Sample Jobs

National Institutes of Natural Sciences
Okazaki Research Facilities
Research Center for Computational Science (RCCS)

Changelog

- Sep. 6, 2019 First version
- Jan. 15, 2020 Use plain text for terminal view
- Aug. 4, 2021 Add jobinfo information

Introduction

The aim of this document is to explain how to run sample jobs of pre-installed applications.

Sample jobs will be a good example when you learn how to run jobs on RCCS. Also, these files can be used as input templates for your jobs.

In this document, we will show some examples using Gromacs and GAMESS samples. But the procedure itself is applicable to other software, since the sample structure is common among.

Table of Contents

Preinstalled Software (1): The List

- Application list is available at https://ccportal.ims.ac.jp/en/installed_applications.
- You can see the list by "module avail" command when you logged in. (Research software can be found at apl_ex category.)

```
[user@ccfep5]$ module avail

--------------------------------------------------------------------------------------------- /local/apl/lx/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/lx/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/lx/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/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/lx/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         siesta/4.0.2(default)
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(default)          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

------------------------------------------------------------------------------------------- /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/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)        nccl/2.3.7-1+cuda9.1(default)

--------------------------------------------------------------------------------------------- /local/apl/lx/modules/misc ----------------------------------------------------------------------------------------------
inteldev intellic pgilic
[user@ccfep5]$
```

apl_ex

(Some of software are not listed here (e.g. Anaconda).)

Preinstalled Software (2): Path

Preinstalled software on RCCS can be found under /local/apl/lx/.
You might guess application names and versions from dir names.

```
[user@ccfep5]$ ls /local/apl/lx/
GRRM@                cuda-10.1/           genesis140-CUDA/     gromacs2018.8-gnu/   lammps22Aug18/       ntchem@              openmpi400-intel/
GRRM11/              cuda-7.5/            gromacs@             gromacs2018.8-gnu-CUDA/ lammps7Aug19/     ntchem2013.5.0/      openmpi400-intel15/
GRRM14/              cuda-8.0/            gromacs2016@         gromacs2019@         lammps7Aug19-CUDA/   nwchem@              openmpi400-intel17/
abinit@              cuda-9.1/            gromacs2016.1/       gromacs2019.2/       luscus086/           nwchem68/            openmpi400-intel18@
abinit782/           dirac180/            gromacs2016.1-CUDA/  gromacs2019.2-CUDA/  modules/             openmm41/            openmpi400-intel19/
abinit883/           dirac180-lm/         gromacs2016.3/       gromacs2019.2-gnu/   modules.2017/        openmolcas20190604/  orca401/
allinea@             espresso@            gromacs2016.3-CUDA/  gromacs2019.2-gnu-CUDA/ modules.2018/     openmpi213/          orca421/
allinea71/           espresso512/         gromacs2016.3-CUDA.bad/ gromacs2019.4/    molcas@              openmpi213-gnu4.8@   pbs14/
amber@               espresso54/          gromacs2016.4/       gromacs2019.4-CUDA/  molcas82/            openmpi213-gnu4.9/   pgi@
amber12@             espresso61/          gromacs2016.4-CUDA/  gromacs2019.4-gnu/   molden@              openmpi213-gnu5.3/   pgi16.5/
amber12-bf21/        espresso63/          gromacs2016.5/       gromacs2019.4-gnu-CUDA/ molden57/         openmpi213-gnu6.3/   pgi17.5/
amber14@             g09@                 gromacs2016.5-CUDA@  gromacs455/          molpro@              openmpi213-gnu7.3/   pgi18.1/
amber14-bf11/        g09b01/              gromacs2016.5-CUDA9/ gromacs514/          molpro2012.1-pl37/   openmpi213-gnu8.3/   ps_walltime*
amber14-bf11.old/    g09c01/              gromacs2016.5-gnu/   gromacs514-CUDA/     molpro2015.1-pl19/   openmpi213-intel/    ps_walltime2*
amber16@             g09d01/              gromacs2016.5-gnu-CUDA@ gromacs515/       molpro2015.1-pl27/   openmpi213-intel15/  psi4@
amber16-bf10/        g09d01-test/         gromacs2016.5-gnu-CUDA8/ gromacs515-CUDA@ molpro2015.1-pl33/   openmpi213-intel17/  psi4-11/
amber16-bf15/        g09e01/              gromacs2016.6/       gromacs515-CUDA8/    molpro2015.1-pl33-i1708/ openmpi213-intel18@ reactionplus@
amber18@             g16@                 gromacs2016.6-CUDA/  gromacs515-gnu/      molpro2018.2/        openmpi213-intel19/  reactionplus10/
amber18-bf1/         g16a03/              gromacs2016.6-gnu/   gromacs515-gnu-CUDA@ molpro2019.1.2/      openmpi310/          siesta@
amber18-bf11v/       g16b01/              gromacs2016.6-gnu-CUDA/ gromacs515-gnu-CUDA8/ molpro2019.2.3/  openmpi310-gnu4.8@   siesta31/
amber18-bf12/        g16c01/              gromacs2018@         intel@               namd@                openmpi310-gnu4.9/   siesta402/
amber18-bf16/        ga-5-5/              gromacs2018.1/       intel2015update1/    namd211/             openmpi310-gnu5.3/   smash@
anaconda2-2019Jul/   gamess@              gromacs2018.1-gnu/   intel2017update4/    namd211-CUDA/        openmpi310-gnu6.3/   smash220/
anaconda3-2019Jul/   gamess2017Apr20/     gromacs2018.3/       intel2017update8/    namd213/             openmpi310-gnu7.3/   spglib-1.11.1/
autodock4@           gamess2017Nov11/     gromacs2018.3-CUDA/  intel2018update2/    namd213-CUDA/        openmpi310-gnu8.3/   turbomole@
autodock426/         gamess2018Feb14/     gromacs2018.3-gnu/   intel2018update4/    nbo60@               openmpi310-intel/    turbomole72/
boost-1.59.0/        gamess2018Feb14-t1/  gromacs2018.3-gnu-CUDA/ intel2019update1/ nbo6015/             openmpi310-intel15/  turbomole721/
boost-1.70.0/        gamess2018Feb14-t2/  gromacs2018.6/       intel2019update5/    nbo6018/             openmpi310-intel17/  turbomole73/
cmake3.8.2/          gamess2018Sep30/     gromacs2018.6-CUDA/  jobstatistic*        nbo6018mod/          openmpi310-intel18@  turbomole74/
cp2k@                gamess2019Sep30/     gromacs2018.6-gnu/   julia-1.3.1/         nbo70@               openmpi310-intel19/  vmd@
cp2k610/             gdvi13/              gromacs2018.6-gnu-CUDA/ lammps@           nbo702/              openmpi400/          vmd193/
cp2k610-gnu/         genesis115/          gromacs2018.7/       lammps16Mar18/       nbo702-i4/           openmpi400-gnu4.8@   watch_memory*
crystal14@           genesis116/          gromacs2018.7-CUDA/  lammps16Mar18-CUDA@  nbo707/              openmpi400-gnu4.9/   wine30/
crystal14-104@       genesis116-CUDA/     gromacs2018.7-gnu/   lammps16Mar18-CUDA8/ nbo707-i4/           openmpi400-gnu5.3/   wine30-win64/
crystal14-104-t1/    genesis130/          gromacs2018.7-gnu-CUDA/ lammps16Mar18-gnu/ nbopro7/            openmpi400-gnu6.3/   wine40-win64/
crystal14-104.bad/   genesis130-CUDA/     gromacs2018.8/       lammps16Mar18-gnu-CUDA@ nboview2/         openmpi400-gnu7.3/
cuda@                genesis140/          gromacs2018.8-CUDA/  lammps16Mar18-gnu-CUDA8/ nccl237-1+cuda91/ openmpi400-gnu8.3/
[user@ccfep5]$
```

(actual view depends on your terminal appli and its settings.)

In case of Gromacs, dirnames are gromacs(version)-(env). For GAMESS, the names are gamess(versioin/release date). Version-less directories (such as “gromacs” dir) would be a link to their default version.

Location of Sample Job Files

- Sample files are available under samples/ in the application directory.
- The sample path would be available from output of "module help (module name)" command for "apl_ex" category software.

```
[user@ccfep5]$ ls /local/apl/lx/cp2k610/samples
H2O-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
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
[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]$
```

Examples:

CP2K 6.1.0 Intel version => /local/apl/lx/cp2k610/samples

Amber 18 + bugfix 12 => /local/apl/lx/amber18-bf12/samples

Gromacs 2018.7 Intel compiler, CPU ver. => /local/apl/lx/gromacs2018.7/samples

GAMESS 2018Sep30 => /local/apl/lx/gamess2018Sep30/samples

Sample Structure (1): Basics

- Samples directory basically contains only single input set.
- However, there can be multiple of job script files (sample****.sh, sample****.csh). Those samples would be different in terms of shell type, whether GPUs are employed etc, although they would use the same input file.
  - e.g.: sample.sh => sample using /bin/sh
  - e.g.: sample.csh => sample using /bin/csh
  - e.g.: sample-gpu.sh => GPU employed sample
- You might need to check the differences among them by “less” or “diff” scommand.

Sample Structure (2): gromacs 2018.7

- Following samples are available for CPU version of gromacs2018.7 compiled with Intel compiler. (*.csh files are omitted for simplicity)
  - sample-mpi.sh : Intel MPI version
  - sample-threadmpi.sh : thread MPI version
  - *-module.sh : “module” command is used for env setting
- You might want to choose sample according to your research plan.
  - e.g. choose MPI version, since multi node runs will be employed later.
  - e.g. choose sample-mpi-module.sh, since you are familiar with bash and module command.

Contents of Gromacs2018.7 sample directory:

```
[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]$
```

Sample Structure (3): GAMESS 2018Sep30

- On the other hand. GAMESS2018Sep30 contains only single sample script, sample.csh.

```
[user@ccfep5]$ ls /local/apl/lx/gamess2018Sep30/samples
exam01.inp  sample.csh*
[user@ccfep5]$
```

Number of sample scripts strongly depends on the nature of application and their functions.

Run Sample: General Procedure

1. To run sample, you need to copy contents of sample directory (or sample directory itself) to your directory. Suitable places would be one of the following two places.
   - Home directory: /home/users/(user ID)
   - Data directory /save/users/(user ID)

   Work directory is not suitable for this purpose, since contents in that directory might be removed instantly.
2. Once sample files are copied, then you “cd” to that dir and submit a sample job via “jsub –q PN sample.sh” (please modify job script file name if necessary).

   Note: you can run most of the sample scripts at the front-end nodes simply by executing “sh ./sample.sh” or something. Please note that some of samples do not support this type of runs. For example, GPU jobs are not runnable on front-end nodes.

Run Sample: Gromacs2018.7(1)

Let's try to submit Gromacs2018.7 sample at your home directory (~).
In this example, sample is copied to ~/gromacs2018.7_sample.
(~ (tilde) is replaced by /home/users/(username))

```
[user@ccfep5]$ cp -r /local/apl/lx/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*
[user@ccfep5]$
```

After the copy, "cd" to that dir and submit "sample-mpi.sh" with jsub.
(Contents of "sample-mpi.sh" are shown in next page.)

```
[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]$
```

Run Sample: Gromacs2018.7(2)

Status of submitted jobs can be verified with “jobinfo -c” command.

```
[user@ccfep5]$ jobinfo -c
--------------------------------------------------------------------------------
Queue   Job ID Name            Status CPUs User/Grp       Elaps Node/(Reason)
--------------------------------------------------------------------------------
PN      4684922 sample-mpi.sh  Run       6  ***/---          -- cccc123
--------------------------------------------------------------------------------
[user@ccfep5]$
```

(Once the job finished, it will disappear from the list.)

If system is not terribly busy, output files will be available promptly.

```
[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]$
```

Run Sample: sample-mpi.sh

```
#!/bin/sh
#PBS -l select=1:ncpus=6:mpiprocs=6:ompthreads=1:jobtype=core  <=  6 cores from 1 node
#PBS -l walltime=00:30:00  <= 30 minutes time limit                (MPI:6, OpenMP:1)

if [ ! -z "${PBS_O_WORKDIR}" ]; then
  cd "${PBS_O_WORKDIR}"  <= “cd” to directory where you exec “jsub”.
fi                          If script directly executed on front-end node, this line
                            would be ignored.
. /local/apl/lx/gromacs2018.7/bin/GMXRC  <= Load gromacs env.

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

N_MPI=6  <= # of MPI processes. Value should be the same as “mpiprocs” value above.
N_OMP=1 <= # of OpenMP threads. Value should be consistent with “ompthreads”.

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

Run Sample (4): GAMESS 2018Sep30(1)

Let's try to run GAMESS 2018Sep30 sample.
Copy sample directory to ~/gamess2018Sep30_sample.

```
[user@ccfep5]$ cp -r /local/apl/lx/gamess2018Sep30/samples ~/gamess2018Sep30_sample
[user@ccfep5]$ ls ~/gamess2018Sep30_sample
exam01.inp  sample.csh*
[user@ccfep5]$
```

Submit sample job "sample.csh".

```
[user@ccfep5]$ cd ~/gamess2018Sep30_sample
[user@ccfep5]$ jsub -q PN sample.csh
4685953.cccms1
[user@ccfep5]$
```

Run Sample (4): GAMESS 2018Sep30(2)

Status of submitted job can be verified with “jobinfo -c“ command.

```
[user@ccfep5]$ jobinfo -c
-------------------------------------------------------------------------------
Queue   Job ID Name           Status CPUs User/Grp       Elaps Node/(Reason)
-------------------------------------------------------------------------------
PN      4685953 sample.csh     Run       4  ***/---          -- cccc123
-------------------------------------------------------------------------------
[user@ccfep5]$
```

In the sample above, job is running (Status = Run) on node cccc123.
When the job status is “Queue”, the job is waiting for some reason.

If system is not terribly busy, job will finish promptly and output files be generated.

```
[user@ccfep5]$ ls ~/gamess2018Sep30_sample
exam01.dat   exam01.log              sample.csh*          sample.csh.o4685953
exam01.inp   nodefile-4685953.cccms1  sample.csh.e4685953
[user@ccfep5]$
```

Run Sample (5): 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
```

<= 4 cores from 1 node
(specify cores for MPI procs, although this GAMESS is not MPI version.)

<= 24 hours limit

<= “cd” to directory where you exec “jsub”.
If script directly executed on front-end node, this line would be ignored.

<= GAMESS launcher.

<= # of cores specification when submitted via jsub
(OMP_NUM_THREADS is set to ompthreads automatically.)

<= # of cores specification for non-jsub run.

Tips: select line

Brief explanation:

Remove when GPU not employed

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

Node number

Resources "per node"
(actual # of cores is N * C)

N: # of nodes
C: # of cores per node
M: # of MPI processes per node
("machine list" would be generated according to this value)
T: # of OpenMP threads per node
(OMP_NUM_THREADS will be set according to this value)
J: Jobtype (one from small, large, core, gpu, gpup, gpuv)
G: # of GPUs per node (when GU employed)
You usually don't need to set CUDA_VISIBLE_DEVICES manually.
(if ngpus=1 specified, id 0 device will be available in your run,
if ngpus=2 specified, ids 0 and 1 will become available.)

Examples are available at https://ccportal.ims.ac.jp/en/node/2377.