## Submit Jobs (jsub)

(Last update: Dec 3, 2024)

You can submit your jobs with "jsub" command. Input files (and jobscript) should be placed on ccfep beforehand using scp or sftp etc. (For file transfer using scp and sftp, please check quick start guide page.)

For Gaussian jobs, there are special comands g16sub and g09sub.Please consider g16sub/g09sub first, although jsub can be used.



### Basic Usage

You need to prepare a job script (named "job.sh" here; the detail is described below) first. You can submit a jobs using "jsub" command and "job.sh" file as follows. (Don't type leading $.) If the job is successfully accepted by the job server, job ID and server name will be shown.

```
$ jsub job.sh
6169323.ccpbs1
$
```

The "jsub" command itself finishes immediately. However, submitted job is not always executed immediately. If there are no available resources (CPU or GPU), the job needs to wait for a while. Submitted jobs remain in the system even when you log out.

The displayed number (6169323 in the example above) is the ID of the job (job ID). This is used when you delete the job. However, it is not necessary to always remember the displayed job ID. This is because job IDs can be listed with "jobinfo" command after the job submission.

Although you don't need to specify queue name, you can specify the queue name like "jsub -q H job.sh".

Optional parameters of "jsub" command can be listed with "jsub --help" command. Please use"jobinfo" command to check the status of submitted jobs. If you want to delete or cancel jobs, please use"jdel" command.

### Prepare Jobscript

For applications installed by RCCS, there are samples of jobscript and input files. For the standard location of those files in RCCS, please check Sample Jobs page. These sample jobscripts can be used as a template file for your own jobs.

We here show an example jobscript. This request 64 CPU cores, 32 MPI processes and 2 OpenMP threads. The maximum wall time of this job is 24 hours. This will run "my-program" installed in the home directory. "my-program" uses Open MPI 4.1.6 installed by RCCS. (NOTE: annotation part beginning with ⇐ in the example shouldn't be present in the actual job script file. It will cause errors.)

```
#!/bin/sh   ⇐ 1 Choose Interpreter (shebang)
#PBS -l select=1:ncpus=64:mpiprocs=32:ompthreads=2 ⇐ 2 Request Resources
#PBS -l walltime=24:00:00 ⇐ 3 Max Walltime of Job
                ⇐ 4 Optional Parameterscan be added here


# chdir to the working directory when you submit job (recommended; possible to skip)
# When job starts on computation node, the working directory is usually your home directory.
```

```
# If you skip this operation, you may have trouble with input and output file paths for example.
cd ${PBS_O_WORKDIR}

# do module setting if necessary
module -s purge
module -s load openmpi/4.1.6

# you can set parameters for your application here
export PATH="/home/users/${USER}/bin:$PATH"

INPUT=myinp.inp
OUTPUT=myout.out

# run program
mpirun -np 32 my-program -i ${INPUT} -o ${OUTPUT}
```

1 - Choose Interpreter (shebang; required)

2 - Request Resources (select; required)

3 - Max Walltime of Job (walltime; required)

4 - Optional Parameters

Job Dependencies (-W depend=afterok, -W depend=afterany)

Stepwise Execution of Jobs (step jobs; --step or --stepany)

Specify Values of Variables upon Job Submission (-v VARNAME=VALUE)

Wait until Job Termination (-W block=true)

## Special Usage

Need Large Memory But not CPU Cores

Do Many Calculations in Single Job