How to Jomon

Version 1.2

By flopez

March 28, 2020

Welcome! You are probably now under Cristopher command (or your professor wants you to do some calculations because why not?) and eventually, he will give you a series of instructions which inevitably will lead to dealing with Jomon. There's a big chance you know nothing about Jomon or Linux (check this http://linuxcommand.org/index.php for some command line relief), so here's a little help on how to proceed. Enjoy!

1 Connecting with Jomon

1.1-1.5 are only applicable for Mac or Linux users, Windows users need 3 programs PuTTY (https://www.putty.org/) (1.1-1.3), pUTTYgen (https://www.puttygen.com/) (1.4) and FileZilla (https://filezilla-project.org/) (1.5) or install the Windows Subsystem for Linux (https://docs.microsoft.com/en-us/windows/wsl/install-win10).

1.1 Establishing an SSH session

Open a terminal program and type:

```
$ ssh your-user-in-jomon@163.178.48.12
```

1.2 Setting the config file

In your computer, go to ∼/.ssh and do the following:

```
$ touch config
```

Using your text editor of choice (probably vim), add the following lines to config:

```
Host jomon
        HostName 163.178.48.12
        User your-user-in-jomon
```

Now connecting with jomon will be easier since you only need to type this:

```
$ ssh jomon
```

1.3 Change your password

```
$ ssh jomon
$ passwd
```

1.4 Deploying SSH keys

Now let's automate even more the log in to jomon by using ssh-keygen (Check this https://www.ssh.com/ssh/keygen/ if you are interested in understanding). In your computer, open a terminal and type:

```
$ ssh-keygen -t rsa -b 4096
$ ssh-copy-id jomon
```

Now you don't need a password for logging in!

1.5 Copying files

1.5.1 Without 1.2 and 1.4

From your computer to Jomon:

```
$ scp (-r) file(directory) your-user-in-jomon@163.178.48.12:[/path/to/file(directory)]
```

From Jomon to your computer:

```
$ scp (-r) your-user-in-jomon@163.178.48.12:[/path/to/file(directory)] [path/you/want]
```

1.5.2 With 1.2 and 1.4

From your computer to Jomon:

```
$ scp (-r) file(directory) jomon:[/path/to/file(directory)]
```

From Jomon to your computer:

```
$ scp (-r) jomon:[/path/to/file(directory)] [path/you/want]
```

These commands must be executed in your computer.

2 Jomon's Queues System

2.1 Writing a PBS job file

Log in to Jomon and create do the following:

```
$ touch job-name-you-want.sh
$ chmod 744 job-name-you-want.sh
```

```
Now add the following lines to the .sh file:
#!/bin/bash

#PBS -l nodes=number-nodes-you-want(*):ppn=number-cores-you-want(**)
#PBS -l walltime=1008:00:00 (***)
#PBS -q default
#PBS -j oe
# * (Most likely 1 since we don't have InfiniBand)
# ** (Minimum is 1 and Maximum is 28 depending on what you are dealing with)
# *** (If you need more than 1008 you are crazy!!!)
```

```
cd $PBS_O_WORKDIR

# Set scratch directory
# Every program has its own variable for it or you can use the one from PBS.
# Always use these paths for scratch:
/scratch/$USER/$PBS_JOBID #Any program, common for Turbomole
/scratch/$USER/ #Any program
/scratch/$PBS_JOBID #GAMESS

# Move everything to the scratch directory
# Sometimes you have to do it, sometimes the program does it by itself

######## EXECUTE YOUR PROGRAM HERE ##########################################

# Eliminate whatever you don't need in the scratch directory
# Again sometimes you have to do it, sometimes the program does it
# Anyhow, ALWAYS eliminate what you don't need anymore!!!
```

See section 3 on how this is done for some programs.
If you are interested in learning about PBS and many more options and directives that can be added, check this: https://www.osc.edu/supercomputing/batch-processing-at-osc

2.2 Submitting a job

```
$ qsub job-name-you-want.sh
```

2.3 Monitoring the jobs

In Jomon, type:

```
$ watch qstat -options
or
$ qstat -options
```

Depending on how much information you want to see.

2.4 Deleting a job

```
$ qdel job-name-from-qstat
```

2.5 Retrieving results

All jobs produce a file corresponding with standard output [job-name].o[job-id], which must be empty if no errors occurred.

2.6 Checking Jomon

First say hi to Jomon, type:

```
$ hi
```

Which should give an output like this

```
n001 is up
.
.
.
n018 is up
```

If there is a missing node or none of them appear, then something bad is happening. Also you can use the pbsnodes command to see the status of all nodes (Valid state strings are "free", "offline", "down", "reserve", "job-exclusive", "job-sharing", "busy", "time-shared", or "state-unknown"). Most of the time nodes are either free (partially too) or job-exclusive, bad things happen if they are down or state-unknown, pray if they are in the last situations.

3 Some examples

These are a series of examples on how to do scripts to run jobs for programs we usually use (specially Turbomole), but, we encourage you to **ALWAYS ALWAYS ALWAYS** review the manual of the program you are willing to use.

3.1 TURBOMOLE

```
#!/bin/bash

#PBS -l nodes=1:ppn=cores
#PBS -l walltime=1008:00:00
#PBS -q default
#PBS -j oe

# Some programs have extra "things" to make the calculations
# easy-going depending on what you set above

## Set locale to C
unset LANG
unset LC_CTYPE

# Set stack size limit to unlimited:
ulimit -s unlimited

# Check if this is a parallel job
if [ $PBS_NUM_PPN -gt 1 ]; then

##### Parallel job
# Set environment variables for a SMP job
    export PARA_ARCH=SMP
    export PATH="${TURBODIR}/bin/`sysname`:${PATH}"
    export PARNODES=$PBS_NUM_PPN
else
##### Sequentiel job
```

```
# set the PATH for Turbomole calculations
    export PATH="${TURBODIR}/bin/`sysname`:${PATH}"
fi

# What's above is something extra from TURBOMOLE
# (you can find it in the manual, but, well, now you have it :) )

cd $PBS_O_WORKDIR

# Set scratch directory
export TURBOTMPDIR=/scratch/$USER/$PBS_JOBID
mkdir -p $TURBOTMPDIR

# move everything to the scratch directory
cp $PBS_O_WORKDIR/* $TURBOTMPDIR
cd $TURBOTMPDIR

######## ENTER YOUR JOB HERE ###################################################
module (dscf, escf, etc) > $PBS_O_WORKDIR/name-of-module.out
################################################################################

#Eliminate whatever you don't need in the scratch directory
cp -rf $TURBOTMPDIR/* $PBS_O_WORKDIR/
rm -rf $TURBOTMPDIR
```

3.2 GAMESS

```
#!/bin/bash

#PBS -l nodes=1:ppn=cores
#PBS -l walltime=1008:00:00
#PBS -q default
#PBS -j oe

cd $PBS_O_WORKDIR

# VERNO is a variable for the version of executable you want to use.
# Right now 00 should do the trick, but perhaps in the future would be 01, etc.
export VERNO=number

# GAMESS is a bit different when it comes to setting the scratch directory
mkdir -p /scratch/$PBS_JOBID

######## ENTER YOUR JOB HERE ###################################################
# You must look where the path of rungms is, it will always start with /opt...
# Some programs (like GAMESS) require the full path, others not
/opt/gamess/30.09.2018/intel/rungms name-input.inp $VERNO cores >& name-output.log;
################################################################################
```

```
#Eliminate whatever you don't need in the scratch directory
rm -rf /scratch/$PBS_JOBID
```

3.3 MOLPRO

```
#!/bin/bash

#PBS -l nodes=1:ppn=cores
#PBS -l walltime=1000:00:00
#PBS -q default
#PBS -j oe

cd $PBS_O_WORKDIR

#Set the scratch directory
SCRDIR=/scratch/$USER
mkdir -p $SCRDIR

######## ENTER YOUR JOB HERE ###############################################################
# In few more things you will see how to avoid using the full path
molpro -G 4000000000 -d $SCRDIR -n cores --no-xml-output name-input.com
############################################################################################

# Molpro erases what's inside the scratch by itself, how, well, check the manual
```

4 Few more things

4.1 Putting a program/library/compiler in your path

In your home directory there's a file called .bashrc, for any Program, Compiler, or Library (PCL) you need to use, you have to export it's path to yours. Open your .bashrc with the text editor of your choice and now add the lines:

```
export PCL_HOME=/path/to/pcl
export PATH=$PCL_HOME:$PATH
```

Example:

```
export MOLPRO_HOME=/opt/molpro/2012.1.41/molprop_2012_1_Linux_x86_64_i8/bin
export PATH=$MOLPRO_HOME:$PATH
```

Sometimes, you also need to execute a script to configure the program or compiler (happens with Turbomole and the Intel compilers), but this should give you a general idea of how to do it.