要であり、物性研スパコンでの利用が見込まれるプログラムを対象
  - 機能追加や拡張、ユーザーインターフェース整備、マニュアル作成などを実施。成果物はオープンソース・ソフトウェアとして広く公開する
- 物性研の公募により募集、共同利用委員会による審査により選定する。
  年度ごとに 2〜3件程度を採択
- 開発体制：
  提案者・協力者 + 物性研開発スタッフ + コーディネータ(所員)

2025年度課題を募集予定
11月8日に公募説明会を開催 (→物性研スパコンウェブサイト)

PASUMS

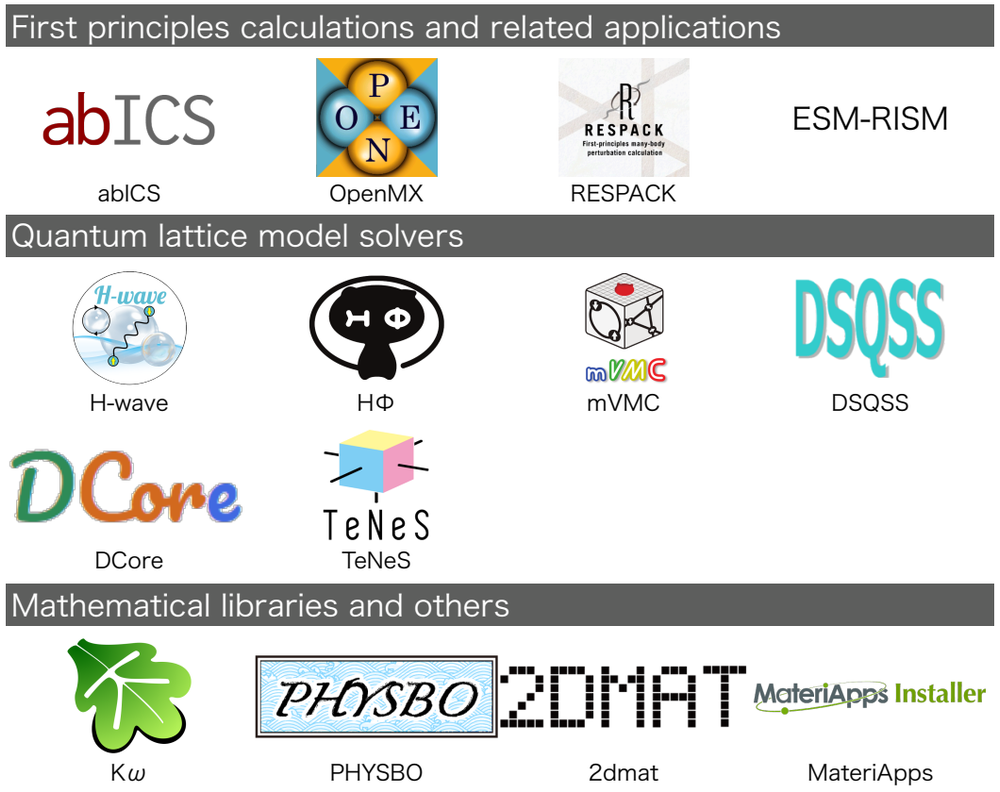
物性研ソフトウェア開発高度化プロジェクト

- 課題終了後は:
  - 成果物の公開 (オープンソース・ライセンス)
    - ソースコード・マニュアル
      → GitHub などで配布
    - サンプルコード・データ、チュートリアル資料など
      → 物性研データリポジトリで公開
  - 論文出版
  - 利用講習会などの普及活動
    - 2023年度はハンズオン講習会を5回開催
    - 2024年度 5月・8月に開催
      次回は10月18日に moller 講習会を開催予定

PASUMS

物性研ソフトウェア開発高度化プロジェクト

- 2015年度にスタートして以来、現在までに15件のアプリケーションが開発されている
  - 第一原理計算関連
  - 量子格子模型ソルバー
  - 数理ライブラリ・他
- 物性研スパコンにプリインストール
  - スパコン利用者はすぐに使える
  - 高度化アプリ以外にも、主要なアプリケーションパッケージが利用可
    - Quantum Espresso, VASP, LAMMPS, ALPS, Pomerol, etc.
  - プロジェクト評価のため利用率を測定



PASUMS

物性研ソフトウェア開発高度化プロジェクト

- 2015年度にスタートして以来、現在までに15件のアプリケーションが開発されている
  - 第一原理計算関連
  - 量子格子模型ソルバ
  - 数理ライブラリ・他
- 物性研スパコンにプ
  - スパコン利用者は
  - 高度化アプリ以外
    - Quantum Espress
  - プロジェクト評価のため利用率を測定

First principles calculations and related applications

abICS

RESPACK

ESM-RISM

mVMC

DSQSS

s

2dmat

MateriApps

HPCIソフトウェア賞 2023年度

開発部門賞 最優秀賞
「mVMC」

開発部門賞 優秀賞
「HΦ」

普及部門賞 最優秀賞
「MateriAppsプロジェクト」
「PASUMSプロジェクト」

ジが利用可

HTP-tools (cif2x + moller)

概要

機械学習を活用した物性予測や物質設計(マテリアルズ・インフォマティクス)

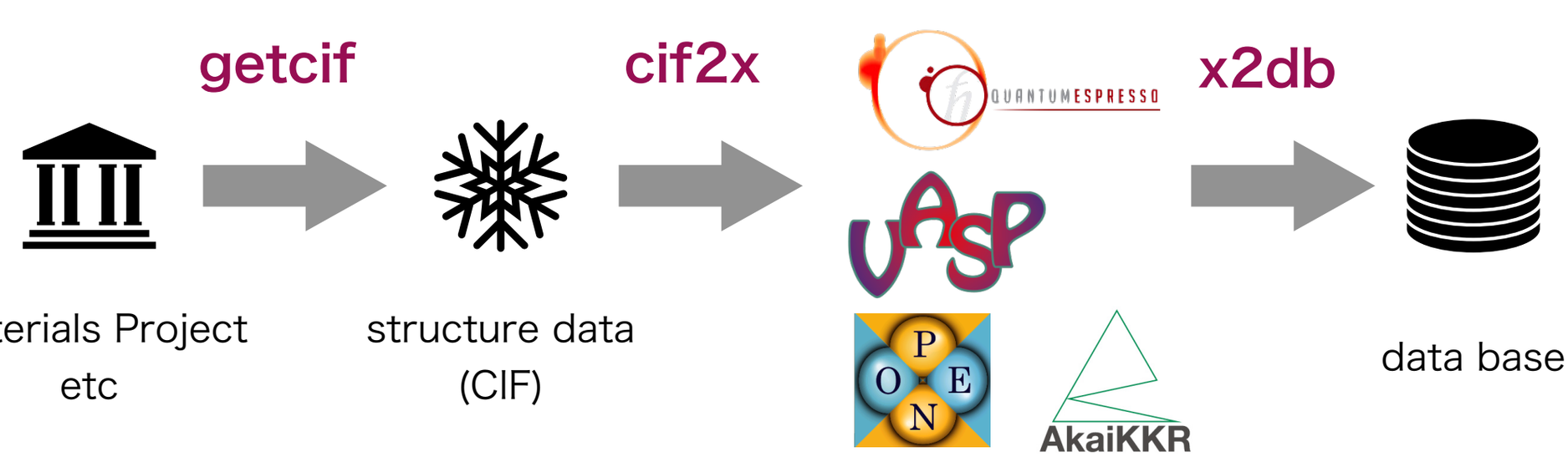
- 機械学習の精度は適切な教師データの準備に大きく依存
- 迅速に大量の教師データを生成するためのツールや環境を整備

大型計算機上での網羅的な第一原理計算の実行を支援するツールの開発

- High-ThroughPut (HTP) tools
  - 2023年度 物性研ソフトウェア開発高度化課題として実施
- 軽量のツールセット
  - 専用マシンや複雑なセットアップなしに、容易に目的のデータセットを構築できるようにする
- プロジェクトサイト
  - https://www.pasums.issp.u-tokyo.ac.jp/htp-tools/

HTP-toolsの構成と機能

① 公開データベースから結晶構造データを検索・一括取得 (getcif) public
② 第一原理計算ソフトウェアの入力データを生成 (cif2x) public
③ データベース構築 (x2db)
④ スパコンやクラスタ計算機で網羅計算の実行を管理 (moller) public



bulk execution

getcif

結晶構造データ取得ツール

- 公開データベースから結晶構造等のデータを取得するツール
  - 現在は Materials Project のデータベースに対応
  - REST API により web service からデータ取得
- 検索条件・取得するデータの種類を指定する
  - YAML形式で記述
    - 条件やデータ種別は `--help` に一覧
- 該当した物質ごとに material_id でフォルダを作成、項目ごとにファイルに書き出し
- 結晶構造データはCIF形式で出力

```yaml
database:
  target: materials project

option:
  output_dir: result

properties:
  band_gap: < 1.0
  is_stable: true
  is_metal: false
  formula: "**O3"
  spacegroup_symbol: Pm-3m

fields: |
  structure
  band_gap
  symmetry
```

検索条件

取得する項目

cif2x

第一原理計算ソフトウェア入力作成ツール

- 結晶構造データ(CIF形式)とパラメータ定義ファイル(YAML形式)から、
  第一原理計算ソフトウェアの入力ファイルを生成するツール
- Quantum ESPRESSO, VASP, OpenMX, AkaiKKR に対応
- 使用イメージ

```
$ cif2x -t qe input_qe.yaml material.cif
```

→ scf.in, band.in, dos.in など、input_qe.yaml の設定に従って生成

```
$ cif2x -t vasp input_vasp.yaml material.cif
```

→ INCAR, KPOINTS, POSCAR, POTCAR のセットを生成

- プロジェクトリポジトリ:

https://github.com/issp-center-dev/cif2x

x2db

第一原理計算プログラムの出力ログ解析ツール

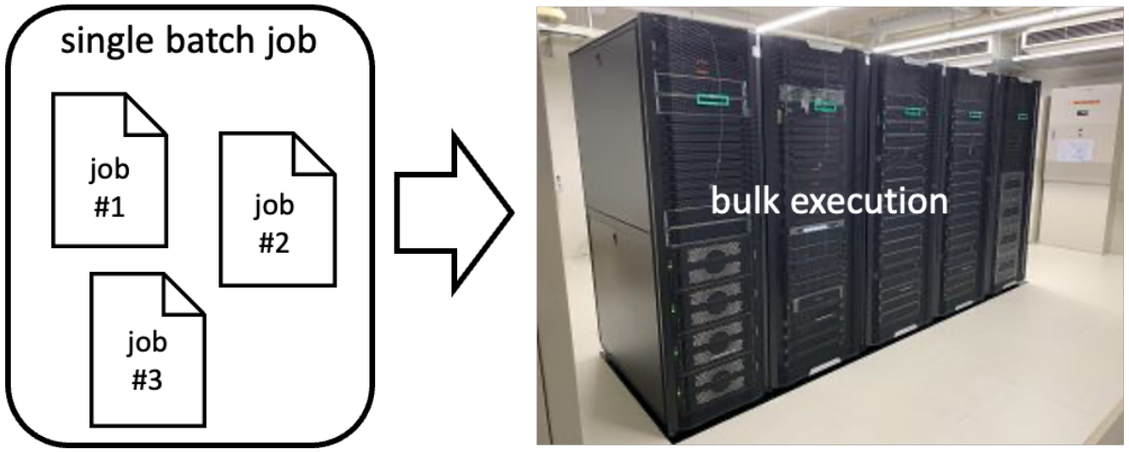
- 出力ログを解析してデータを抽出、JSON形式で書き出すツール [開発中]
  - 標準出力に書き出されたデータを抽出する
  - 系統的・自動化に適した形でツールとして整備する
    - データベースとの接続・データ投入へ
    - パラメータ自動チューニングのためのデータ取り出しなど
- 既存ライブラリを主に利用

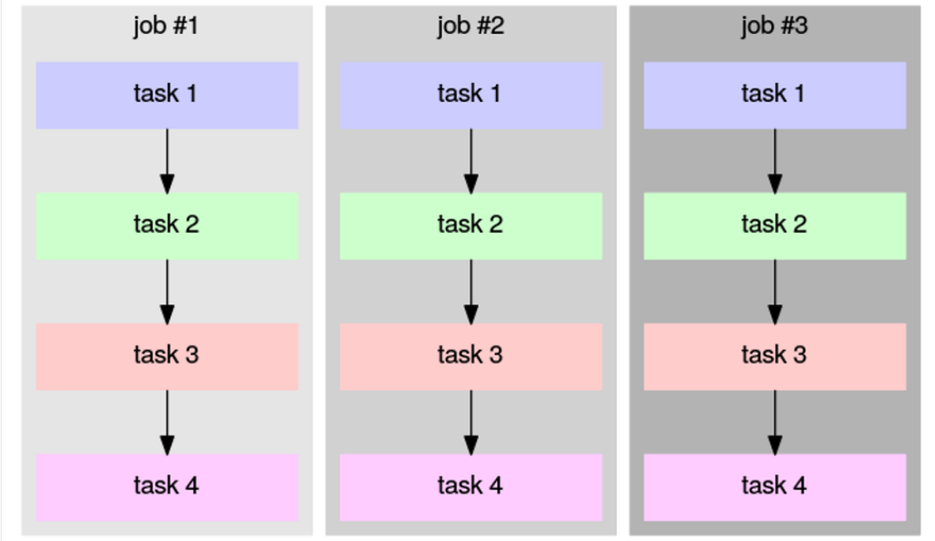
target	log files	libraries	used in x2db
VASP	OUTCAR vasprun.xml	pymatgen.io.vasp ase.calculators.vasp	pymatgen
QE	prefix.save/pwscf.xml stdout	pymatgen.io.pwscf ase.calculators.espresso	pymatgen xml parser
OpenMX	label.dat, label.dat#, label.out	ase.calculators.openmx	ase
AkaiKKR	stdout	AkaiKKRPythonUtil AiiDA plugin	inhouse code

moller

網羅計算ツール

- スパコン上でジョブのバルク実行を支援するツール
  - バルク実行=大型ジョブクラス内で複数のタスクを一括並列実行





- マルチプラットフォーム対応:
  - 物性研スパコン: ohtaka (slurmジョブスケジューラ)・kugui (PBS)
  - 一般的なクラスター計算機・ワークステーション
- レジューム・リトライ:
  - 中断したジョブの再開・失敗したタスクの再試行
- プロジェクトリポジトリ:
  - https://github.com/issp-center-dev/Moller

ジョブをまとめて実行する

パラメータサーチなど、条件を変えながら
同じ計算を大量に実行する場合に
スパコンを効率よく使いたい

多数のジョブを
1つの大型のジョブクラスで
一括して実行 (バルク実行)

ジョブをまとめて実行する

- スパコンでプログラムを実行するには:
  - 通常はバッチジョブ形式で実行する
    - ユーザはノード数・実行時間などをリクエスト
      ↓
      ジョブ管理システムがリソースを割り当て
  - リソース量や実行内容をジョブスクリプトに記述
    - シェルスクリプト形式
    - コメント部分にジョブスケジューラへの指示を記載
- 最大ノード数や実行時間などで区分(システムによる)
  - 例: 物性研システムB　　ジョブクラス例: F4cpu
    - ノード数: 1, 4, 16, 36, 72, 144 (最大)
    - 実行時間: 12時間, 24時間
    - ノード種類: CPUノード, FATノード(大容量メモリ)

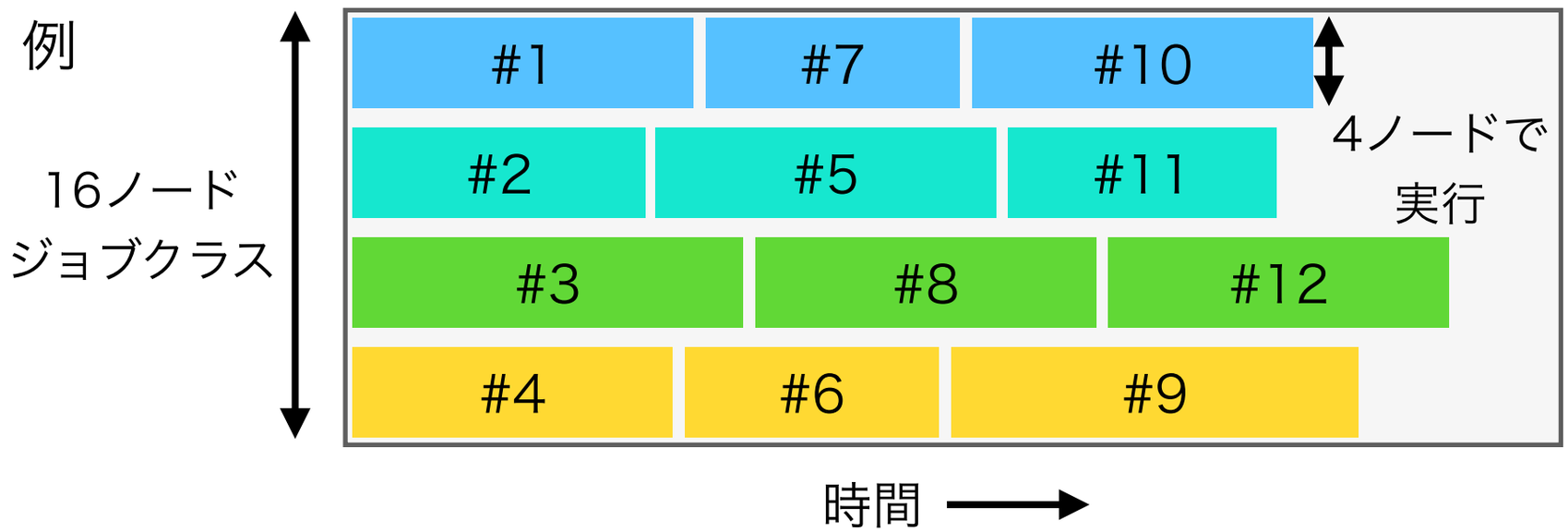
```
#!/bin/sh
#SBATCH -p i8cpu
#SBATCH -N 2
#SBATCH -n 8
#SBATCH -c 32
#SBATCH -t 00:10:00

ulimit -s unlimited


srun HPhi -s stan.in
```

ジョブをまとめて実行する

- 例: 4ノードを使ったMPIプログラムの実行を、12通りの入力データに対して行う

  A. 小型のジョブクラス(4ノード・クラス)で個別のバッチジョブとして実行する

  B. 大型のジョブクラス (例: 16ノード・クラス) のバッチジョブの中にジョブを束ねて実行する (バルク実行)



ジョブをまとめて実行する

- メリット:
  - ノード内の複数コアを効率よく使える
  - 細かいジョブを大量に投入するとジョブスケジューラの負荷・オーバーヘッドが大きい →1つのバッチジョブにまとめて実行
  - 大型のジョブクラスは比較的空いている(こともある)
- デメリット:
  - バルク実行のやり方はプラットフォーム依存
  - →プラットフォーム依存性を隠蔽する汎用ツールを作成

ジョブをまとめて実行する

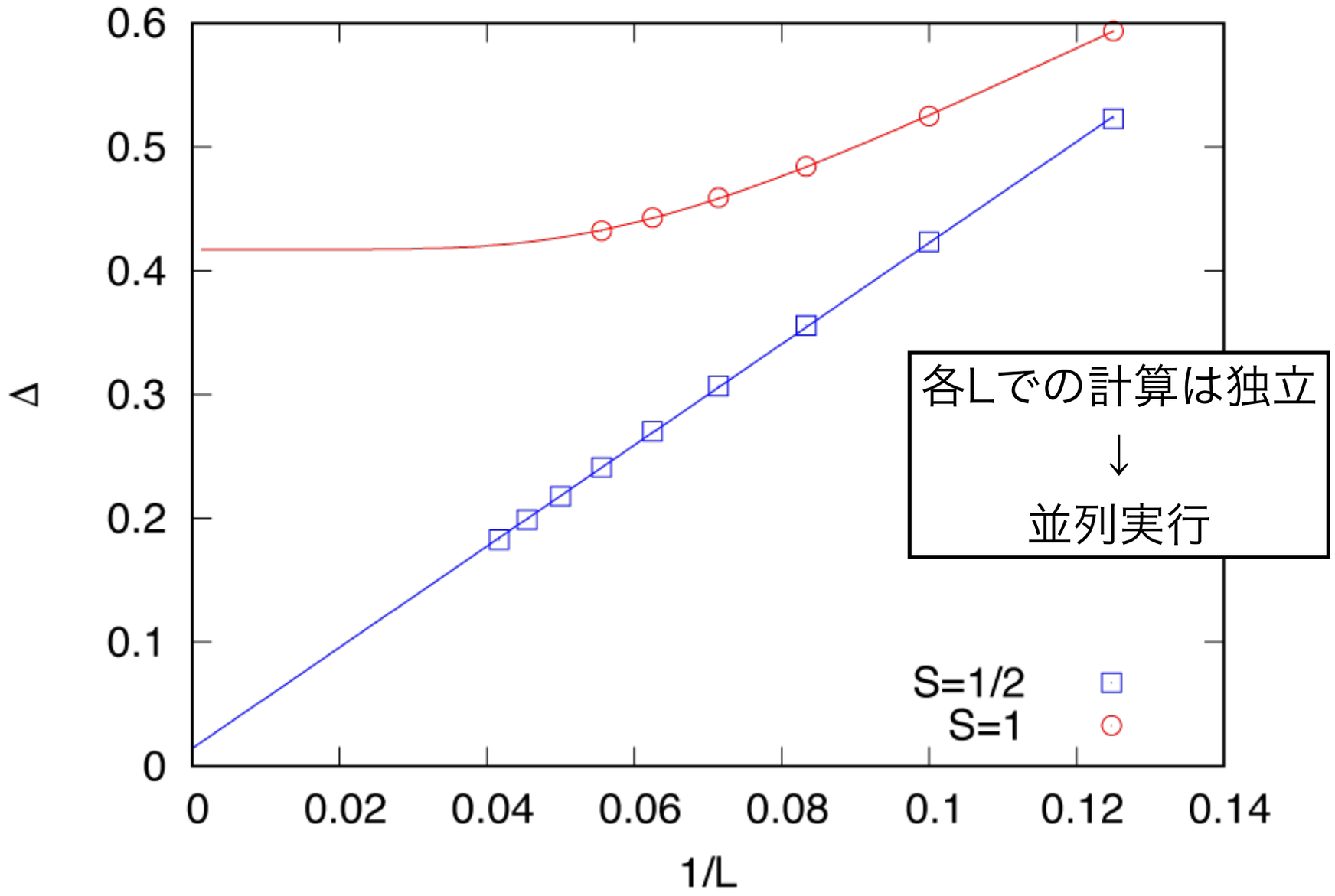
- **moller**とは、

> 「スパコン上でプログラムをバルク実行する
> ジョブスクリプト」を生成するツール

- スパコンのフロントエンド上、またはオフラインで実行する
- YAML形式(構造化テキスト)でジョブの内容を記述
- バルク実行に関するプラットフォーム依存な詳細を隠蔽
- マルチプラットフォーム (現在は物性研スパコン ohtaka, kugui に対応)

mollerを使う

- 例として、厳密対角化ソフトウェア HΦ を用い、S=1/2 反強磁性ハイゼンベルク鎖の励起ギャップΔのシステムサイズL依存性を評価する。



mollerを使う

1. mollerの入力ファイル input.yaml を用意する

input.yaml

```
name: HPhi
description: AFH chain

platform:
  system:  ohtaka
  queue:   i8cpu
  node:    8
  elapsed: 00:30:00

prologue:
  code: |
    source /home/issp/materiapps/
oneapi_compiler_classic-2023.0.0--
openmpi-4.1.5/hphi/hphivars.sh

    ulimit -s unlimited
    export KMP_STACKSIZE=512m
    export UCX_TLS='self,sm,ud'

jobs:
  hphi:
    description: run HPhi
    node: [2,8,32]
    run: |
      srun HPhi -s stan.in
```

molleを使う

1. mollerの入力ファイル input.yaml を用意する

input.yaml

```
name: HPhi
description: AFH chain

platform:
  system:   ohtaka
  queue:    i8cpu
  node:     8
  elapsed:  00:30:00

prologue:
  code: |
    source /home/issp/materiapps/
oneapi_compiler_classic-2023.0.0--
openmpi-4.1.5/hphi/hphivars.sh

    ulimit -s unlimited
    export KMP_STACKSIZE=512m
    export UCX_TLS='self,sm,ud'

jobs:
  hphi:
    description: run HPhi
    node: [2,8,32]
    run: |
      srun HPhi -s stan.in
```

バッチジョブの設定

各タスクの並列度

single版の job.sh

```
#!/bin/sh
#SBATCH -p i8cpu
#SBATCH -N 2
#SBATCH -n 8
#SBATCH -c 32
#SBATCH -t 00:10:00

source /home/issp/materiapps/
oneapi_compiler_classic-2023.0.0--
openmpi-4.1.5/hphi/hphivars.sh

ulimit -s unlimited
export KMP_STACKSIZE=512m
export UCX_TLS='self,sm,ud'

srun HPhi -s stan.in
```

プラットフォーム固有の設定

計算内容

mollerを使う

2. moller の実行環境をセットアップする

実際は1行

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

(物性研システムB(ohtaka)の場合)

3. moller を実行してジョブスクリプトを生成する

```
$ moller input.yaml -o job.sh
```

- ジョブスクリプト job.sh が生成される

4. リストファイルを作成する

list.dat

```
L_8
L_10
L_12
L_14
L_16
L_18
L_20
L_22
L_24
```

- パラメータセットごとに個別のディレクトリを用意する
- ディレクトリのリスト list.dat を作成する

mollerを使う

ジョブ投入コマンドは
システムによる

5. バッチジョブを投入する

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

6. 各ジョブのステータスは

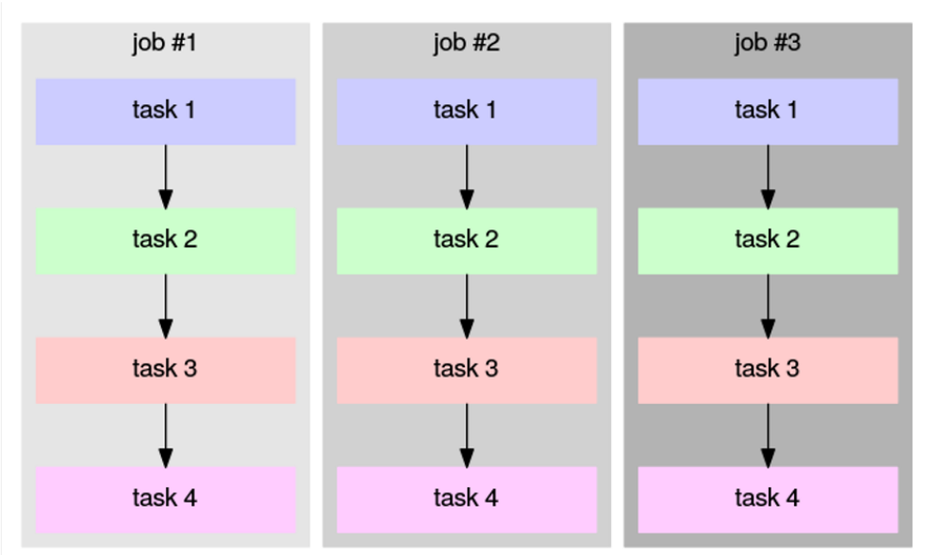
```
$ moller_status input.yaml list.dat
```

で確認できる。

```
| job    | hphi     |
|--------|----------|
| L_8    | o        |
| L_10   | o        |
| L_12   | o        |
| L_14   | o        |
| L_16   | o        |
| L_18   | o        |
| L_20   | o        |
| L_22   | o        |
| L_24   | o        |
```

mollerの機能

- 一つのパラメータセットに対する計算("ジョブ")を複数の"タスク"で構成できる
  - "タスク"ごとにジョブを並列して実行する
  - "逐次タスク"(ジョブによらない処理)を指定可
- 前のタスクが失敗した場合は、後続のタスクは実行されない
  (ジョブごと。他のジョブは影響を受けない)



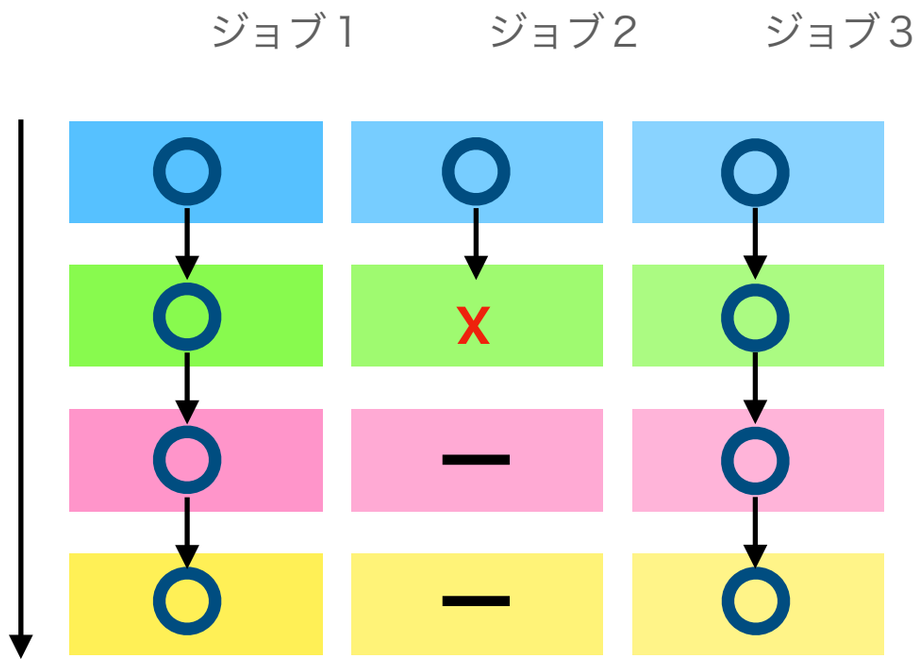
mollerの機能

- レジューム・リトライ
  - バッチジョブが途中で終了した場合、未処理のタスクから実行を継続する(レジューム)

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

  - 失敗したタスクを再実行する(リトライ)

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



mollerをインストールする

- moller はオープンソース・ソフトウェアとして公開
  - GitHub リポジトリから取得・インストールできる

```
$ git clone https://github.com/issp-center-dev/Moller.git
$ cd Moller
$ python3 -m pip install .
```

- system = default を指定すると、一般のクラスタ計算機(ジョブスケジューラなし)で動作するスクリプトを生成

mollerをインストールする

- moller が生成するジョブスクリプトの構造
  - GNU parallel を用いて並列実行

    parallel [task] ::: [job list]

    - [task] ... taskの内容をbashの関数化
      - task 内の並列実行コマンドをプラットフォームに合わせて書き換え
    - [job list] ... list.dat の内容を展開
- リソース管理
  - ノード数・コア数から同時実行数を計算
  - バッチジョブに割り当てられたノード・コアを各タスクに配分

    この部分はジョブスケジューラとMPIランタイムに依存

    プラットフォームごとの書き換えが必要

mollerをインストールする

- 現状のプラットフォーム対応状況
  - ジョブスケジューラ:
    - SLURM, PBS系 (PBS Pro, Torque), NQSV
    - ToDo: Fujitsu, (SGE)
  - MPIランタイム
    - OpenMPI, Intel MPI
    - ToDo: MPICH, MVAPICH, etc
  - (アクセラレータは別途対応が必要)
- 対応プラットフォームを拡充予定
  - 国内の主要なスパコンサイトへの展開

対応プラットフォーム

- 物性研究所
  - システムB ohtaka (SLURM)
  - システムC kugui (PBS Pro)
- クラスタ計算機 (Torque)

(対応中)

- 東北大 AOBA-S/A/B (NQSV)

まとめと展望

- 物性研スパコンとソフトウェア開発高度化プロジェクトの紹介
  - 全国共同利用により国内の物性コミュニティーに計算資源を提供
  - 物性研スパコンを有効活用するためのアプリ開発支援プロジェクトを実施
- ハイスループット計算ツール HTP-tools と moller の紹介
  - moller の開発・構成・機能
  - HΦを用いた実行例を解説
- 展望
  - moller の対応プラットフォーム拡大
  - 機能拡充でより使いやすいツールに