

ジョブ投入に関するTips

基本

- ▶ 各アプリ用のサンプルが /local/apl/lx/(アプリ名)/samples にあります。テンプレートとしてもご利用できます。
- ▶ sh/bash での基本形

```
#!/bin/sh
#PBS -l select=...
#PBS -l walltime=[時間]

# PBS_O_WORKDIR はジョブを投入した時の作業ディレクトリ(CWD)に当たります
# cctep でのジョブの投入時と演算ノードでの作業ディレクトリをそろえるために
# スクリプト内でディレクトリ移動を行います
if [ ! -z ${PBS_O_WORKDIR} ]; then
  cd ${PBS_O_WORKDIR} # PBS_O_WORKDIR 変数が存在している場合、そのディレクトリへ移動
fi

(実際の処理)
```

- ▶ csh/tcsh での基本形

```
#!/bin/csh -f
#PBS -l select=...
#PBS -l walltime=[時間]

if ( $?PBS_O_WORKDIR ) then
  cd $PBS_O_WORKDIR # sh/bash の場合と同じ
endif

(実際の処理)
```

ジョブヘッダのサンプル(select文のサンプル)

- ▶ select= の直後にくる数字は **ノードの数**を表します(省略すると 1)。
- ▶ 他の数字(ncpus,mpiprocs,ompthreads,ngpus)は **ノードあたりの数字**です。
- ▶ OMP_NUM_THREADS 環境変数は自動的に ompthreads の設定値になります。
- ▶ /local/apl/lx に導入されている MPI 環境(IntelMPI もしくは OpenMPI)の場合、実行ホストのリストを指定する必要はありません。
 - ▶ mpirun -np (mpiprocsで指定した数字) command options の形で大丈夫です。
 - ▶ (ご自分で OpenMPI をビルドされる場合でも、configure 時に --with-tm=/local/apl/lx/pbs14 を加えていれば実行ホストリストの指定は必要ありません)
- ▶ GPU 利用時に CUDA_VISIBLE_DEVICES 環境変数に特別な注意を払う必要はありません。(リソースはジョブサーバできちんと管理されています)

jobtype=small, 1 ノード, ノードあたり: 40 CPU コア, MPI*40, OpenMP 無し (Flat MPI) (合計 1 ノード, 40 CPU コア), 72 時間(3日)

```
#PBS -l select=1:ncpus=40:mpiprocs=40:ompthreads=1:jobtype=small
#PBS -l walltime=72:00:00
```

jobtype=small, 4 ノード, ノードあたり: 40 CPU コア, MPI*40, OpenMP 無し (合計 4 ノード, 160 CPU コア), 168 時間(一週間)

```
#PBS -l select=4:ncpus=40:mpiprocs=40:ompthreads=1:jobtype=small
#PBS -l walltime=168:00:00
```

jobtype=large, 1 ノード, ノードあたり: 40 CPU コア, MPI*20, OpenMP*2 (合計 1 ノード, 40 CPU コア), 30 分

```
#PBS -l select=1:ncpus=40:mpiprocs=20:ompthreads=2:jobtype=large
#PBS -l walltime=00:30:00
```

jobtype=large, 2 ノード, ノードあたり: 40 CPU コア, MPI*40, OpenMP 無し (合計 2 ノード, 80 CPU コア), 1 時間

```
#PBS -l select=2:ncpus=40:mpiprocs=40:ompthreads=1:jobtype=large
#PBS -l walltime=01:00:00
```

jobtype=core, 1 CPU コア, 168 時間

```
#PBS -l select=1:ncpus=1:mpiprocs=1:ompthreads=1:jobtype=core
#PBS -l walltime=168:00:00
```

jobtype=core, 12 CPU コア, MPI*4, OpenMP 無し, 12 時間

```
#PBS -l select=1:ncpus=12:mpiprocs=4:ompthreads=1:jobtype=core
#PBS -l walltime=12:00:00
```

(利用可能なメモリ量はncpusの値に比例します。そのため、jobtype=core で MPI のプロセス数を増やさずにメモリ量を増やしたい場合はこの記述が有効です。)

jobtype=core, 18 CPU コア, MPI*9, OpenMP*2, 3時間

```
#PBS -l select=1:ncpus=18:mpiprocs=9:ompthreads=2:jobtype=core
#PBS -l walltime=02:00:00
```

jobtype=core, 32 CPU コア, MPI*8, OpenMP*4, 168 時間

```
#PBS -l select=1:ncpus=32:mpiprocs=8:omphreads=4:jobtype=core
#PBS -l walltime=168:00:00
```

jobtype=core, 36 CPU コア, OpenMP*36, 168 時間

```
#PBS -l select=1:ncpus=36:mpiprocs=1:omphreads=36:jobtype=core
#PBS -l walltime=168:00:00
```

jobtype=gpu, 1 CPU コア, GPU*1, 24時間

```
#PBS -l select=1:ncpus=1:mpiprocs=1:omphreads=1:ngpus=1:jobtype=gpu
#PBS -l walltime=24:00:00
```

jobtype=gpu, 12 CPU コア, MPI*12, OpenMP 無し, 1 GPU, 12 時間

```
#PBS -l select=1:ncpus=12:mpiprocs=12:omphreads=1:ngpus=1:jobtype=gpu
#PBS -l walltime=12:00:00
```

ngpus=1 の時は ncpus=12 が最大です。

jobtype=gpu, 4 ノード, ノードあたり: 8 CPU コア, MPI*2, OpenMP*4, 2 GPU (合計 4 ノード, 32 CPU コア, 8 GPU), 12 時間

```
#PBS -l select=4:ncpus=8:mpiprocs=2:omphreads=4:ngpus=2:jobtype=gpu
#PBS -l walltime=12:00:00
```

mpiprocs の値は ngpus の値の倍数である必要があります。実際には GPU だけを使い、CPU を使わない場合でも select 文では CPU を使うという指定が必要です。

jobtype=gpu, 2 ノード, ノードあたり: 24 CPU コア, MPI*24, OpenMP 無し, 2 GPU (合計 2 ノード, 48 CPU コア, 4 GPU), 24 時間

```
#PBS -l select=2:ncpus=24:mpiprocs=24:omphreads=1:ngpus=2:jobtype=gpu
#PBS -l walltime=24:00:00
```

jobtype=gpu or gpup, ngpus=2 で複数ノードを使う場合、最大で指定数の倍のノード(この場合ならば 4 ノード)にプロセスが散らばる可能性があります。

jobtype=gpuv, 1 ノード, ノードあたり: CPU 4 コア, MPI*4, OpenMP 無し, 4 GPU, 30 分

```
#PBS -l select=1:ncpus=4:mpiprocs=4:omphreads=1:ngpus=4:jobtype=gpuv
#PBS -l walltime=00:30:00
```

jobtype=gpuv, 1 ノード, ノードあたり: CPU 8 コア, MPI*8, OpenMP 無し, 8 GPU, 30 分

```
#PBS -l select=1:ncpus=8:mpiprocs=8:omphreads=1:ngpus=8:jobtype=gpuv
#PBS -l walltime=00:30:00
```

(プログラムは mpirun -np 8 (prog) のように実行します。jobtype=gpuv の場合は複数ノードの利用はできません。)

jobtype=gpuv, 1 ノード, ノードあたり: CPU 24 コア, MPI*8, OpenMP*3, 8 GPU, 30 分

```
#PBS -l select=1:ncpus=24:mpiprocs=8:omphreads=3:ngpus=8:jobtype=gpuv
#PBS -l walltime=00:30:00
```

(プログラムは mpirun -np 8 (prog) のように実行します。jobtype=gpuv の場合は複数ノードの利用はできません。)

jobtype=gpu, 8ノード, ノードあたり: CPU 2 コア, MPI*2, OpenMP*1, 2 GPU, 24 時間

```
#PBS -l select=8:ncpus=2:mpiprocs=2:omphreads=1:ngpus=2:jobtype=gpu
#PBS -l walltime=24:00:00
```

以下の表記でも同じ意味になります。jobtype=gpu, ngpus=2 で複数ノード指定の場合は特殊な取扱いとなっています。奇数台の GPU を使う場合には下側の表記をご利用下さい。

```
#PBS -l select=16:ncpus=1:mpiprocs=1:omphreads=1:ngpus=1:jobtype=gpu
#PBS -l walltime=24:00:00
```