

Gaussian job submission using g16sub

Last Update: Feb 19, 2021

Introduction

Aim of this page is to show how to run Gaussian jobs with g16sub command on RCCS.

Prerequisites

Following conditions must be satisfied beforehand.

- ▶ You can login to RCCS frontend node (ccfep) with SSH.
- ▶ You can send/receive files to/from RCCS via scp or sftp.
- ▶ Your Gaussian input file. Just to test the procedure, please use sample input below.

Please visit [quickstartguide page](#) if ssh, scp, sftp settings are not yet complete.

sample Gaussian input file

In the following, we use [ch3cl.gjf](#) file (.gjf is one of the common file extensions of Gaussian input file) as an example. We need to send this file to RCCS first, and then run Gaussian.

```
%chk=ch3cl.chk
# HF/6-31G(d,p) Opt
methyl chloride
0,1
C -0.000004 1.127470 0.000000
H -0.511417 1.468491 0.885898
H -0.511417 1.468491 -0.885898
H 1.022922 1.468527 0.000000
Cl -0.000004 -0.657078 0.0000
```

Please note that %mem, %nprocshared, %cpu are usually overwritten by g16sub/g09sub. Number of CPU cores can be specified by -np option of g16sub/g09sub. For the memory amount, you don't need to set it manually. This is because reasonable value will be automatically assigned according to the jobtype and number of CPU cores.

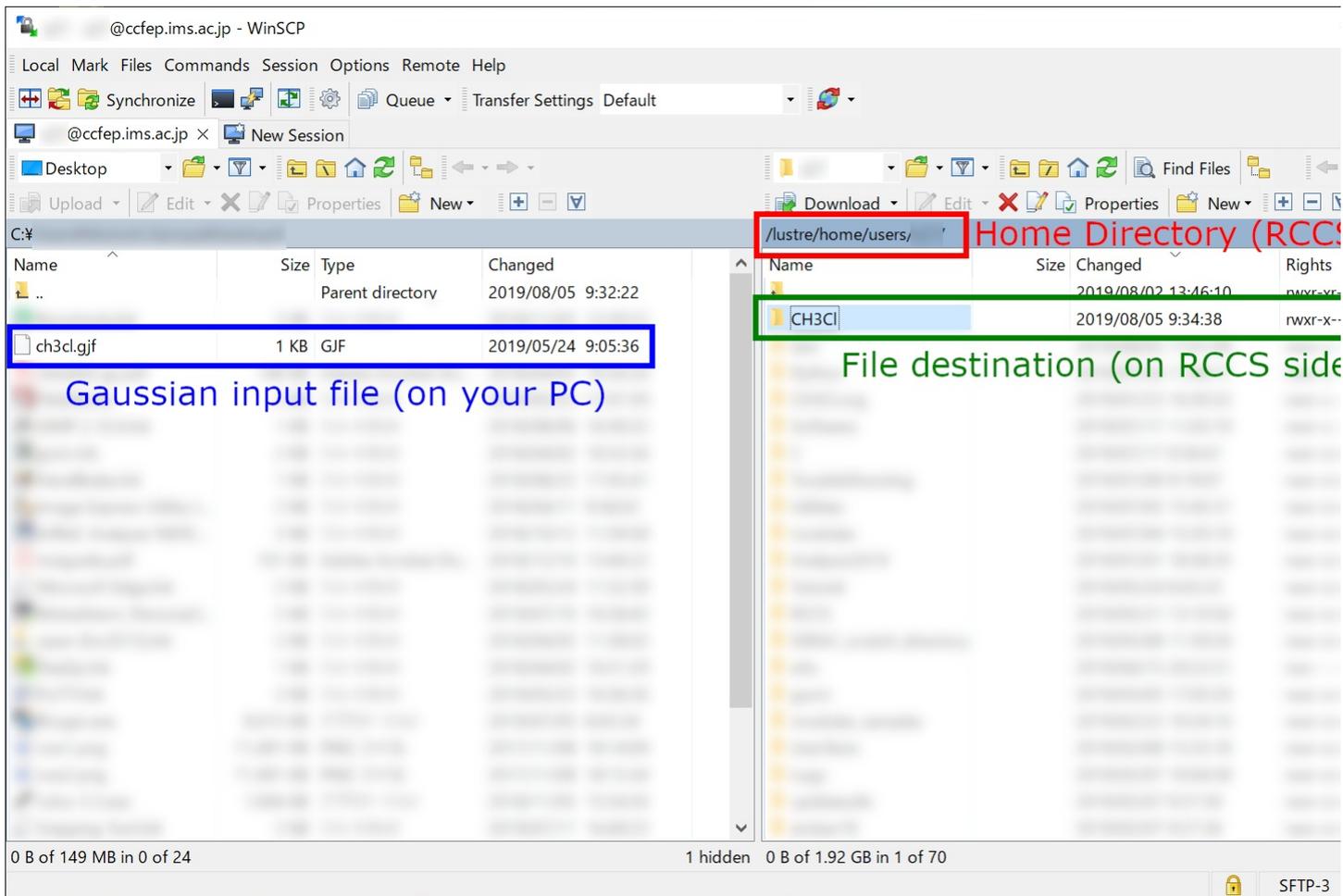
File Transfer (1)

We employ WinSCP in this example.

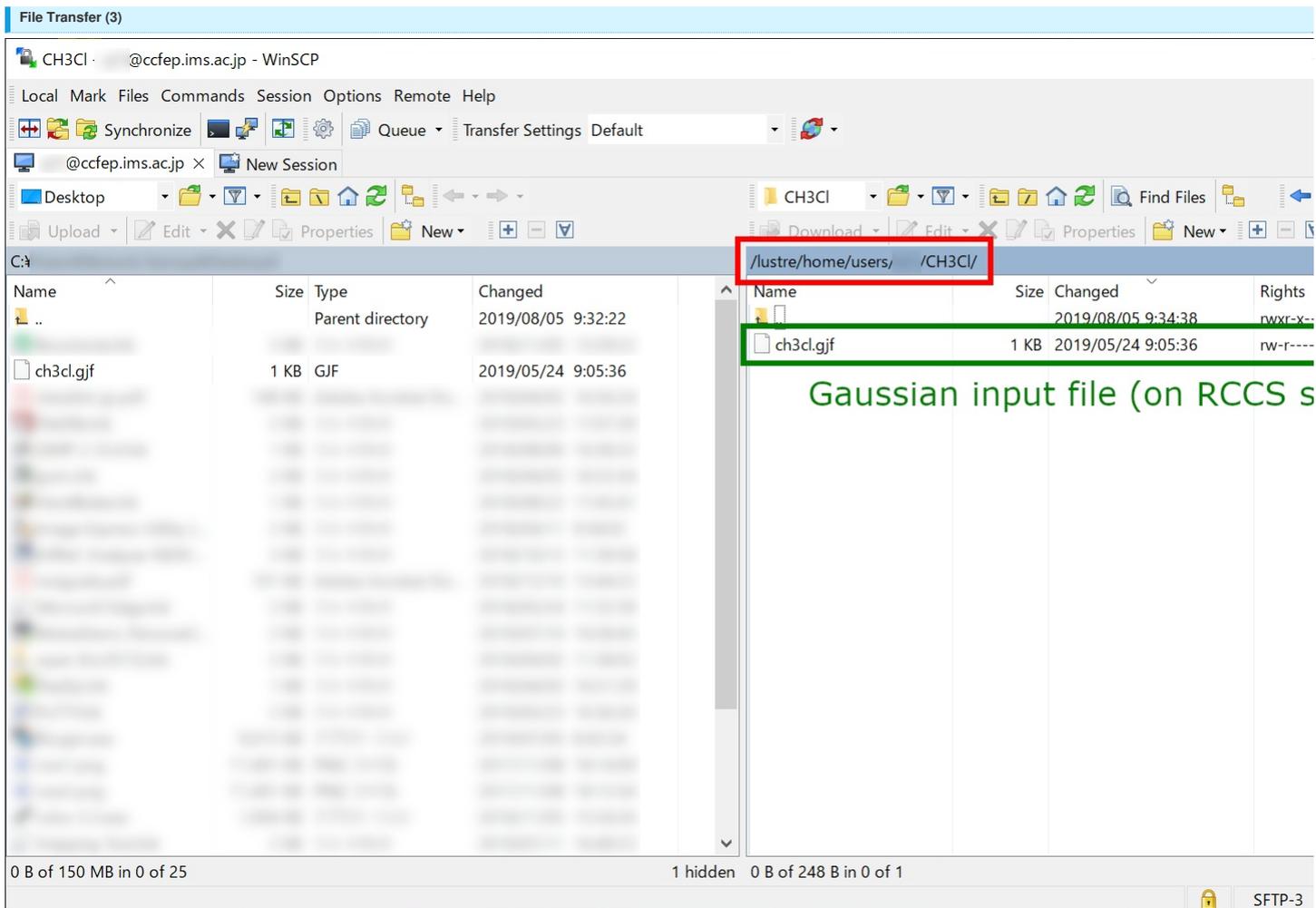
The screenshot shows the WinSCP interface with two panes. The left pane shows the local PC's file system with a file named 'ch3cl.gjf' highlighted in blue. The right pane shows the remote server's file system with a directory structure. A blue text overlay reads 'Gaussian Input (on your PC)'. The status bar at the bottom indicates '0 B of 149 MB in 0 of 23' for the local side and '0 B of 1.92 GB in 0 of 69' for the remote side. The connection is identified as 'SFTP-3'.

Connect to ccfep.ims.ac.jp, and move to the folder (your PC side) where the Gaussian input file exists. (Note: on RCCS system, /lustre/home/users/(username) and /home/users/(username) are the same path.)

File Transfer (2)



You can create a directory on RCCS side for this job. In this example, we created CH3Cl directory under the home directory (/home/users/(username)/CH3Cl).



Send input file (e.g. ch3cl.gjf) to RCCS. Once the file transfer finished, you then need to login to RCCS using SSH.

Submit a Gaussian job with g16sub

Login to cccp.ims.ac.jp (using PuTTY or something). Then, go to the directory where you placed input file (CH3CI directory in this example) using "cd" command. You can see file list of directories using "ls" command.

```
Last login: Thu May 23 17:26:40 2019 from *****
[user@cclep8 ~]$ ls CH3CI/
ch3cl.gjf
[user@cclep8 ~]$ cd CH3CI/
[user@cclep8 CH3CI]$ ls
ch3cl.gjf
[user@cclep8 CH3CI]$
```

In the target directory (CH3CI here), submit a Gaussian job with g16sub command.

```
[user@cclep8 CH3CI]$ g16sub ch3cl.gjf
```

If you submit a job without any arguments other than input file name like above, this job will use 6 CPU cores on a node of "core" jobtype. The walltime limitation of the job is set to 72 hours. If the job won't finish within the time, the job will be killed by the system.

Some lengthy information like below will be shown when you run the command.

- ▶ Green colored part in the box below shows basic parameters of the job queue, which involve memory about per CPU core for example.
- ▶ Blue part in the box below will show the summary of your job.
- ▶ At the last line, you may see (number).ccoms1 if your job is successfully submitted. The number (colored red in the box below) is the unique job ID of your job.

```
[user@cclep8 CH3CI]$ g16sub ch3cl.gjf
QUEUE detail
-----
QUEUE(MACH) Jobtype MaxMem DefMem TimLim DefCPUs(Min-Max)
-----
PN( k) core 4.8GB 4.0GB 72:00:00 6(1-36)
-----
Job detail
-----
MCL name(s) : ch3cl
INP file(s) : ch3cl.gjf.lx
OUT file(s) : ch3cl.out
Current dir : /lustre/home/users/****/CH3CI
SCRATCH dir : /work/users/$(USER)/$(PBS_JOBID)
QUEUE      : PN
Memory     : 24.0GB
Time limit : 72:00:00
Job script  : /lustre/home/users/****/CH3CI/PN_28254.sh
Input modified : y
-----
/lustre/local/bin/jsub -q PN /lustre/home/users/****/CH3CI/PN_28254.sh
4529602.ccoms1
[user@cclep8 CH3CI]$
```

Verify job status

You can see the status of the job with "jobinfo" command. (Job might not be shown immediately after the job submission. Please wait a while in this case.)

```
[user@cclep8 CH3CI]$ jobinfo -q PN -c -l
-----
Queue Job ID Name      Status CPUs User/Grp Elaps Node/Reason
-----
PN 4529602 PN_28254.sh Run 6 ***/... 00:00:00 cccc120
-----
[user@cclep8 CH3CI]$
```

- ▶ Job ID: the unique ID of your job, corresponding to the last line of g16sub output.
- ▶ Status of the job: "Run" => job is running, "Queue" => job is not yet running.
- ▶ Elapsed time and running node name: The elapsed time is running/waiting time for "Run"/"Queue" job, respectively.
 - ▶ If "Run", the rightmost column shows the node name(s) where your job is running.
 - ▶ If "Queue", the rightmost column shows the reason why the job is not running.
 - ▶ (cpu) enough CPU cores not available
 - ▶ (long) job walltime is too long (jobs must be finished before the maintenance time). It will run after the next maintenance.
 - ▶ (other) other reasons. Just submitted jobs will show this reason.

Once the job is successfully accepted by the queue, the job exists (and will run eventually) even after you closed the SSH connection.

Job finished...

Once the job finished, you may see new files in the directory. Among them, PN_*** (PN_28254.sh, PN_28254.sh.e4529602, PN_28254.sh.o4529602) and ***.lx (ch3cl.gjf.lx) are the files created by g16sub/g09sub. They are not necessary if the job finished successfully.

```
[user@cclep8 CH3CI]$ ls
PN_28254.sh PN_28254.sh.o4529602 ch3cl.gjf ch3cl.out
PN_28254.sh.e4529602 ch3cl.chk ch3cl.gjf.lx
[user@cclep8 CH3CI]$
```

Gaussian output file (ch3cl.out here) can be checked during job execution using "less" or "tail" command. Checkpoint file (if requested) will also be found in the same directory.

Run formchk

You can run formchk on RCCS frontend. But you need to run the following command before running formchk.

In case your login shell is /bin/bash or /bin/zsh:

```
[user@cclep8 CH3CI]$ source /local/apl/lx/g16/g16/bsd/g16.profile
```

In case your login shell is /bin/csh:

```
[user@cclep8 CH3CI]$ source /local/apl/lx/g16/g16/bsd/g16.login
```

This command won't output anything. But the setting will be modified. Now you can run formchk.

```
[user@cclep8 CH3CI]$ formchk ch3cl.chk ch3cl.fchk
Read checkpoint file ch3cl.chk
Write formatted file ch3cl.fchk
FCHKPn: Coordinates translated and rotated
FCHKPn: Coordinatesmatch /B/ after translation and rotation
[user@cclep8 CH3CI]$
```

If you are not sure what your login shell is, please run "echo \$SHELL" command like below.

```
[user@cclep8 CH3CI]$ echo $SHELL
```

Tips (1): g16sub options

These two lines below are equivalent.

```
[user@cclep8 CH3CI]$ g16sub ch3cl.gjf
[user@cclep8 CH3CI]$ g16sub -j core -rev g16c01 -np 6 -walltime 72:00:00 ch3cl.gjf
```

- ▶ -j: specify jobtype (default: core). You might want to use "small" jobtype (-j small) for large jobs.
- ▶ -np: specify number of CPU cores (if jobtype is core, default value is 6).
- ▶ -walltime: time limit of the job (default: 72:00:00 (72 hours)).

You can change settings by these options. For example, if you specify -np 4 in the command line, the submitted job will use 4 CPU cores.

Tips (2): available Gaussian revisions

Following versions and revisions of Gaussian are available on RCCS system.

- ▶ Gaussian 16 Rev. C.01 (g16sub -rev g16c01) (default for g16sub)
- ▶ Gaussian 16 Rev. B.01 (g16sub -rev g16b01)
- ▶ Gaussian 16 Rev. A.03 (g16sub -rev g16a03)

- ▶ Gaussian 09 Rev. E.01 (g09sub -rev g09e01) (default for g09sub)
- ▶ Gaussian 09 Rev. D.01 (g09sub -rev g09d01)
- ▶ Gaussian 09 Rev. C.01 (g09sub -rev g09c01)
- ▶ Gaussian 09 Rev. B.01 (g09sub -rev g09b01)

Tips (3): number of CPU cores

- ▶ (mainly tips for jobtype=core; multi-node parallel run is not available in RCCS due to lack of TCP Linda)
- ▶ More of CPU cores does not mean more faster run. Too many CPU cores could decrease performance.
 - ▶ In terms of cost-performance, smaller number of cores is favourable.
 - ▶ You job will required quite a long time with too less number of CPU cores, though.
 - ▶ The optimal number severely depends on the situation, your intention, and many others.

Tips (4): working (scratch) directory

Working (scratch) directory of the job is determined using the unique job ID.

```
SCRATCH dir: /work/users/****/PBS_JOBID)
```

"\${PBS_JOBID}" will be replaced by jobid (4529602.cccms1 for the example above). This directory name can be found at the beginning of the output file (ch3cl.out for the example in this page). But please note that the working directory will be automatically removed when the job is finished within the specified walltime.

If you want to change this behaviour, please see the section below.

Tips (5): run Gaussian without g09sub/g16sub

You need to prepare input file template. Once the job script is prepared, you just need to run that job with "jsub" command instead of g16sub.

- ▶ Use sample in /local/apl/lx/g16c01/samples (Gaussian 16 C.01 case) directory as a template file.
- ▶ Run g16sub with -P option; you can use the generated job script (PN_****) as your input template.

In either of cases, please submit the job with "jsub" command instead of g16sub/g09sub.