

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)

更新履歷

- 2019/9/6 初稿作成

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 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。)

```
@ccfep4 ~]$ module avail

----- /local/apl/lx/modules/suite -----
intel_parallelstudio/2015update1          intel_parallelstudio/2018update2          scl/devtoolset-3
intel_parallelstudio/2017update4          intel_parallelstudio/2018update4(default) scl/devtoolset-4
intel_parallelstudio/2017update8          intel_parallelstudio/2019update1          scl/devtoolset-6

----- /local/apl/lx/modules/comp -----
cuda/7.5          cuda/9.1(default)          intel/17.0.4          intel/18.0.2          intel/19.0.1          psi/17.5
cuda/8.0          intel/15.0.1          intel/17.0.8          intel/18.0.5(default)          intel/16.5          psi/18.1(default)

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

----- /local/apl/lx/modules/apl_ex -----
GRRM/11-g09          espresso/6.1          gromacs/2016.3/intel-CUDA          gromacs/2018.6/gnu-CUDA          lammps/16Mar18/gnu-CUDA          reactionplus/1.0
GRRM/14-g09(default)          espresso/6.3(default)          gromacs/2016.4/intel          gromacs/2018.6/intel          lammps/16Mar18/intel          siesta/3.1
abinit/7.8.2          gamess/2018Feb14          gromacs/2016.4/intel-CUDA          gromacs/2018.6/intel-CUDA          lammps/16Mar18/intel-CUDA          siesta/4.0.2(default)
abinit/8.8.3(default)          gamess/2018Sep30(default)          gromacs/2016.5/gnu          gromacs/2018.7/gnu          lammps/22Aug18/intel          smash/2.2.0
amber/16/bugfix10          gaussian/g09/b01          gromacs/2016.5/gnu-CUDA          gromacs/2018.7/gnu-CUDA          lammps/22Aug18/intel-CUDA          turbomole/7.2.1-MPI
amber/16/bugfix15          gaussian/g09/c01          gromacs/2016.5/intel          gromacs/2018.7/intel          lammps/22Aug18/intel-CUDA-volta          turbomole/7.2.1-SMP
amber/18/bugfix1          gaussian/g09/e01          gromacs/2016.5/intel-CUDA          gromacs/2018.7/intel-CUDA          molpro/8.2          turbomole/7.2.1-serial
amber/18/bugfix11-volta          gaussian/g09/e01          gromacs/2016.6/gnu          gromacs/2019.2/gnu          turbomole/7.3-MPI(default)
amber/18/bugfix12          gaussian/g16/b03          gromacs/2016.6/gnu-CUDA          gromacs/2019.2/gnu-CUDA          namd/2.11          turbomole/7.3-SMP
autodock/4.2.6          gaussian/g16/b01          gromacs/2016.6/intel          gromacs/2019.2/intel          namd/2.11-CUDA          turbomole/7.3-serial
cp2k/6.1.0/gnu          gaussian/g16/c01          gromacs/2016.6/intel-CUDA          gromacs/2019.2/intel-CUDA          namd/2.13(default)          turbomole/7.4-MPI
cp2k/6.1.0/gnu-CUDA          gaussian/g16/c01          gromacs/2018.1/gnu          gromacs/2019.2/intel-CUDA          namd/2.13-CUDA          turbomole/7.4-SMP
cp2k/6.1.0/intel          genesis/1.1.6          gromacs/2018.1/intel          gromacs/5.1.4/intel          ntchem/2013-5.0-mpi          turbomole/7.4-serial
cp2k/6.1.0/intel-CUDA          genesis/1.3.0          gromacs/2018.3/gnu          gromacs/5.1.4/intel-CUDA          ntchem/2013-5.0-mpiomp          ntchem/2013-5.0-serial
crystal/14-104          genesis/1.3.0-CUDA          gromacs/2018.3/gnu-CUDA          gromacs/5.1.5/gnu          ntchem/2013-5.0-serial          nwhem/6.8
diac/18.0          gromacs/2016.1/intel          gromacs/2018.3/intel          gromacs/5.1.5/gnu-CUDA          openmolcas/20190604          psi4/1.1
espresso/5.1.2          gromacs/2016.1/intel-CUDA          gromacs/2018.3/intel-CUDA          lammps/16Mar18/gnu          psi4/1.1
espresso/5.4          gromacs/2016.3/intel          gromacs/2018.6/gnu

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

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

----- /local/apl/lx/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)          nocl/2.3.7-1+cuda9.1(default)

----- /local/apl/lx/modules/misc -----
inteldev          intellc          pgilic
```

apl_ex

(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。

```
ccfep4 ~] $ ls /local/apl/lx/
GRRM@ crystal14-104.bad/ genesis116-CUDA/ gromacs2018.7/ lammps22Aug18/ nwchem@ openmpi400-intel17/
GRRM11/ cuda/ genesis130/ gromacs2018.7-CUDA/ luscus086/ openmpi400-intel18@
GRRM14/ cuda-10.1/ genesis130-CUDA/ gromacs2018.7-gnu/ modules/ openmm41/ openmpi400-intel19/
abinit@ cuda-7.5/ gromacs@ gromacs2018.7-gnu-CUDA/ modules.2017/ openmolcas20190604/ orca401/
abinit782/ cuda-8.0/ gromacs2016@ gromacs2019/ modules.2018/ openmpi213/ pbs14/
abinit883/ cuda-9.1/ gromacs2016.1/ gromacs2019.2/ molcas@ openmpi213-gnu4.8@ pgi@
allinea@ dirac180/ gromacs2016.1-CUDA/ gromacs2019.2-CUDA/ molcas82/ openmpi213-gnu4.9/ psi16.5/
allinea71/ dirac180-lm/ gromacs2016.3/ gromacs2019.2-gnu/ molder@ openmpi213-gnu5.3/ pgi17.5/
amber@ espresso/ gromacs2016.3-CUDA/ gromacs2019.2-gnu-CUDA/ molder57/ openmpi213-gnu6.3/ pgi18.1/
amber12@ espresso512/ gromacs2016.3-CUDA.bad/ gromacs455/ molpro@ openmpi213-gnu7.3/ ps_walltime*
amber12-bf21/ espresso54/ gromacs2016.4/ gromacs514/ molpro2012.1-pl37/ openmpi213-intel/ ps_walltime2*
amber14@ espresso61/ gromacs2016.4-CUDA/ gromacs514/ molpro2015.1-pl19/ openmpi213-intel15/ psi4@
amber14-bf11/ espresso63/ gromacs2016.5/ gromacs515/ molpro2015.1-pl27/ openmpi213-intel17/ psi4-11/
amber14-bf11.old/ g09@ gromacs2016.5-CUDA@ gromacs515-CUDA@ molpro2015.1-pl33/ openmpi213-intel18@ reactionplus@
amber16@ g09b01/ gromacs2016.5-CUDA9/ gromacs515-CUDA8/ molpro2015.1-pl33-i1708/ openmpi213-intel19/ reactionplus10/
amber16-bf10/ g09c01/ gromacs2016.5-gnu/ gromacs515-gnu/ molpro2018.2/ openmpi310/ siesta@
amber16-bf15/ g09d01/ gromacs2016.5-gnu-CUDA@ gromacs515-gnu-CUDA@ molpro2019.1.2/ openmpi310-gnu4.8@ siesta31/
amber18@ g09d01-test/ gromacs2016.5-gnu-CUDA8/ gromacs515-gnu-CUDA8/ namd@ openmpi310-gnu4.9/ siesta402/
amber18-bf1/ g09e01/ gromacs2016.5-gnu-CUDA9/ intel@ namd211/ openmpi310-gnu5.3/ smash@
amber18-bf11v/ g16@ intel2015update1/ namd211-CUDA/ openmpi310-gnu6.3/ smash220/
amber18-bf12/ g16a03/ intel2017update4/ namd213/ openmpi310-gnu7.3/ spglib-1.11.1/
anaconda2-2019Jul/ g16b01/ intel2017update8/ namd213-CUDA/ openmpi310-intel/ turbomole@
anaconda3-2019Jul/ g16c01/ intel2018update2/ nbo60/ openmpi310-intel15/ turbomole72/
autodock4@ ga-5-5/ intel2018update4/ nbo6015/ openmpi310-intel17/ turbomole721/
autodock426/ gamess@ intel2019update1/ nbo6018/ openmpi310-intel18@ turbomole73/
boost-1.59.0/ gamess2017Apr20/ jobstatistic* nbo6018mod/ openmpi310-intel19/ turbomole74/
boost-1.70.0/ gamess2017Nov11/ lammps@ nbo70@ openmpi400/ vmd@
cmake3.8.2/ gamess2018Feb14/ lammps16Mar18/ nbd702/ openmpi400-gnu4.8@ vmd193/
cp2k@ gamess2018Feb14-t1/ lammps16Mar18-CUDA@ nbo702-i4/ openmpi400-gnu4.9/ watch_memory*
cp2k610/ gamess2018Feb14-t2/ lammps16Mar18-CUDA8/ nbopro7/ openmpi400-gnu5.3/ wine30/
cp2k610-gnu/ gamess2018Sep30/ lammps16Mar18-gnu/ nboview2/ openmpi400-gnu6.3/ wine30-win64/
crystal14@ sdv113/ gromacs2018.6-CUDA/ lammps16Mar18-gnu-CUDA@ ncc1237-1tcuda91/ openmpi400-gnu7.3/ wine40-win64/
crystal14-104@ gromacs2018.6-gnu-CUDA/ lammps16Mar18-intel17/ ncc1237-1tcuda91/ openmpi400-intel17/
crystal14-104@ gromacs2018.6-gnu-CUDA/ lammps16Mar18-intel17/ ncc1237-1tcuda91/ openmpi400-intel17/
```

(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 `gamess(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)

```
@ccfep4 ~]$ ls /local/apl/lx/cp2k610/samples/
H2O-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
@ccfep4 ~]$ 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*
@ccfep4 ~]$ 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*
@ccfep4 ~]$ ls /local/apl/lx/games2018Sep30/samples/
exam01.inp  sample.csh*
```

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/games2018Sep30/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:

```
@ccfep4 ~]$ ls /local/apl/ix/gromacs2018.7/samples
cont.gro          sample-threadmpi-module.csh*
grompp.mdp        sample-threadmpi-module.sh*
sample-mpi-module.csh*  sample-threadmpi.csh*
sample-mpi-module.sh*  sample-threadmpi.sh*
sample-mpi.csh*      topol.top
sample-mpi.sh*
```

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。

```
[~@ccfep4 ~]$ ls /local/apl/lx/game2018Sep30/samples/  
exam01.inp  sample.csh*
```

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)

```
@ccfep4 ~]$ cp -r /local/apl/lx/gromacs2018.7/samples/ ~/gromacs2018.7_sample
@ccfep4 ~]$ 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。)

```
@ccfep4 ~]$ cd ~/gromacs2018.7_sample/
@ccfep4 gromacs2018.7_sample]$ 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*
@ccfep4 gromacs2018.7_sample]$ jsub -q PN sample-mpi.sh
4684922.cccms1
@ccfep4 gromacs2018.7_sample]$
```

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

```
@ccfep4 gromacs2018.7_sample]$ 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
ener.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
@ccfep4 gromacs2018.7_sample]$
```

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。

```
@ccfep3 ~]$ cp -r /local/apl/lx/gameess2018Sep30/samples ~/gameess2018Sep30_sample
@ccfep3 ~]$ ls ~/gameess2018Sep30_sample
exam01.inp  sample.csh*
```

sample.csh ジョブを投入してみます。

```
@ccfep3 ~]$ cd ~/gameess2018Sep30_sample/
@ccfep3 gameess2018Sep30_sample]$ jsub -q PN sample.csh
4685953.cccms1
```

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

```
@ccfep3 gameess2018Sep30_sample]$ ls
exam01.dat  exam01.log  sample.csh*  sample.csh.o4685953
exam01.inp  nodefile-4685953.cccms1  sample.csh.e4685953
@ccfep3 gameess2018Sep30_sample]$
```

サンプルの実行: sample.csh

```
#!/bin/csh -f
#PBS -l select=1:ncpus=4:mpiprocs=4:ompthreads=1:jobtype=core
#PBS -l walltime=24:00:00
#
# Gamess is compiled with sockets and OpenMP enabled.
#
if ($?PBS_O_WORKDIR) then
cd ${PBS_O_WORKDIR}
endif

set gamess = gamess2018Sep30
set RUNGMS = /local/apl/lx/${gamess}/rungms
set INPUT = exam01.inp

if ($?PBS_O_WORKDIR) then
set nproc="nodefile-${PBS_JOBID}"
uniq -c ${PBS_NODEFILE} | sed 's/^ *%([0-9]*) *%(.*)$/%2 %1/' > $nproc
else
set nproc=4
setenv OMP_NUM_THREADS 1
endif

${RUNGMS} ${INPUT:r} 00 $nproc >& ${INPUT:r}.log
```

`<= 1ノード中の4コア`
(GAMESSではMPIを使っていないが、MPIとして指定)

`<= 制限時間 24 時間`

`<= jsub を実行したディレクトリへ移動。`
フロントエンドで直接実行した場合は無視されます。

`<= GAMESS 設定読み込み`

`<= jsub時のコア数指定(OMP_NUM_THREADS は自動的に ompthreads の値に設定されます)`

`<= 直接実行時のコア数指定`

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。