サンプルジョブの実行方法

最終更新: 2021/8/4

はじめに

RCCS では共通領域に導入したアプリケーションについて、サンプルジョブスクリプトを用意しています。
RCCS で用意したものをそのまま使う場合はもちろん、自身のディレクトリに導入した別バージョンを使う際のテンプレートとしても利用できます。
以下では Gromacs と GAMESS を例にして、サンプルをどのように実行するのかを説明します。
なお、Gaussian については、専用の g16sub/g09sub を用いた方法をまずはご覧ください。

導入済みアプリケーションのリスト

いくつかの方法があります。

1. ウェブページ上の情報を参照する

パッケージプログラム一覧のページに掲載されている情報で確認できます。

2. module avail コマンドの出力を見る

以下のように出力されます。通常、ジョブとして投入するようなアプリについては以下で赤で示した apl_ex 内にあります。

```
[user@ccfep4 ~]$ module avail

------------------------------ /local/apl/lx/modules/suite ------------------------------
intel_parallelstudio/2015update1        intel_parallelstudio/2020update2
intel_parallelstudio/2017update4        scl/devtoolset-3
(中略)

------------------------------ /local/apl/lx/modules/comp -------------------------------
cuda/10.1            intel/15.0.1         intel/19.0.1         pgi/16.5
cuda/11.1            intel/17.0.4         intel/19.0.5         pgi/17.5
cuda/7.5             intel/17.0.8         intel/19.1.2         pgi/18.1(default)
cuda/8.0             intel/18.0.2         julia/1.3.1          pgi/20.4
cuda/9.1(default)    intel/18.0.5(default) julia/1.5.3

------------------------------- /local/apl/lx/modules/apl --------------------------------
mpi/intelmpi/2017.3.196   mpi/openmpi/2.1.3/intel19 mpi/openmpi/4.0.0/gnu8.3
mpi/intelmpi/2017.4.262   mpi/openmpi/3.1.0/gnu4.8  mpi/openmpi/4.0.0/intel15
(中略)

------------------------------ /local/apl/lx/modules/apl_ex ------------------------------
GRRM/11-g09                  gromacs/2018.7/gnu
GRRM/14-g09(default)         gromacs/2018.7/gnu-CUDA
GRRM/17-g09                  gromacs/2018.7/intel
GRRM/17-g16                  gromacs/2018.7/intel-CUDA
abinit/7.8.2                 gromacs/2018.8/gnu
abinit/8.8.3(default)        gromacs/2018.8/gnu-CUDA
amber/16/bugfix10            gromacs/2018.8/intel
amber/16/bugfix15            gromacs/2018.8/intel-CUDA
amber/18/bugfix1             gromacs/2019.2/gnu
amber/18/bugfix11-volta      gromacs/2019.2/gnu-CUDA
(中略)
gromacs/2018.3/gnu-CUDA      turbomole/7.4-serial
gromacs/2018.3/intel         turbomole/7.4.1-MPI(default)
gromacs/2018.3/intel-CUDA    turbomole/7.4.1-SMP
gromacs/2018.6/gnu           turbomole/7.4.1-serial
gromacs/2018.6/gnu-CUDA      turbomole/7.5-MPI
gromacs/2018.6/intel         turbomole/7.5-SMP
gromacs/2018.6/intel-CUDA    turbomole/7.5-serial

---------------------------- /local/apl/lx/modules/apl_viewer ----------------------------
luscus/0.8.6 molden/5.7   nboview/2.0  vmd/1.9.3

---------------------------- /local/apl/lx/modules/apl_util ----------------------------
allinea/7.1          cmake/3.16.3
cmake/2.8.12.2(default) cmake/3.8.2

------------------------------ /local/apl/lx/modules/lib -------------------------------
boost/1.53.0(default)      mkl/2017.0.4              mkl/2020.0.2
boost/1.59.0               mkl/2018.0.2              nccl/2.3.7-1+cuda9.1(default)
boost/1.70.0               mkl/2018.0.4(default)     spglib/1.11.1(default)
mkl/11.2.1                 mkl/2019.0.1
mkl/2017.0.3               mkl/2019.0.5

------------------------------ /local/apl/lx/modules/misc -------------------------------
inteldev intellic pgilic
[user@ccfep4 ~]$
```

3. /local/apl/lx を参照する

OS 非標準のアプリやライブラリについては、/local/apl/lx に導入されています。そちらで直接確認することもできます。

```
[user@ccfep4 ~]$ ls /local/apl/lx
GRRM@                  gromacs2016.5-gnu/              nbo6018mod/
GRRM11/                gromacs2016.5-gnu-CUDA@         nbo70@
GRRM14/                gromacs2016.5-gnu-CUDA8/        nbo702/
GRRM17/                gromacs2016.6/                  nbo702-i4/
abinit@                gromacs2016.6-CUDA/             nbo707/
abinit782/             gromacs2016.6-gnu/              nbo707-i4/
abinit883/             gromacs2016.6-gnu-CUDA/         nbopro7/
(中略)
gromacs2016.4/         namd213/                        wine30/
gromacs2016.4-CUDA/    namd213-CUDA/                   wine30-win64/
gromacs2016.5/         nbo60@                          wine40-win64/
gromacs2016.5-CUDA@    nbo6015/
gromacs2016.5-CUDA9/   nbo6018/
[user@ccfep4 ~]$
```

(一部はシンボリックリンクになっています。)

サンプルファイルの場所

module の場合は module help (パッケージ名) を実行すると、出力にサンプルのディレクトリ名があります。
あるいは /local/apl/lx 以下でパッケージのディレクトリ(通常は(パッケージ名)(バージョン名)という名前になっています)を
探し、その直下にある samples/ ディレクトリに行っても大丈夫です。

例1: module help で GAMESS 2019Sep30 のサンプルディレクトリを探す

```
[user@ccfep4 ~]$ module help gamess/2019Sep30
(省略)
  Doc(local): /local/apl/lx/gamess2019Sep30/INPUT.DOC

SAMPLES:
  /local/apl/lx/gamess2019Sep30/samples

INFO:
(省略)
[user@ccfep4 ~]$ cd /local/apl/lx/gamess2019Sep30/samples
[user@ccfep4 samples]$ ls
exam01.inp  sample.csh
[user@ccfep4 samples]$
```

例2: gromacs 2019.6 のサンプルを /local/apl/lx から探す

```
[user@ccfep4 ~]$ cd /local/apl/lx
[user@ccfep4 lx]$ ls -d gromacs*
gromacs                gromacs2018.3-CUDA      gromacs2019.4-gnu-CUDA
gromacs2016            gromacs2018.3-gnu       gromacs2019.6
gromacs2016.1          gromacs2018.3-gnu-CUDA  gromacs2019.6-CUDA
gromacs2016.1-CUDA     gromacs2018.6           gromacs2019.6-gnu
gromacs2016.3          gromacs2018.6-CUDA      gromacs2019.6-gnu-CUDA
gromacs2016.3-CUDA     gromacs2018.6-gnu       gromacs2020.2
(省略)
[user@ccfep4 lx]$ cd gromacs2019.6/samples
[user@ccfep4 samples]$ 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@ccfep4 samples]$
```

(-CUDA がついているものは GPU 版です。-gnu がついているものは GCC でビルド、ついていないものは intel compiler でビルドしています。)

サンプルディレクトリ内のファイルについて

サンプルディレクトリ内で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については複数ある場合があります。
例えば、シェルが違う場合、並列方式が違う場合、設定の読み込み方法が違う場合、GPU利用の有無などが挙げられます。

- 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 => /bin/sh を使い、GPU も利用するサンプル

必要に応じて中身を比較するなどしてください。

例: gamess 2019Sep30 の場合

既に上で示したように、実行スクリプトは sample.csh の一つだけです。

```
[user@ccfep4 samples]$ ls
exam01.inp  sample.csh
```

例: gromacs 2019.6 の場合

こちらは実行スクリプトがたくさんあります。が、インプットファイルは 1 セットだけです。

```
[user@ccfep4 samples]$ 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
```

- -mpi がついているものは Intel MPI を使った並列版です(マルチノード対応)
- -threadmpi がついているものは組み込みの thread MPI を使った並列版です(マルチノード非対応)
- -module がついているものは module コマンドで環境設定を行います

サンプルの実行: 一般的な手順

- まず、サンプルのファイルを自分の書き込める領域にコピーします。
- サンプルをコピーしたディレクトリで jsub -q PN sample.sh のように実行します。
- (大抵の場合は、sh ./sample.sh のような形でログインノードで実行可能にもなっています。)
  - CPU 数の指定が jsub 時と変わる場合があります
  - GPU を使うものについては ccfep 上では実行できません。(ccgpup, ccgpuv にて実行ください)

例1: gamess 2019Sep30 の場合

サンプルを ~/gamess2019Sep30_test で実行する場合です。

```
[user@ccfep4 ~]$ mkdir -p ~/gamess2019Sep30_test
[user@ccfep4 ~]$ cd ~/gamess2019Sep30_test
[user@ccfep4 gamess2019Sep30_test]$ cp /local/apl/lx/gamess2019Sep30/samples/* .
[user@ccfep4 gamess2019Sep30_test]$ ls
exam01.inp  sample.csh
[user@ccfep4 gamess2019Sep30_test]$ jsub -q PN sample.csh
4685953.cccms1
```

実行中のジョブの状況は jobinfo -c コマンドで確認できます。

```
[user@ccfep4 gamess2019Sep30_test]$ jobinfo -c
--------------------------------------------------------------------------------
Queue   Job ID Name          Status CPUs User/Grp      Elaps Node/(Reason)
--------------------------------------------------------------------------------
PN      4685953 sample.csh    Run      4  ***/---         -- cccc123
--------------------------------------------------------------------------------
```

混雑していなければ、しばらく待てばジョブが終了し、出力結果が得られます。

```
[user@ccfep4 gamess2019Sep30_test]$ ls ~/gamess2019Sep30_test
exam01.dat   exam01.log              sample.csh*         sample.csh.o4685953
exam01.inp   nodefile-4685953.cccms1  sample.csh.e4685953
[user@ccfep4 gamess2019Sep30_test]$
```

参考: sample.csh の中身について(青文字は説明文で、本来のファイルに含まれない部分です)

```
#!/bin/csh -f
#PBS -l select=1:ncpus=4:mpiprocs=4:ompthreads=1:jobtype=core # <= 1 ノード, 4 コア
#PBS -l walltime=24:00:00 # <= 制限時間は 24 時間
#
#  Gamess is compiled with sockets and OpenMP enabled.
#
if ($?PBS_O_WORKDIR) then
cd ${PBS_O_WORKDIR} # <= jsub で投入する場合はこの文を入れて投入したディレクトリに移動する必要有
endif

set gamess = gamess2019Sep30
set RUNGMS = /local/apl/lx/${gamess}/rungms # <= GAMESS の設定とインプットファイルの指定
set INPUT = exam01.inp

if ($?PBS_O_WORKDIR) then
set nproc="nodefile-${PBS_JOBID}" # <= jsub 時のノードリスト指定
uniq -c ${PBS_NODEFILE} | sed 's/^ *\([0-9]*\) *\(.*\)$/\2 \1/' > $nproc
else
set nproc=4 # <= 直接実行時のコア数指定(4コア)
setenv OMP_NUM_THREADS 1
endif
${RUNGMS} ${INPUT:r} 00 $nproc >& ${INPUT:r}.log
```

この gamess は socket 並列でビルドしていて MPI を利用していませんが、ノードリスト(PBS_NODEFILE)を利用するために MPI 並列として(mpiprocs=4)実行しています。

例2: gromacs 2019.6 の場合

サンプルを ~/gromacs2019.6_test で実行する場合です。

```
[user@ccfep4 ~]$ mkdir -p ~/gromacs2019.6_test
[user@ccfep4 ~]$ cd ~/gromacs2019.6_test
[user@ccfep4 gromacs2019.6_test]$ cp /local/apl/lx/gromacs2019.6/samples/* .
[user@ccfep4 gromacs2019.6_test]$ 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@ccfep4 gromacs2019.6_test]$ jsub -q PN sample-mpi.sh
4684922.cccms1
```

実行中のジョブの状況は jobinfo -c コマンドで確認できます。

```
[user@ccfep4 gromacs2019.6_test]$ jobinfo -c
--------------------------------------------------------------------------------
Queue   Job ID Name          Status CPUs User/Grp      Elaps Node/(Reason)
--------------------------------------------------------------------------------
PN      4684922 sample-mpi.sh  Run      6  ***/---         -- cccc123
--------------------------------------------------------------------------------
```

混雑していなければ、しばらく待てばジョブが終了し、出力結果が得られます。

```
[user@ccfep4 gromacs2019.6_test]$ ls ~/gromacs2019.6_test
conf.gro     mdout.mdp                    sample-mpi.csh              state.cpt
confout.gro  mdrun.out                    sample-mpi.sh               topol.top
ener.edr     sample-mpi-module.csh        sample-threadmpi-module.csh  topol.tpr
grompp.mdp   sample-mpi-module.csh.e4684922  sample-threadmpi-module.sh   traj.trr
grompp.out   sample-mpi-module.csh.o4684922  sample-threadmpi.csh
md.log       sample-mpi-module.sh         sample-threadmpi.sh
[user@ccfep4 gromacs2019.6_test]$
```

参考: sample-mpi.sh の中身(青文字は説明文で、本来のファイルに含まれない部分です)

```
#!/bin/sh
#PBS -l select=1:ncpus=6:mpiprocs=6:ompthreads=1:jobtype=core # <= 6 MPI * 1 OMP
#PBS -l walltime=00:30:00 # <= 制限時間 30 分

if [ ! -z "${PBS_O_WORKDIR}" ]; then
  cd "${PBS_O_WORKDIR}" # <= jsub で投入時は jsub を実行したディレクトリに移動します
fi

. /local/apl/lx/gromacs2019.6/bin/GMXRC # <= gromacs 環境設定読み込み

###############################################################################

N_MPI=6 # <= MPI 並列数; 上記の mpiprocs に合わせる
N_OMP=1 # <= OpenMP 並列数; 上記の ompthreads 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
```

ジョブスクリプトの書き方に関する参考情報

https://ccportal.ims.ac.jp/node/2376 に select 文のサンプルや簡単な説明があります。