

# サンプルジョブの実行手順

自然科学研究機構  
岡崎共通研究施設  
計算科学研究センター(RCCS)

# 更新履歴

- 2019/9/6 初稿作成
- 2020/1/15 ターミナル表示を画像からテキストに変更

# イントロダクション

この資料ではRCCSで導入したアプリケーションのサンプルジョブ実行手順を説明します。

これらサンプルジョブの関連ファイルはジョブスクリプトのテンプレートとしても利用いただけます。RCCSで導入したソフトを用いる場合はもちろん、ご自身でビルドした別バージョンを使う際にも有用でしょう。

この資料では、分子動力学計算パッケージ Gromacs と、量子化学計算パッケージ GAMESS を例に説明しますが、他のパッケージの場合でも同様の手順でサンプルを実行できます。

# 目次

- 導入済みソフトウェア
- サンプルジョブファイルの場所
- サンプルの構成
- サンプル実行
- ヒント

# 導入済みソフトウェア(1): アプリリスト

- アプリのリストは [https://ccportal.ims.ac.jp/installed\\_applications](https://ccportal.ims.ac.jp/installed_applications) にて確認できます。
- また、ログイン後に **“module avail”** コマンドでも以下のように確認できます。(apl\_ex カテゴリ内が通常のアプリケーションです。)

```
[user@ccfep5]$ module avail

----- /local/apl/1x/modules/suite -----
intel_parallelstudio/2015update1          intel_parallelstudio/2018update2          intel_parallelstudio/2019update5          scl/devtoolset-6
intel_parallelstudio/2017update4          intel_parallelstudio/2018update4(default) scl/devtoolset-3                          scl/devtoolset-7
intel_parallelstudio/2017update8          intel_parallelstudio/2019update1          scl/devtoolset-4                          scl/devtoolset-8

----- /local/apl/1x/modules/comp -----
cuda/10.1                                cuda/8.0                                intel/15.0.1                                intel/17.0.8                                intel/18.0.5(default) intel/19.0.5                                pgi/16.5                                pgi/18.1(default)
cuda/7.5                                  cuda/9.1(default)                        intel/17.0.4                                intel/18.0.2                                intel/19.0.1                                julia/1.3.1                                pgi/17.5

----- /local/apl/1x/modules/apl -----
mpi/intelmpi/2017.3.196                    mpi/intelmpi/2019.1.144                    mpi/openmpi/2.1.3/gnu5.3                    mpi/openmpi/2.1.3/intel17                    mpi/openmpi/3.1.0/gnu5.3                    mpi/openmpi/3.1.0/intel17                    mpi/openmpi/4.0.0/gnu5.3                    mpi/openmpi/4.0.0/intel17
mpi/intelmpi/2017.4.262                    mpi/intelmpi/2019.5.281                    mpi/openmpi/2.1.3/gnu6.3                    mpi/openmpi/2.1.3/intel18                    mpi/openmpi/3.1.0/gnu6.3                    mpi/openmpi/3.1.0/intel18                    mpi/openmpi/4.0.0/gnu6.3                    mpi/openmpi/4.0.0/intel18
mpi/intelmpi/2018.2.199                    mpi/intelmpi/5.0.2.044                    mpi/openmpi/2.1.3/gnu7.3                    mpi/openmpi/2.1.3/intel19                    mpi/openmpi/3.1.0/gnu7.3                    mpi/openmpi/3.1.0/intel19                    mpi/openmpi/4.0.0/gnu7.3                    mpi/openmpi/4.0.0/intel19
mpi/intelmpi/2018.4.274                    mpi/openmpi/2.1.3/gnu4.8                    mpi/openmpi/2.1.3/gnu8.3                    mpi/openmpi/3.1.0/gnu4.8                    mpi/openmpi/3.1.0/gnu8.3                    mpi/openmpi/3.1.0/gnu8.3                    mpi/openmpi/4.0.0/gnu4.8                    mpi/openmpi/4.0.0/gnu8.3
mpi/intelmpi/2019                          mpi/openmpi/2.1.3/gnu4.9                    mpi/openmpi/2.1.3/intel15                    mpi/openmpi/3.1.0/gnu4.9                    mpi/openmpi/3.1.0/intel15                    mpi/openmpi/4.0.0/gnu4.9                    mpi/openmpi/4.0.0/intel15

----- /local/apl/1x/modules/apl_ex -----
GRRM/11-g09                                espresso/5.1.2                                genesis/1.3.0-CUDA                            gromacs/2018.1/gnu                            gromacs/2018.8/intel-CUDA                    lammps/16Mar18/intel                            openmolcas/20190604
GRRM/14-g09(default)                      espresso/5.4                                genesis/1.4.0                                gromacs/2018.1/intel                            gromacs/2019.2/gnu                            lammps/16Mar18/intel-CUDA                       orca/4.2.1
abinit/7.8.2                                espresso/6.1                                genesis/1.4.0-CUDA                            gromacs/2018.3/gnu                            gromacs/2019.2/gnu-CUDA                       lammps/22Aug18/intel                            psi4/1.1
abinit/8.8.3(default)                      espresso/6.3(default)                        gromacs/2016.1/intel                            gromacs/2018.3/gnu-CUDA                       gromacs/2019.2/intel                            lammps/22Aug18/intel-CUDA                       reactionplus/1.0
amber/16/bugfix10                          gamess/2018Feb14                            gromacs/2016.1/intel-CUDA                       gromacs/2018.3/intel                            gromacs/2019.2/intel-CUDA                    lammps/22Aug18/intel-CUDA-volta                 siesta/3.1
amber/16/bugfix15                          gamess/2018Sep30(default)                    gromacs/2016.3/intel                            gromacs/2018.3/intel-CUDA                       gromacs/2019.4/gnu                            lammps/7Aug19/intel                            smash/2.2.0
amber/18/bugfix1                            gamess/2019Sep30                            gromacs/2016.3/intel-CUDA                       gromacs/2018.6/gnu                            gromacs/2019.4/gnu-CUDA                       lammps/7Aug19/intel-CUDA                       smash/2.2.0
amber/18/bugfix11-volta                    gaussian/g09/b01                            gromacs/2016.4/intel                            gromacs/2018.6/gnu-CUDA                       gromacs/2019.4/intel                            molcas/8.2                                       turbomole/7.2.1-MPI
amber/18/bugfix12                          gaussian/g09/c01                            gromacs/2016.4/intel-CUDA                       gromacs/2018.6/intel                            gromacs/2019.4/intel-CUDA                    molpro                                           turbomole/7.2.1-SMP
amber/18/bugfix16                          gaussian/g09/d01                            gromacs/2016.5/gnu                            gromacs/2018.6/intel-CUDA                       gromacs/5.1.4/intel                            namd/2.11                                       turbomole/7.2.1-serial
autodock/4.2.6                             gaussian/g09/e01                            gromacs/2016.5/gnu-CUDA                       gromacs/2018.7/gnu                            gromacs/5.1.4/intel-CUDA                       namd/2.11-CUDA                                  turbomole/7.3-MPI(default)
cp2k/6.1.0/gnu                             gaussian/g16/a03                            gromacs/2016.5/intel                            gromacs/2018.7/gnu-CUDA                       gromacs/5.1.5/gnu                            namd/2.13-CUDA                                  turbomole/7.3-SMP
cp2k/6.1.0/gnu-CUDA                        gaussian/g16/b01                            gromacs/2016.5/intel-CUDA                       gromacs/2018.7/intel                            gromacs/5.1.5/gnu-CUDA                       namd/2.13-CUDA                                  turbomole/7.3-serial
cp2k/6.1.0/intel                            gaussian/g16/c01                            gromacs/2016.6/gnu                            gromacs/2018.7/intel-CUDA                       gromacs/5.1.5/intel                            ntchem/2013-5.0-mpi                             turbomole/7.4-MPI
cp2k/6.1.0/intel-CUDA                      genesis/1.1.6                                gromacs/2016.6/gnu-CUDA                       gromacs/2018.8/gnu                            gromacs/5.1.5/intel-CUDA                       ntchem/2013-5.0-mpiomp                         turbomole/7.4-SMP
crystal/14-104                             genesis/1.1.6-CUDA                          gromacs/2016.6/intel                            gromacs/2018.8/gnu-CUDA                       lammps/16Mar18/gnu                             ntchem/2013-5.0-serial                          turbomole/7.4-serial
dirac/18.0                                 genesis/1.3.0                                gromacs/2016.6/intel-CUDA                       gromacs/2018.8/intel                            lammps/16Mar18/gnu-CUDA                       nwchem/6.8                                       turbomole/7.4-serial

----- /local/apl/1x/modules/apl_viewer -----
tuscus/0.8.6 molder/5.7                    nbview/2.0                                vmd/1.9.3

----- /local/apl/1x/modules/apl_util -----
allinea/7.1                                cmake/2.8.12.2(default)                   cmake/3.8.2

----- /local/apl/1x/modules/lib -----
boost/1.53.0(default)                       boost/1.70.0                                mkl/2017.0.3                                mkl/2018.0.2                                mkl/2019.0.1                                spglib/1.11.1(default)
boost/1.59.0                                mkl/11.2.1                                  mkl/2017.0.4                                mkl/2018.0.4(default)                       ncc1/2.3.7-1+cuda9.1(default)

----- /local/apl/1x/modules/misc -----
inteldev intellc pgilic
[user@ccfep5]$
```

(このリストに入っていないものもいくつかあります)

# 導入済みソフトウェア(2): インストール場所

RCCSで導入したアプリケーションは **/local/apl/lx** 以下に保存されています。アプリケーション名とバージョン名をつなげた名前が基本です。

```
[user@ccfep5]$ ls /local/apl/lx/
GRRM@          cuda-10.1/      genesis140-CUDA/  gromacs2018.8-gnu/  lammps22Aug18/  ntchem@
GRRM11/        cuda-7.5/       gromacs@          gromacs2018.8-gnu-CUDA/  lammps7Aug19/  ntchem2013.5.0/
GRRM14/        cuda-8.0/       gromacs2019@     gromacs2019.2-gnu-CUDA/  lammps7Aug19-CUDA/  nwchem@
abinit@        cuda-9.1/       gromacs2016.1/   gromacs2019.2-CUDA/   luscus086/        nwchem68/
abinit782/     dirac180/       gromacs2016.1-CUDA/  gromacs2019.2-gnu/   modules/          openmm41/
abinit883/     dirac180-lm/    gromacs2016.3-CUDA/  gromacs2019.2-gnu-CUDA/  modules.2017/    openmolcas20190604/
allinea@       espresso@        gromacs2016.3-CUDA/bad/  gromacs2019.4-CUDA/  modules.2018/    openmpi213/
allinea71/     espresso512/    gromacs2016.4/      gromacs2019.4-gnu/    molcas@           openmpi213-gnu4.8@
amber@         espresso54/     gromacs2016.4-CUDA/  gromacs2019.4-gnu-CUDA/  molcas82/        openmpi213-gnu4.9/
amber12@       espresso61/     gromacs2016.5/      gromacs2019.4-gnu/    molden@           openmpi213-gnu5.3/
amber12-bf21/  espresso63/     gromacs2016.5-CUDA@  gromacs2019.4-gnu-CUDA/  molden57/        openmpi213-gnu6.3/
amber14@       g09@           gromacs2016.5-CUDA9/  gromacs2019.4-gnu-CUDA/  molpro@          openmpi213-gnu7.3/
amber14-bf11/  g09b01/        gromacs2016.5-gnu/    gromacs2019.4-gnu-CUDA/  molpro2012.1-p137/  openmpi213-gnu8.3/
amber14-bf11.o1d/  g09c01/       gromacs2016.5-gnu-CUDA@  gromacs2019.4-gnu-CUDA/  molpro2015.1-p119/  openmpi213-intel/
amber16@       g09d01/        gromacs2016.5-gnu-CUDA8/  gromacs2019.4-gnu-CUDA/  molpro2015.1-p127/  openmpi213-intel15/
amber16-bf10/  g09d01-test/   gromacs2016.6/        gromacs2019.4-gnu-CUDA/  molpro2015.1-p133/  openmpi213-intel17/
amber16-bf15/  g09e01/        gromacs2016.6-CUDA/    gromacs2019.4-gnu-CUDA/  molpro2015.1-p133-i1708/  openmpi213-intel18@
amber18@       gl6@           gromacs2016.6-gnu/     gromacs2019.4-gnu-CUDA/  molpro2015.1-p133-i1708/  openmpi213-intel19/
amber18-bf1/   gl6a03/        gromacs2016.6-gnu/     gromacs2019.4-gnu-CUDA/  molpro2019.1.2/    openmpi310/
amber18-bf11v/  gl6b01/        gromacs2016.6-gnu-CUDA/  gromacs2019.4-gnu-CUDA/  molpro2019.2.3/    openmpi310-gnu4.8@
amber18-bf12/  gl6c01/        gromacs2018@          gromacs2019.4-gnu-CUDA/  namd@            openmpi310-gnu4.9/
amber18-bf16/  ga-5-5/        gromacs2018.1/        gromacs2019.4-gnu-CUDA/  namd211/         openmpi310-gnu5.3/
anaconda2-2019Jul/  games@         gromacs2018.1-gnu/     gromacs2019.4-gnu-CUDA/  namd213-CUDA/    openmpi310-gnu6.3/
anaconda3-2019Jul/  games2017Apr20/  gromacs2018.3/        gromacs2019.4-gnu-CUDA/  namd213/         openmpi310-gnu7.3/
autodock426/     games2017Nov11/  gromacs2018.3-CUDA/    gromacs2019.4-gnu-CUDA/  nbo60@          openmpi310-gnu8.3/
boost-1.59.0/    games2018Feb14/  gromacs2018.3-gnu/     gromacs2019.4-gnu-CUDA/  nbo6015/        openmpi310-intel/
boost-1.70.0/    games2018Feb14-t1/  gromacs2018.3-gnu-CUDA/  gromacs2019.4-gnu-CUDA/  nbo6018/        openmpi310-intel15/
cmake3.8.2/      games2018Sep30/  gromacs2018.6/        gromacs2019.4-gnu-CUDA/  nbo6018mod/     openmpi310-intel17/
cp2k@           games2019Sep30/  gromacs2018.6-gnu/     gromacs2019.4-gnu-CUDA/  nbo70@          openmpi310-intel18@
cp2k610/        gdv13/          gromacs2018.6-gnu-CUDA/  gromacs2019.4-gnu-CUDA/  nbo702/         openmpi310-intel19/
cp2k610-gnu/    games115/       gromacs2018.7/        gromacs2019.4-gnu-CUDA/  nbo707-14/     openmpi400/
crystal14@      genesis116/     gromacs2018.7-CUDA/    gromacs2019.4-gnu-CUDA/  nbo707/         openmpi400-gnu4.8@
crystal14-104@  genesis116-CUDA/  gromacs2018.7-gnu/     gromacs2019.4-gnu-CUDA/  nbo707-14/     openmpi400-gnu4.9/
crystal14-104-t1/  genesis130/     gromacs2018.8/        gromacs2019.4-gnu-CUDA/  nbopro7/        openmpi400-gnu5.3/
crystal14-104.bad/  genesis130-CUDA/  gromacs2018.8-CUDA/    gromacs2019.4-gnu-CUDA/  nboprov2/       openmpi400-gnu6.3/
cuda@           genesis140/     gromacs2018.8-CUDA/    gromacs2019.4-gnu-CUDA/  ncc1237-1+cuda91/  openmpi400-gnu7.3/
[user@ccfep5]$
```

(実際の表示は利用するターミナルアプリの種類や設定により異なります)

Gromacsならばgromacs(バージョン名)-(コンパイル環境)の形で、GAMESSならばgames(バージョン名)となっています。一方、バージョン名の無いものはそのアプリケーションのデフォルトバージョンです(このファイルは通常リンクです)。

# サンプルジョブファイルの場所

- 各アプリケーションのサンプルはインストールディレクトリ直下の **samples/** ディレクトリにあります。
- **module help (パッケージ名)** の出力にもパスの記述があります。  
(apl\_ex カテゴリの module 限定です)

```
[user@ccfep5]$ ls /local/apl/lx/cp2k610/samples
H20-64.inp  sample-cuda-module.csh  sample-cuda-module.sh  sample-cuda.csh  sample-cuda.sh  sample-module.csh
[user@ccfep5]$ ls /local/apl/lx/cp2k610/samples
H20-64.inp          sample-cuda-module.sh  sample-cuda.sh          sample-module.sh  sample.sh
sample-cuda-module.csh  sample-cuda.csh      sample-module.csh      sample.csh
[user@ccfep5]$ ls /local/apl/lx/amber18-bf12/samples
inpcrd  prmtop          sample-gpu-module.sh*  sample-gpu.sh*      sample-module.sh*  sample.sh*
mdin    sample-gpu-module.csh*  sample-gpu.csh*      sample-module.csh*  sample.csh*
[user@ccfep5]$ ls /local/apl/lx/gromacs2018.7/samples
conf.gro          sample-mpi-module.sh*  sample-threadmpi-module.csh*  sample-threadmpi.sh*
grompp.mdp        sample-mpi.csh*        sample-threadmpi-module.sh*    topol.top
sample-mpi-module.csh*  sample-mpi.sh*        sample-threadmpi.csh*
[user@ccfep5]$ ls /local/apl/lx/gamess2018Sep30/samples
exam01.inp  sample.csh*
[user@ccfep5]$
```

例:

CP2K 6.1.0 Intelコンパイラ版

=> /local/apl/lx/cp2k610/samples

Amber 18 + bugfix 12

=> /local/apl/lx/amber18-bf12/samples

Gromacs 2018.7 Intelコンパイラ, CPU版 => /local/apl/lx/gromacs2018.7/samples

GAMESS 2018Sep30

=> /local/apl/lx/gamess2018Sep30/samples

# サンプルの構成(1): 一般的な構成

- **samples** ディレクトリ内ではインプットそのものは原則一つです。
- ただし、それを実行するためのジョブスクリプト (**sample\*\*\*\*.sh, sample\*\*\*\*.csh**) は大抵の場合複数あります。違うシェルを使っている場合、GPUを使っている場合等色々です。
  - 例1: sample.sh => /bin/sh を使うサンプル
  - 例2: sample.csh => /bin/csh を使うサンプル
  - 例3: sample-gpu.sh => GPU を使ったサンプル
- 必要に応じて中身を確認、比較してください。(less, diff 等のコマンドをご利用ください)



# サンプルの構成(2): gromacs 2018.7の場合

- 例えば、gromacs2018.7(Intel コンパイラ、CPU版)の場合は以下のようなジョブスクリプトがあります。  
(\* .sh と \*.csh は利用するシェルが違うだけなので省略します。)
  - sample-mpi.sh : Intel MPI版のサンプル
  - sample-threadmpi.sh : thread MPI を使うバージョン
  - \*-module.sh : 環境設定を module コマンドで行うバージョン
- サンプルスクリプトをテンプレートとして使うならば、実際の用途に合わせて選ぶと効率的です。
  - 例1: 多ノード実行を計画しているので thread-MPI ではなく MPI 版を選ぶ
  - 例2: bash と module コマンドに慣れているので sample-mpi-module.sh を選ぶ

Gromacs2018.7 サンプルディレクトリの中身:

```
[user@ccfep5]$ ls /local/ap1/1x/gromacs2018.7/samples
conf.gro          sample-mpi-module.sh*  sample-threadmpi-module.csh*  sample-threadmpi.sh*
grompp.mdp        sample-mpi.csh*        sample-threadmpi-module.sh*    topol.top
sample-mpi-module.csh*  sample-mpi.sh*        sample-threadmpi.csh*
```

## サンプルの構成(3): GAMESS 2018Sep30の場合

- 一方、GAMESS2018Sep30の場合はsample.cshの一つしかありません。

```
[user@ccfep5]$ ls /local/ap1/1x/games2018Sep30/samples  
exam01.inp  sample.csh*  
[user@ccfep5]$
```

アプリケーションの種類やバージョンによって用意されているサンプル数は大きく異なります。

# サンプルの実行: 一般的な手順

1. サンプルを実行するには、まずサンプルディレクトリの中身(もしくは sample ディレクトリそのもの)を自身のディレクトリ以下にコピーしてください。代表的な利用可能領域は以下の通りです
  - ホームディレクトリ /home/users/(ユーザID)
  - 保存用ディレクトリ /save/users/(ユーザID)
2. サンプルをコピーしたディレクトリで `jsub -q PN sample.sh` のように `jsub` コマンドでサンプルジョブを投入します。手順としては以上です。(ジョブスクリプトの名前は適宜変更してください)

(なお、大抵のスクリプトは `sh ./sample.sh` のような形でフロントエンドノード上でそのまま実行できるようになっています。ただし、CPU数の指定が `jsub` 時と変わってしまう場合があります。また、GPUを利用するジョブは `ccfep` では実行できません。)

# サンプルの実行: Gromacs2018.7 実例(1)

Gromacs2018.7 のサンプルを~(ホームディレクトリ)以下で投入します。  
今回サンプルは `~/gromacs2018.7_sample` ディレクトリに置きます。  
(~ (チルダ)は /home/users/(ユーザ名)に置換されます)

```
[user@ccfep5]$ cp -r /local/ap1/1x/gromacs2018.7/samples ~/gromacs2018.7_sample
[user@ccfep5]$ ls ~/gromacs2018.7_sample
conf.gro          sample-mpi-module.sh*  sample-threadmpi-module.csh*  sample-threadmpi.sh*
grompp.mdp        sample-mpi.csh*        sample-threadmpi-module.sh*    topol.top
sample-mpi-module.csh*  sample-mpi.sh*        sample-threadmpi.csh*
```

このディレクトリに移動(cd)し、sample-mpi.sh ジョブを投入してみます。  
(sample-mpi.sh の中身は次ページで確認できます。)

```
[user@ccfep5]$ cd ~/gromacs2018.7_sample
[user@ccfep5]$ ls
conf.gro          sample-mpi-module.sh*  sample-threadmpi-module.csh*  sample-threadmpi.sh*
grompp.mdp        sample-mpi.csh*        sample-threadmpi-module.sh*    topol.top
sample-mpi-module.csh*  sample-mpi.sh*        sample-threadmpi.csh*
```

```
[user@ccfep5]$ jsub -q PN sample-mpi.sh
4684922.cccms1
[user@ccfep5]$
```

混雑していなければ速やかに終了し、以下のように出力が得られます

```
[user@ccfep5]$ ls ~/gromacs2018.7_sample
conf.gro      md.log          sample-mpi.csh*          sample-threadmpi-module.sh*  topol.tpr
confout.gro  mdout.mdp       sample-mpi.sh*          sample-threadmpi.csh*       traj.trr
energ.edr    mdrun.out       sample-mpi.sh.e4684922  sample-threadmpi.sh*
grompp.mdp   sample-mpi-module.csh*  sample-mpi.sh.o4684922  state.cpt
grompp.out   sample-mpi-module.sh*  sample-threadmpi-module.csh*  topol.top
[user@ccfep5]$
```

# サンプルの実行: sample-mpi.sh

```
#!/bin/sh
#PBS -l select=1:ncpus=6:mpiprocs=6:ompthreads=1:jobtype=core <= 1ノード中の6コア
#PBS -l walltime=00:30:00 <= 制限時間 30分 (MPI:6, OpenMP:1)

if [ ! -z "${PBS_O_WORKDIR}" ]; then
  cd "${PBS_O_WORKDIR}" <= jsub を実行したディレクトリへ移動。
fi <= フロントエンドで直接実行した場合は無視されます。

./local/apl/lx/gromacs2018.7/bin/GMXRC <= gromacs 設定読み込み

#####

N_MPI=6 <= MPIプロセス数の指定。上のmpiprocsの値に合わせる
N_OMP=1 <= OpenMPスレッド数の指定。上のompthreadsの値に合わせる

gmx_d grompp -f grompp.mdp >& grompp.out
mpirun -n ${N_MPI} gmx_mpi mdrun -ntomp ${N_OMP} -s topol >& mdrun.out
```

# サンプルの実行: GAMESS 2018Sep30 実例

GAMESS 2018Sep30 のサンプルをキューに投入してみます。  
サンプルは `~/gameess2018Sep30_sample` ディレクトリに置きます。

```
[user@ccfep5]$ cp -r /local/ap1/1x/gameess2018Sep30/samples ~/gameess2018Sep30_sample
[user@ccfep5]$ ls ~/gameess2018Sep30_sample
exam01.inp  sample.csh*
[user@ccfep5]$
```

sample.csh ジョブを投入してみます。

```
[user@ccfep5]$ cd ~/gameess2018Sep30_sample
[user@ccfep5]$ jsub -q PN sample.csh
4685953.cccms1
[user@ccfep5]$
```

混雑していなければ速やかに終了し、以下のように出力が得られます

```
[user@ccfep5]$ ls ~/gameess2018Sep30_sample
exam01.dat  exam01.log  sample.csh*  sample.csh.o4685953
exam01.inp  nodefile-4685953.cccms1  sample.csh.e4685953
[user@ccfep5]$
```

# サンプルの実行: sample.csh

```
#!/bin/csh -f
#PBS -l select=1:ncpus=4:mpiprocs=4:ompthreads=1:jobtype=core <= 1ノード中の4コア
#PBS -l walltime=24:00:00 <= 制限時間 24 時間 (GAMESS では MPI を使っていないが、MPIとして指定)
#
# Gamess is compiled with sockets and OpenMP enabled.
#
if ($?PBS_O_WORKDIR) then
cd ${PBS_O_WORKDIR} <= jsub を実行したディレクトリへ移動。
endif
                                フロントエンドで直接実行した場合は無視されます。

set gamess = gamess2018Sep30
set RUNGMS = /local/apl/lx/${gamess}/rungms <= GAMESS 設定読み込み
set INPUT = exam01.inp

if ($?PBS_O_WORKDIR) then
set nproc="nodefile-${PBS_JOBID}" <= jsub時のコア数指定(OMP_NUM_THREADS は自動的に
uniq -c ${PBS_NODEFILE} | sed 's/^ *%([0-9]*%) *%(.*)$/%2 %1/' > $nproc ompthreads の値に設定されます)
else
set nproc=4 <= 直接実行時のコア数指定
setenv OMP_NUM_THREADS 1
endif
${RUNGMS} ${INPUT:r} 00 $nproc >& ${INPUT:r}.log
```

# ヒント: select 文

基本的な考え方(厳密ではありません)

GPU使わない  
場合は外す

```
#PBS -l select=N:ncpus=C:mpiprocs=M:ompthreads=T:jobtype=J:ngpus=G
```

ノード数指定

ノードあたりのリソース指定部分  
(実際に使用する総コア数は  $N * C$  です)

N: ノード数

C: ノードあたりのコア数

M: ノードあたりのMPIプロセス数

(machine list がこの指定に従って作られます)

T: ノードあたりのOpenMPスレッド数

(OMP\_NUM\_THREADS環境変数はこの指定に従って設定されます)

J: ジョブタイプ(small, large, core, gpu, gpup, gpuv のいずれか)

G: ノードあたりのGPU数(GPU利用時のみ)

なお、CUDA\_VISIBLE\_DEVICES 環境変数を手動で設定する必要はありません。  
(ngpus=1 ならば 0 番のデバイス、ngpus=2 ならば 0,1 のデバイスを利用できる  
ようになっています。)

<https://ccportal.ims.ac.jp/node/2376> で実際の例を確認することもできます。