

## 並列プログラムのビルドと実行

(最終更新日: 2024/5/9: mvapich の --with-pbs オプションについての記述を追加, --disable-cuda を削除)

- [OpenMP 並列](#)
- [非 OpenMP スレッド並列](#)
- [MPI 並列](#)
  - [RCCS 提供の環境を使う](#)
  - [自身で環境を構築する](#)
  - [Open MPI 環境の構築](#)
  - [実行時の注意](#)

### OpenMP 並列

OpenMP を有効にしたバイナリを作成し、ジョブスクリプト中で `omphthreads=` で並列数を指定すれば問題無く実行できるはずです。(ジョブスクリプト中で `OMP_NUM_THREADS` 環境変数を設定すれば、値を置き換えることも一応可能ですが、MPI-OpenMP ハイブリッド並列の場合は利用する MPI 環境によってはおかしいことが起きるかもしれません。)

MKL や OpenBLAS 等の OpenMP 対応のライブラリを使う場合も同様に `omphthreads=` を指定すれば問題ありません。MKL\_NUM\_THREADS や OPENBLAS\_NUM\_THREADS 環境変数で MKL や OpenBLAS 部分だけスレッド数を変えるようなことも可能です。

libiomp5

が存在しないというエラーが出る場合にはインテルコンパイラーの実行時環境の読み込みが必要です。他に妙なエラーが出る OpenMP のライブラリが読み込まれている可能性があります。必要な module を load あるいは unload するなどして正しいライブラリを使うように設定してください。

### 非 OpenMP スレッド並列

OpenMP では無いスレッド並列の場合はジョブスクリプトの `omphthreads` は特に何もおこないません。ジョブスクリプトやインプットファイル中で適宜スレッド数を指定してください。OpenMP を一切使わないのであれば、ジョブスクリプトヘッダーの `omphthreads` には 1 を入れても、スレッド数を入れておいてもどちらでも問題ありません。CPU 点数の消費に関しても影響はありません(ncpus, ngpus と実行時間のみで決定されるためです)。

ただし、OpenBLAS や MKL のように OpenMP を使うものが紛れ込んでいる場合はその実行方法に合わせて適宜設定しておく必要があるかもしれません。

### MPI 並列

#### RCCS 提供の環境を使う

Intel MPI、Open MPI、MVAPICH2 の module

を用意しています。適宜それらをご利用ください。ただし、インテルコンパイラを必要とする場合はまずそれをご自身で導入 [こちらのページ](#)にも何か参考となる情報があるかもしれません。GCC と Intel MPI を組み合わせて使う場合は oneAPI

環境の設定は不要です。

module の読み込み例 (Open MPI, MVAPICH, Intel MPI): gcc11 と合わせて使う場合(先頭の \$ は入力しないでください)

```
$ module load openmpi/4.1.6/gcc11
```

```
$ module load mvapich/2.3.7/gcc11
```

```
$ module load intelmpi/2021.11
```

(注意) Intel MPI の場合は gcc のバージョン設定は不要です。ただし、指定したバージョンの gfortran に対応したモジュール (mpi.mod) が存在しない場合があります。その場合は Intel MPI のより新しいバージョンを使ったり、gcc のバージョンを変えるなどして対応してください。

読み込みができれば mpicc, mpicxx, mpif90 などのコマンドが使えるようになります。AOCC と組み合わせる場合は、aocc と対応する MPI 環境を読み込みようにしてください。

```
$ module load aocc/4.2.0
```

```
$ module load openmpi/4.1.6/aocc4.1
```

## 自身で環境を構築する

- Open MPI => 後述
- Intel MPI 環境(インテルコンパイラ使わない場合) => oneAPI HPC Toolkit を導入してください。
  - 上述のように RCCS 提供の Intel MPI を使うことも可能です。
- Intel Compiler + Intel MPI 環境 => oneAPI Base Toolkit と HPC Toolkit を自身のホームディレクトリに導入してください
  - [こちらのページ](#)にも何か参考となる情報があるかもしれません。
- MVAPICH => 特別な指定は必要ありません。CC, CXX, FC, F77 でコンパイラを指定し、configure && make && make install すれば通常は大丈夫です。追加で make check も実行するとより安全です。
  - configure 時に --with-pbs=/apl/pbs/22.05.11 をつけるとキューイングシステムとより緊密な連携が可能です
    - (オプションを指定しない場合でも \$PBS\_NODEFILE は自動的に考慮されます)
  - configure 時に --enable-fortran --enable-cxx あたりも追加しても良いかもしれません。

後は用意した mpicc, mpiicc, mpif90, mpiifort

などを使ってアプリをビルドすることになるかと思います。上記以外の MPI 環境については未検証です。

## Open MPI 環境の構築

CC, CXX, FC でコンパイラを指定するところは普通ですが、キューイングシステムとの連携を図るために configure に

```
--with-tm=/apl/pbs/22.05.11
```

の指定を推奨しています。キューイングシステムと連携させることで、machine file の指定などを省略できたり、各種環境変数の指定を省略できたりする利点があります。RCCS で導入している Open

MPI でもこの指定を追加しています。(例: [Open MPI 4.1.6](#))上記に加えて `--enable-mpi-cxx` や `--enable-mpi1-compatibility` を加えると便利な場合があるかもしれません。

CUDA-aware 版の Open MPI が必要な場合は、あらかじめ module を読み込んでおいた上で configure に以下のように指定します(例: 12.2 update 2 の場合)。他の部分については CPU 版と同じで問題ないはずです。

```
--with-cuda=/apl/cuda/12.2u2
```

GPU も利用する MPI 並列の動作検証については `ccgpu` で行うことができます。`ccfep` にログインした状態で `ssh cccpu` を実行すればログインできます。

NVIDIA HPC SDK 向けの Open MPI の場合は上記 `--with-tm=/apl/pbs/22.05.11` に加えて `CFLAGS` などに `-fPIC` オプションを指定する必要があります。CUDA については `--with-cuda=/apl/nvhpc/23.9/Linux_x86_64/23.9/cuda` のように指定できます。

## 実行時の注意

ジョブスクリプト中で使う場合は(.bash\_profile 等に記述していなければ)スクリプト中で以下のように MPI 環境を読み込む必要があります。ホストのリストは `mpiprocs` の指定に従って環境変数 `PBS_NODEFILE` で指定されるファイル中に記述されます。`mpirun: command not found` のようなエラーが出る場合は、MPI 環境を読み込めているのかチェックしてください。また、`module` コマンドに `-s` オプションを追加している場合、読み込み時のエラーが捨てられてしまっている可能性があります。そのような場合は `-s` オプションを外してお試してください。

MPI 環境がキューイングシステムを認識している場合は以下の例のように何もしなくても大丈夫です。

```
#!/bin/sh
#PBS -l select=1:ncpus=32:mpiprocs=16:ompthreads=2
#PBS -l walltime=24:00:00

cd ${PBS_O_WORKDIR}
# list of nodes given by queuing system
cat ${PBS_NODEFILE}

module -s purge
module -s load openmpi/4.1.6/gcc11

MYPROG=/home/users/${USER}/bin/myprog-mpi

# mpirun command from openmpi/4.1.6/gcc11 package
mpirun -np 16 ${MYPROG} input.inp > output.out
```

MPI 環境がキューイングシステムの存在を考慮してくれない場合は `machine file` や `host file` と呼ばれるファイルの指定が必要です。例えば Open MPI ならば `mpirun` コマンド実行時に `--hostfile ${PBS_NODEFILE}` を追加で指定する必要があります。