

General Interest Seminar

# Using Computational Chemistry Software Effectively on Graham

Jemmy Hu

SHARCNET HPC Consultant

University of Waterloo

March 28, 2018

# Packages available on Graham

<http://wiki.computecanada.ca>

Discipline guides

or

Search: **Computational chemistry**

[https://docs.computecanada.ca/wiki/  
Computational\\_chemistry](https://docs.computecanada.ca/wiki/Computational_chemistry)

# The nature of the popular packages (I)

ab initio	DFT	Molecular mechanics	Tools
Gaussian	ADF	Amber	Molden
NWChem	Quantum Espresso	Gromacs	VMD
Gamess	CP2K	CPMD	Plumed
ORCA	deMon	NAMD	OpenMM
VASP	PSI4	LAMMPS	
Siesta		DL_POLY	

# The nature of the popular packages (II)

- Commercial packages
  - **Gaussian** (soft\_Gaussian, Graham only at the moment)
  - **ADF** (Graham only)
  - **Amber** (Graham only)
  
  - **VASP** (soft\_vasp5, PI's group license)  
<https://docs.computecanada.ca/wiki/VASP>
  - **ORCA** (soft\_orca, register to ORCA is required)  
<https://docs.computecanada.ca/wiki/ORCA>
- Parallel coding approach
  - **MPI** distributed parallel solution, run on cpus across multi-node
  - **OpenMP** shared memory solution (Gaussian), cpus on one node

# Software modules

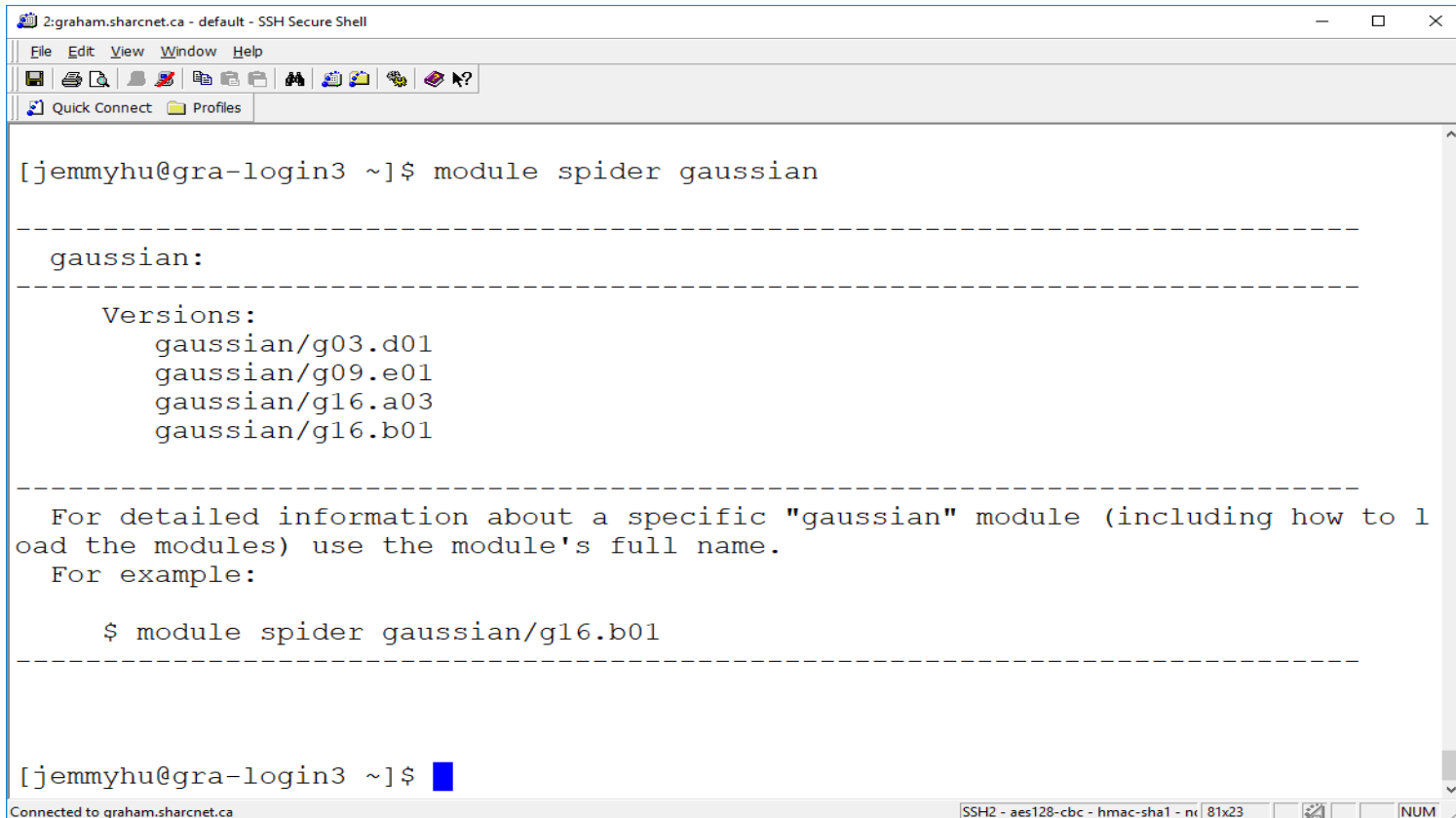
- Software are installed as Module packages
- Basic module commands

`module avail`

# to list all available module software

`module spider softwareName`

# to see what versions are available



```
2:graham.sharcnet.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles

[jemmyhu@gra-login3 ~]$ module spider gaussian

-----
gaussian:
-----

Versions:
  gaussian/g03.d01
  gaussian/g09.e01
  gaussian/g16.a03
  gaussian/g16.b01

-----

For detailed information about a specific "gaussian" module (including how to load the modules) use the module's full name.
For example:

  $ module spider gaussian/g16.b01

-----

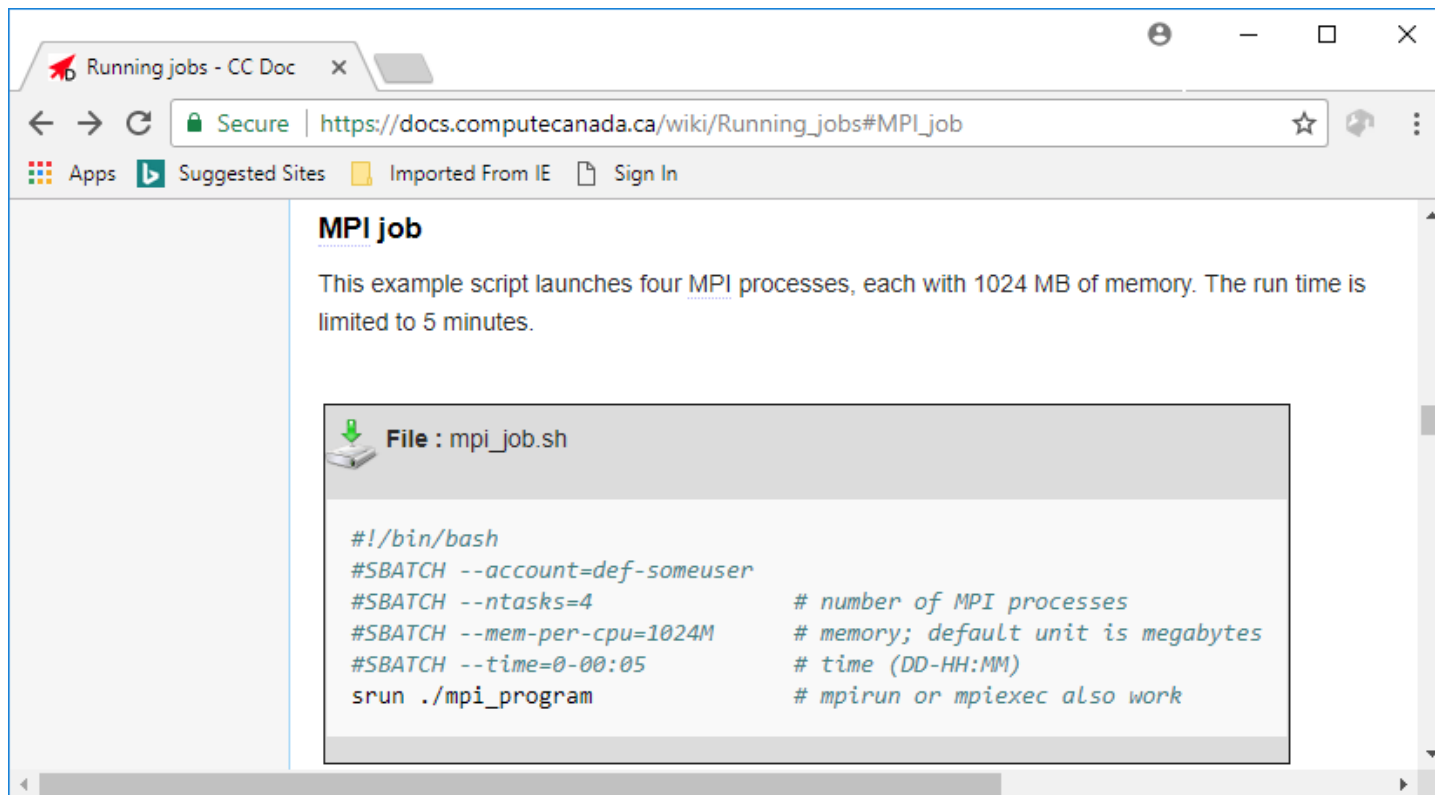
[jemmyhu@gra-login3 ~]$
```

Connected to graham.sharcnet.ca

SSH2 - aes128-cbc - hmac-sha1 - nt 81x23 NUM

# Submit/Run script:

- account type (default or RAC account)
  - compute resource (cpus, memory, runtime)
  - job type (mpi, openmp, serial)
  - software (module load, run command)
- 
- `mpi_job.sh`



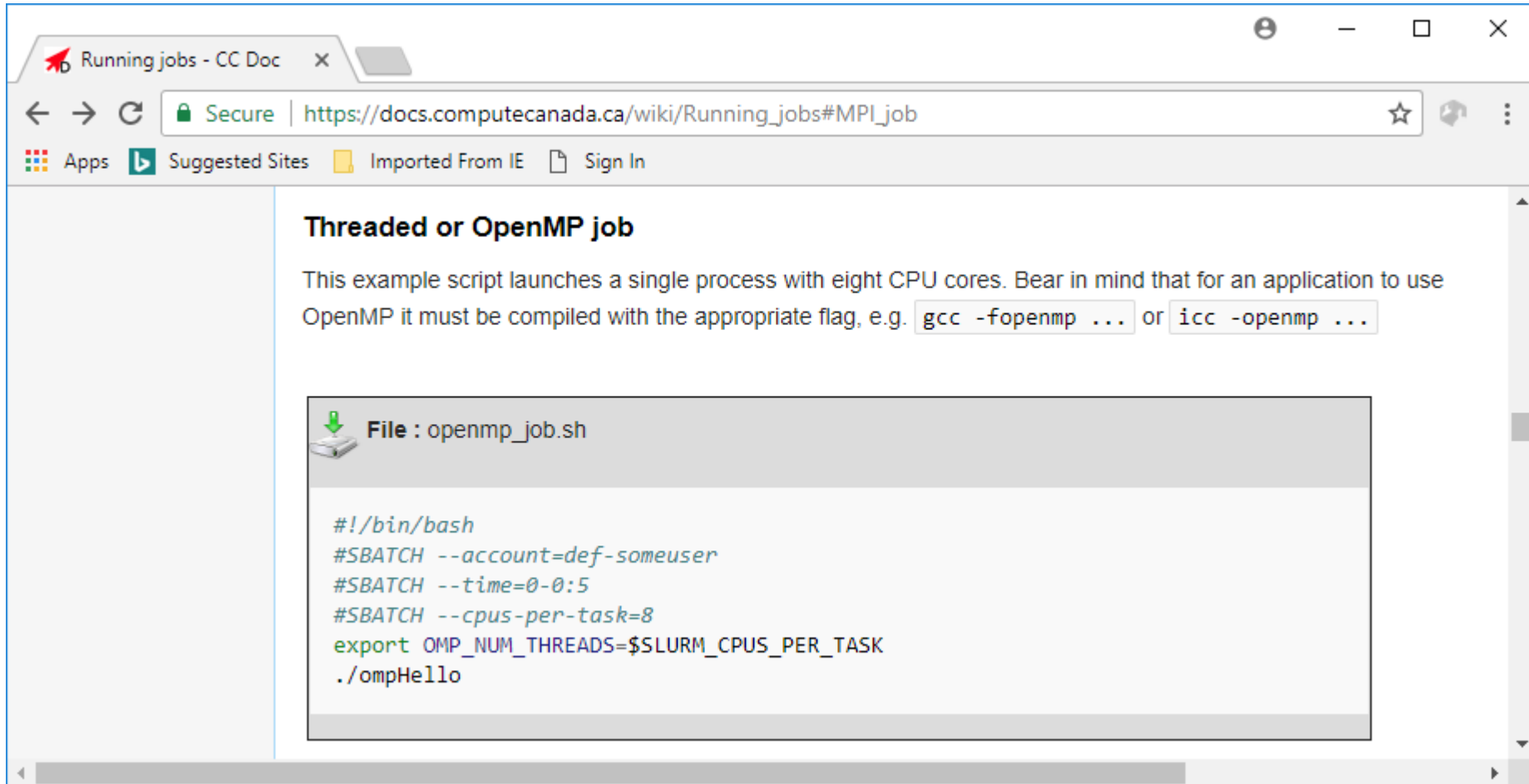
The screenshot shows a web browser window with the address bar displaying `https://docs.computecanada.ca/wiki/Running_jobs#MPI_job`. The page title is "MPI job". The main content area contains the following text:

This example script launches four MPI processes, each with 1024 MB of memory. The run time is limited to 5 minutes.

Below the text is a code block for the script `mpi_job.sh`:

```
#!/bin/bash
#SBATCH --account=def-someuser
#SBATCH --ntasks=4           # number of MPI processes
#SBATCH --mem-per-cpu=1024M  # memory; default unit is megabytes
#SBATCH --time=0-00:05      # time (DD-HH:MM)
srun ./mpi_program          # mpirun or mpiexec also work
```

# •openmp\_job.sh



Running jobs - CC Doc

Secure | [https://docs.computecanada.ca/wiki/Running\\_jobs#MPI\\_job](https://docs.computecanada.ca/wiki/Running_jobs#MPI_job)

Apps Suggested Sites Imported From IE Sign In

## Threaded or OpenMP job

This example script launches a single process with eight CPU cores. Bear in mind that for an application to use OpenMP it must be compiled with the appropriate flag, e.g. `gcc -fopenmp ...` or `icc -openmp ...`

**File : openmp\_job.sh**

```
#!/bin/bash
#SBATCH --account=def-someuser
#SBATCH --time=0-0:5
#SBATCH --cpus-per-task=8
export OMP_NUM_THREADS=$SLURM_CPUS_PER_TASK
./ompHello
```

[... ~]\$ **SBATCH** --help

# Slurm commands

[https://docs.computecanada.ca/wiki/Running\\_jobs](https://docs.computecanada.ca/wiki/Running_jobs)

- `sbatch name.sh` # submit the job
- `squeue -u username` # check your job status
- `scontrol show job -dd jobid` # detailed info about a job
- `sacct -j jobid` # info about a finished job
- `scancel jobid` # cancel a job with the jobid
- `scancel -t PENDING -u username` # cancel all your pending jobs

interactive job (test input):

```
salloc --time=1:0:0 --ntasks=8 --mem=10g --account=def-someuser
```



```
graham.sharcnet.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles

[jemmyhu@gra-login3 g16.b01]$ queue -u jemmyhu
      JOBID      USER      ACCOUNT      NAME  ST  START_TIME      TIME_LEFT  NODES
CPUS  GRES  MIN_MEM  NODELIST  (REASON)
3504611  jemmyhu  def-jemmyhu_  g16_test_G16.s  R  2018-03-21T15:18  1-01:55:39    1
  16 (null)      8G  gra91  (None)
[jemmyhu@gra-login3 g16.b01]$
[jemmyhu@gra-login3 g16.b01]$
[jemmyhu@gra-login3 g16.b01]$
[jemmyhu@gra-login3 g16.b01]$ ls
g16_test.chk  g16_test.com  g16_test_g16.sh  g16_test_G16.sh  g16_test.log  tests
[jemmyhu@gra-login3 g16.b01]$ scontrol show job -dd 3504611
JobId=3504611 JobName=g16_test_G16.sh
  UserId=jemmyhu(3000561) GroupId=jemmyhu(3000561) MCS_label=N/A
  Priority=883632 Nice=0 Account=def-jemmyhu_cpu QOS=normal
  JobState=RUNNING Reason=None Dependency=(null)
  Requeue=0 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
  DerivedExitCode=0:0
  RunTime=00:05:06 TimeLimit=1-02:00:00 TimeMin=N/A
  SubmitTime=2018-03-21T15:18:37 EligibleTime=2018-03-21T15:18:37
  StartTime=2018-03-21T15:18:39 EndTime=2018-03-22T17:18:39 Deadline=N/A
  PreemptTime=None SuspendTime=None SecsPreSuspend=0
  LastSchedEval=2018-03-21T15:18:39
  Partition=cpubase_bycore_b4 AllocNode:Sid=gra-login3:14407
  ReqNodeList=(null) ExcNodeList=(null)
  NodeList=gra91
  BatchHost=gra91
  NumNodes=1 NumCPUs=16 NumTasks=1 CPUs/Task=16 ReqB:S:C:T=0:0:*:*
  TRES=cpu=16,mem=8G,node=1,billing=16
  Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
    Nodes=gra91 CPU_IDs=4-8,16-20,22,24,28-31 Mem=8192 GRES_IDX=
  MinCPUsNode=16 MinMemoryNode=8G MinTmpDiskNode=0
```

# Gaussian

<https://docs.computecanada.ca/wiki/Gaussian> (g## or G##)

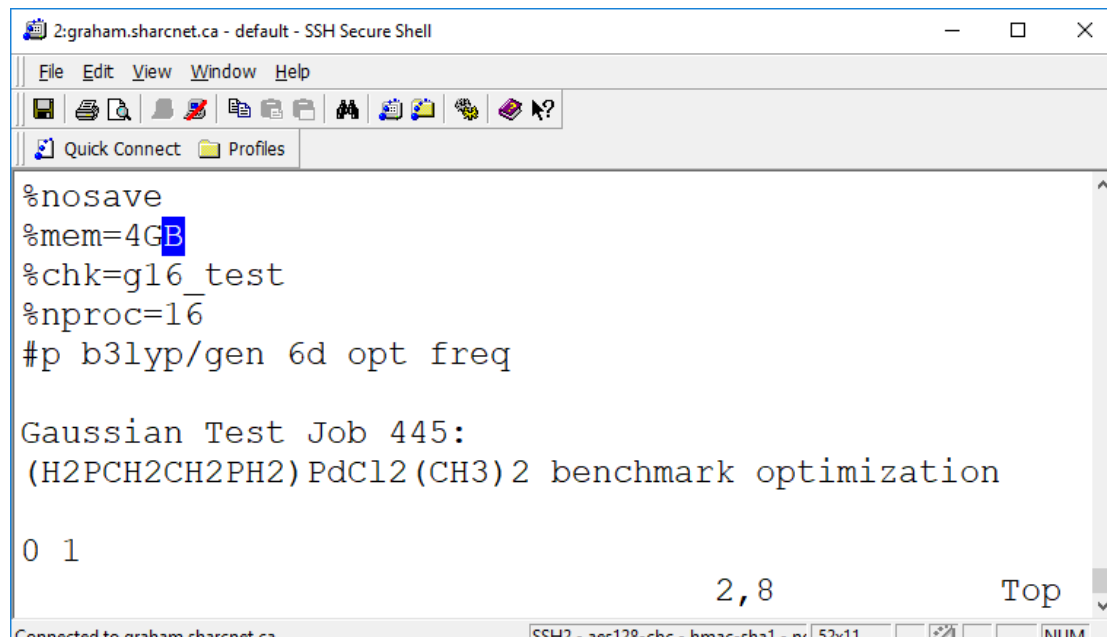
- Versions

- g03.d01

- g09.e01

- g16.a03

- g16.b01

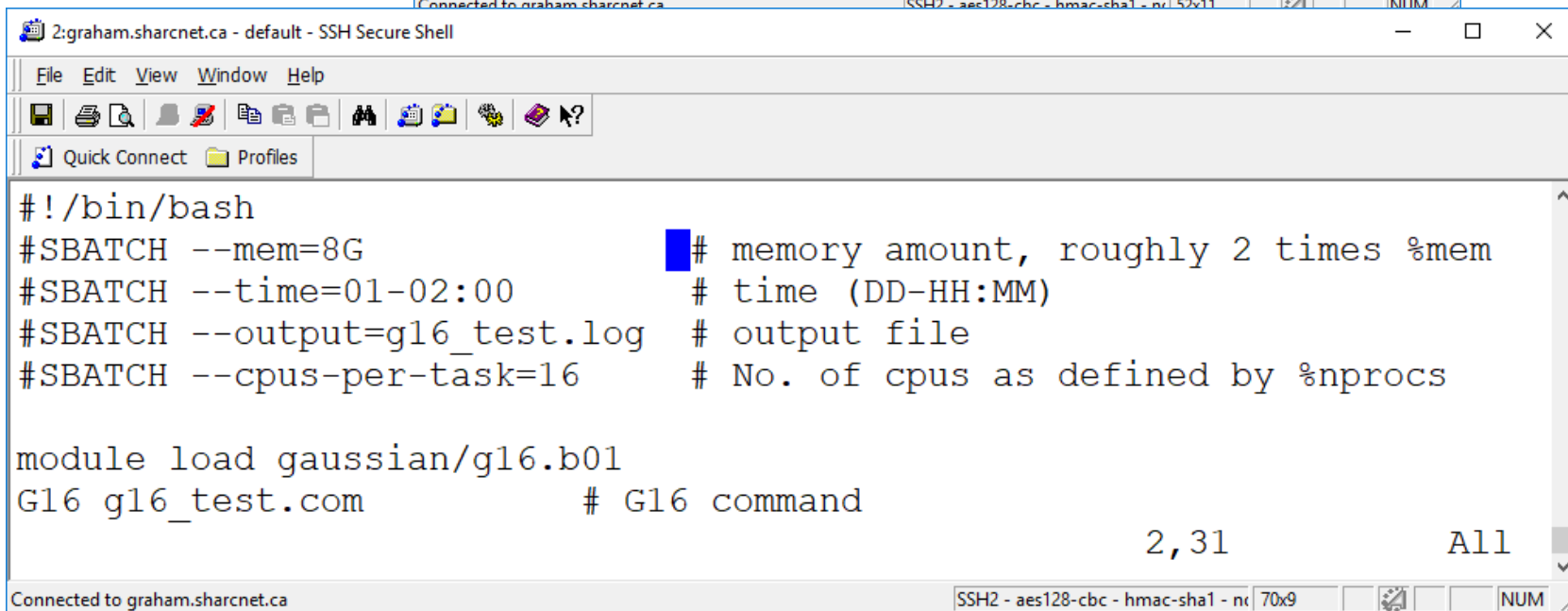


```
2:graham.sharcnet.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
%nosave
%mem=4GB
%chk=g16_test
%nproc=16
#p b3lyp/gen 6d opt freq

Gaussian Test Job 445:
(H2PCH2CH2PH2)PdCl2(CH3)2 benchmark optimization

0 1

2,8 Top
```



```
2:graham.sharcnet.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
#!/bin/bash
#SBATCH --mem=8G          # memory amount, roughly 2 times %mem
#SBATCH --time=01-02:00  # time (DD-HH:MM)
#SBATCH --output=g16_test.log # output file
#SBATCH --cpus-per-task=16 # No. of cpus as defined by %nprocs

module load gaussian/g16.b01
G16 g16_test.com          # G16 command

2,31 All
```

# Tips to run Gaussian on Graham

- use the latest version

g16.b01

g09.e01 (if you need to include NBO6)

- use 16 or 8 cpus:

```
%nproc=16 #input name.com
```

```
#SBATCH --cpus-per-task=16 # gaussian name.sh
```

- #SBATCH --mem=8G is ~2 times %mem=4GB in the input \*.com

- File location: G16/G09, /scratch/userid/jobid/

- Testing input .com

```
[ ~$] salloc --time=1:0:0 --cpus-per-task=16 --mem=10g --account=def-PI
```

```
[ ~$] module load Gaussian/g16.b01
```

```
[ ~$] G16 g16_test.com
```

# ADF

<https://docs.computecanada.ca/wiki/ADF>

- input files
  - one step job
  - multi-step job
- submit script (node-based)



This mysub.sh script is for a whole-node job. The last two lines load version 2017.207 and call ADF directly.

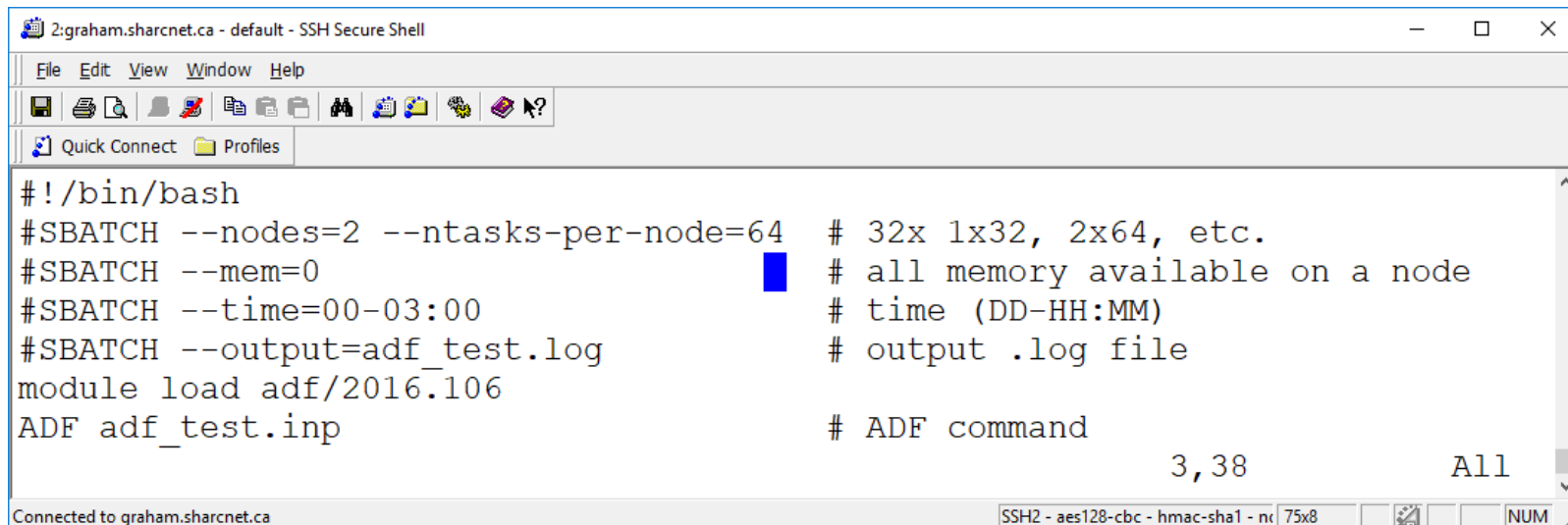
```
File : mysub.sh

#!/bin/bash
#SBATCH --nodes=1 --ntasks-per-node=32 # 1 node with 32 cpus, you can modify it
#SBATCH --mem=0 # request all memory on node
#SBATCH --time=00-03:00 # time (DD-HH:MM)
#SBATCH --output=adf_test-%j.Log # output file

module load adf/2017.207
ADF adf_test.inp
```

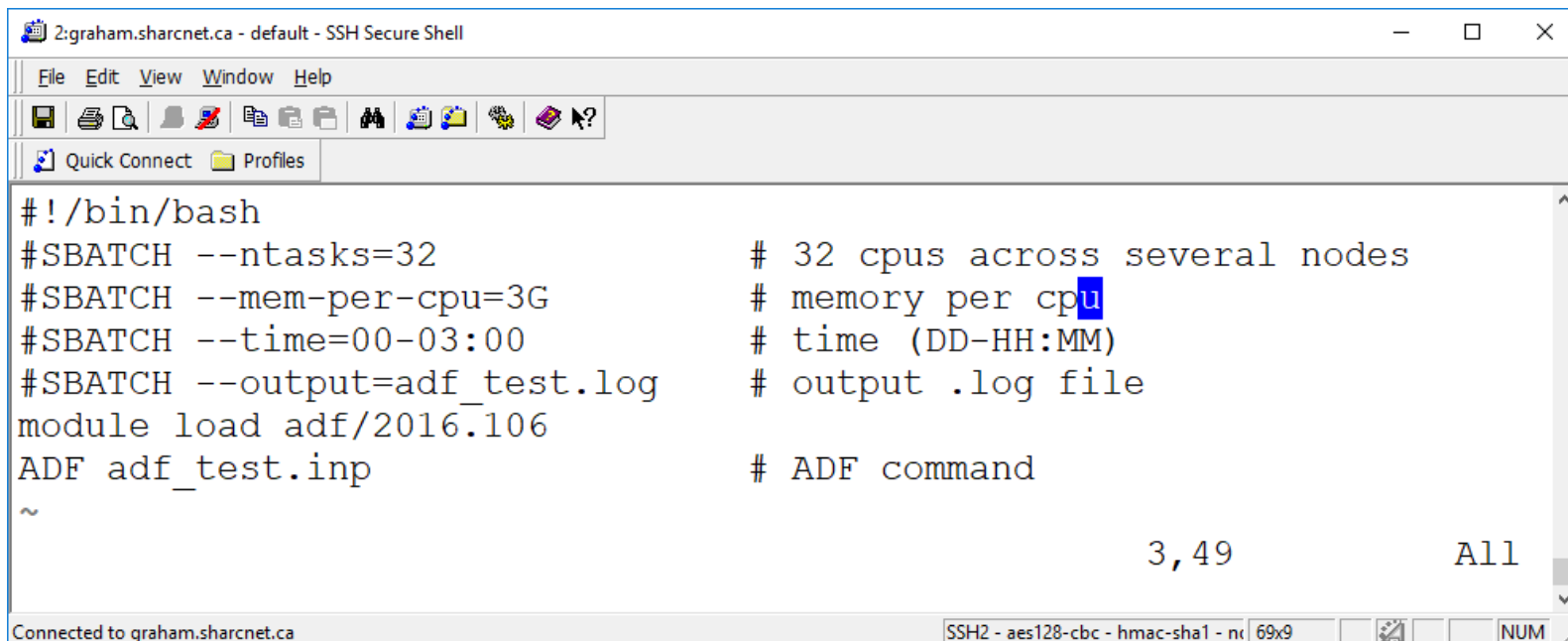
- Submit the job: **sbatch mysub.sh**

## MPI by-node partition: --nodes --ntasks-per-node (32x)



```
2:graham.sharcnet.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
#!/bin/bash
#SBATCH --nodes=2 --ntasks-per-node=64 # 32x 1x32, 2x64, etc.
#SBATCH --mem=0 # all memory available on a node
#SBATCH --time=00-03:00 # time (DD-HH:MM)
#SBATCH --output=adf_test.log # output .log file
module load adf/2016.106
ADF adf_test.inp # ADF command
3,38 All
Connected to graham.sharcnet.ca SSH2 - aes128-cbc - hmac-sha1 - nc 75x8 NUM
```

## MPI by-core partition: --ntasks, --mem-per-cpu



```
2:graham.sharcnet.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
#!/bin/bash
#SBATCH --ntasks=32 # 32 cpus across several nodes
#SBATCH --mem-per-cpu=3G # memory per cpu
#SBATCH --time=00-03:00 # time (DD-HH:MM)
#SBATCH --output=adf_test.log # output .log file
module load adf/2016.106
ADF adf_test.inp # ADF command
~
3,49 All
Connected to graham.sharcnet.ca SSH2 - aes128-cbc - hmac-sha1 - nc 69x9 NUM
```

## ORCA: binary orca/4.0.1.2

load the corresponding modules



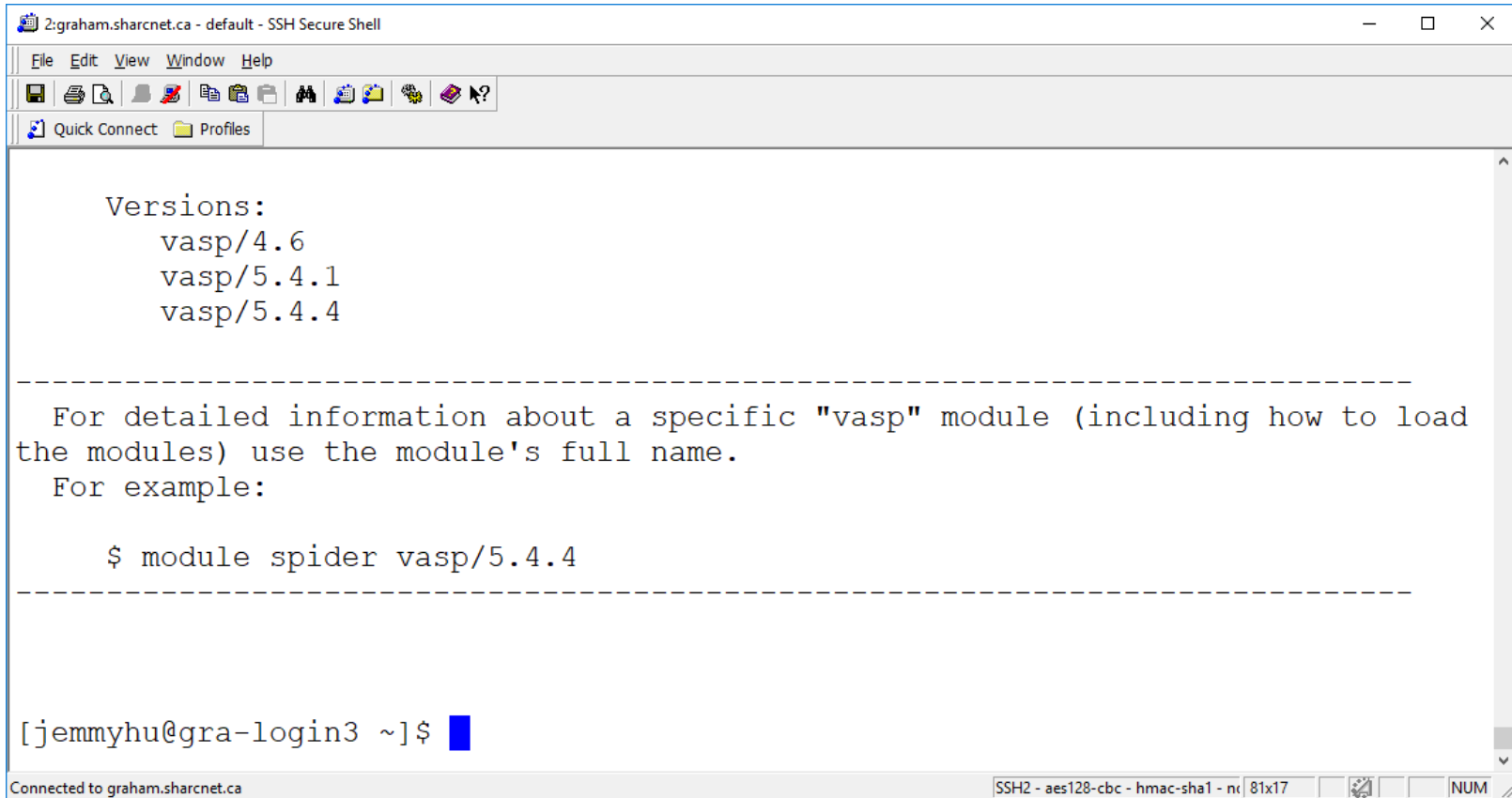
The screenshot shows a web browser window with three tabs: SHARCNET: ANSYS (App), GAUSSIAN - Documenta, and ORCA - CC Doc. The address bar shows the URL <https://docs.computecanada.ca/wiki/ORCA>. The main content area displays a file named `run_orca.sh` with the following shell script content:

```
#!/bin/bash
#SBATCH --ntasks=8           # cpus, the nprocs defined in the input file
#SBATCH --mem-per-cpu=3G     # memory per cpu
#SBATCH --time=00-03:00      # time (DD-HH:MM)
#SBATCH --output=benzene.log # output .log file

module load openmpi/2.0.2
module load orca/4.0.1.2
$EBROOTORCA/orca benzen.inp
```

<https://docs.computecanada.ca/wiki/ORCA>

```
[jemmyhu@gra-login3 ~]$ module spider vasp
```



```
z:graham.sharcnet.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles

Versions:
  vasp/4.6
  vasp/5.4.1
  vasp/5.4.4

-----

For detailed information about a specific "vasp" module (including how to load
the modules) use the module's full name.
For example:

  $ module spider vasp/5.4.4

-----

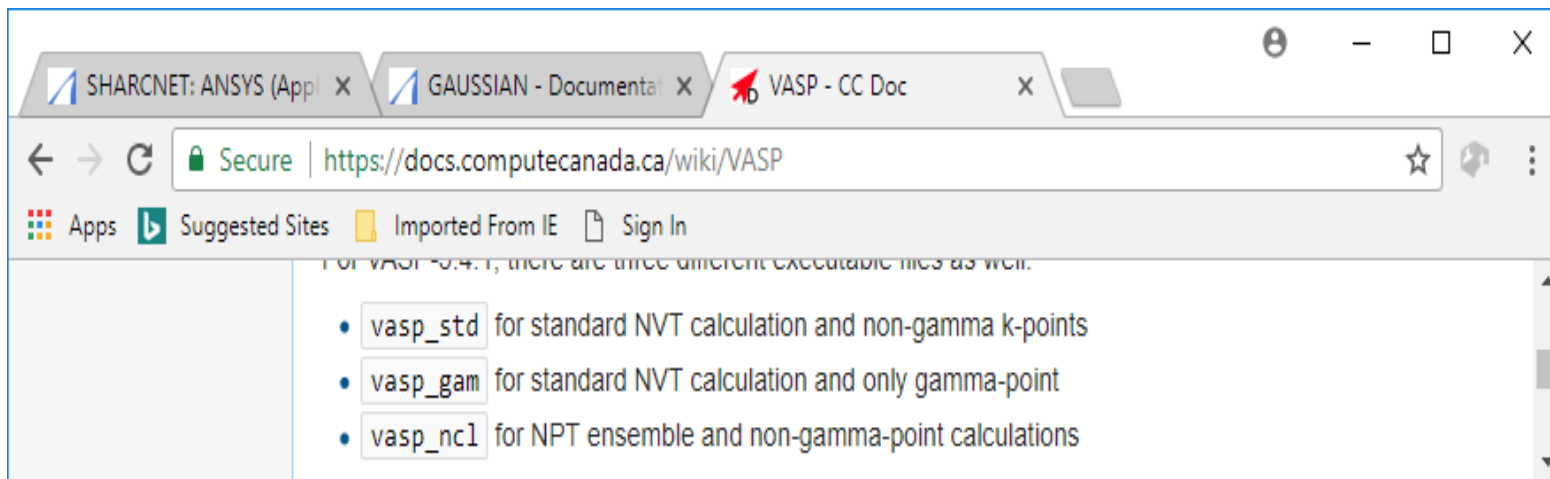
[jemmyhu@gra-login3 ~]$ █
```

Connected to graham.sharcnet.ca

SSH2 - aes128-cbc - hmac-sha1 - nt 81x17 NUM

<https://docs.computecanada.ca/wiki/VASP>

## <VASP> executables



**VASP input files:** INCAR, KPOINTS, POSCAR, POTCAR (each job in a subdirectory)

```
1:graham.sharcnet.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
#!/bin/bash
#SBATCH --nodes=1 --ntasks-per-node=32 # 1 node with 32 cpus
#SBATCH --mem=0 # all memory on the node
#SBATCH --time=00-03:00 # time (DD-HH:MM)
#SBATCH --output=vasp_test.log # output .log file
module load vasp/5.4.1
srun vasp_std
5,55 All
Connected to graham.sharcnet.ca SSH2 - aes128-cbc - hmac-sha1 - n... 70x8 NUM
```

[ ~\$] sbatch vasp.sh

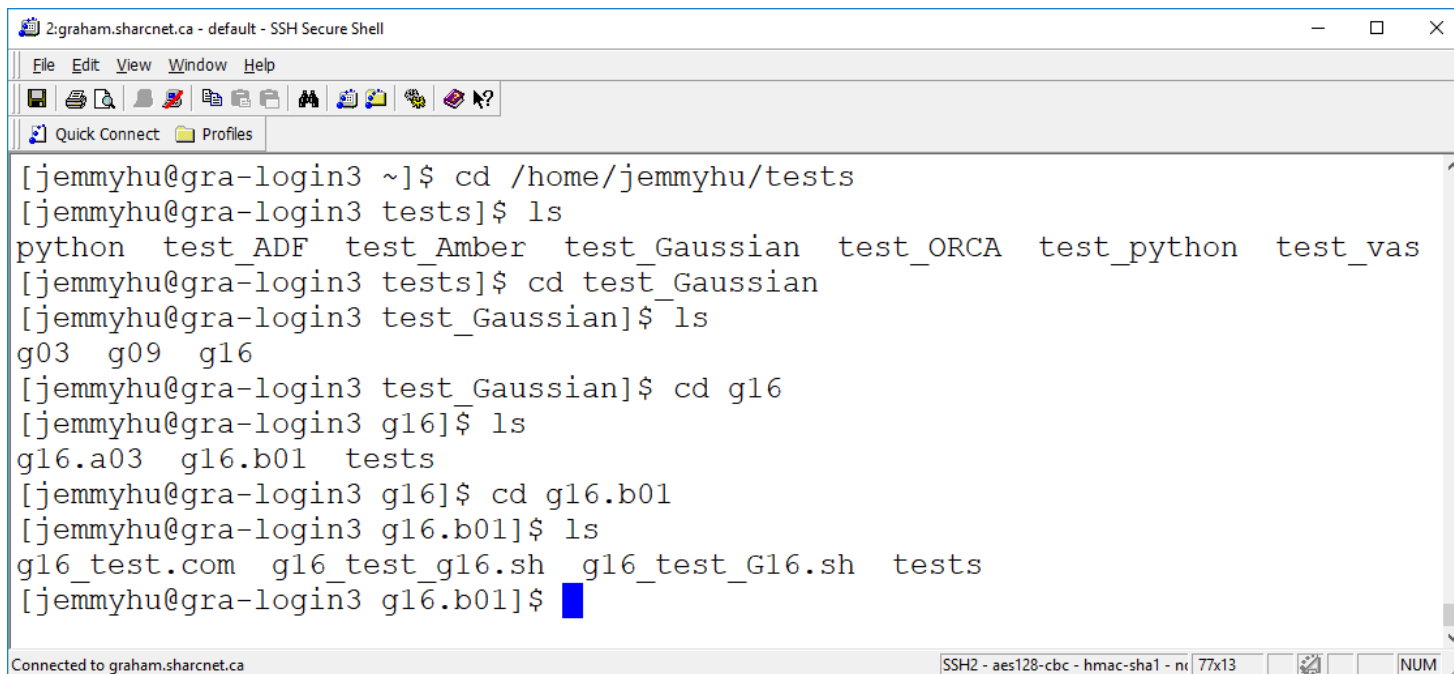


# Resource and Question

- Online wiki documents

<http://wiki.computecanada.ca>

- Examples



```
2:graham.sharcnet.ca - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
[jemmyhu@gra-login3 ~]$ cd /home/jemmyhu/tests
[jemmyhu@gra-login3 tests]$ ls
python test_ADF test_Amber test_Gaussian test_ORCA test_python test_vas
[jemmyhu@gra-login3 tests]$ cd test_Gaussian
[jemmyhu@gra-login3 test_Gaussian]$ ls
g03 g09 g16
[jemmyhu@gra-login3 test_Gaussian]$ cd g16
[jemmyhu@gra-login3 g16]$ ls
g16.a03 g16.b01 tests
[jemmyhu@gra-login3 g16]$ cd g16.b01
[jemmyhu@gra-login3 g16.b01]$ ls
g16_test.com g16_test_g16.sh g16_test_G16.sh tests
[jemmyhu@gra-login3 g16.b01]$
```

Connected to graham.sharcnet.ca SSH2 - aes128-cbc - hmac-sha1 - ni 77x13 NUM

- Questions to

- email to [support@computecanada.ca](mailto:support@computecanada.ca)

to the online ticketing system