

My JASMIN User Experience

Tianran Zhang

1. Background

I am a second year PhD student at NCEO-King's College London, working on fire detection using satellite remote sensing data. For my project I need to deal with more than 20TB of remote sensing data, which poses a problem due to the long time it would take to download via FTP from the US National Oceanic and Atmospheric Administration (NOAA). It would take more than two months to download! So my supervisor suggested I try the JASMIN computing infrastructure at the Centre for Environmental Data Analysis (CEDA).

I have almost no experience of using a Linux system. It took me some time to get used to the non-graphical interface and learn how to open a file and download and process data with my Python code that I used on my windows machine. But over the past few weeks I have successfully achieved my goals with lots of help from people at CEDA. I hope to share some of my experience here with other beginners.

2. Logging onto the JASMIN system

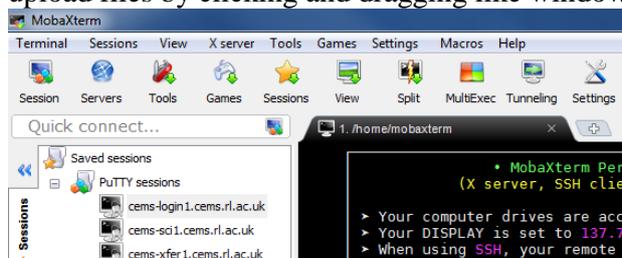
After applying for a user account for the JASMIN system I followed the steps described on the CEDA website (<http://jasmin.ac.uk/workflow/ssh-keys/ssh-key-windows/>) to generate my private and public keys using PuTTYgen, and submitted the public key to CEDA. Every time I login to JASMIN I just add my private key to Pageant and keep it running. Then with the account issued from CEDA, sth looks like xxx@cems-login1.cems.rl.ac.uk, and I can directly login to JASMIN through tools like Putty.

So I use PuTTYgen, Pageant and Putty to get access to the JASMIN system, which can all be downloaded from: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>.

2.2 First touch of Linux and some useful Windows system software

For a Windows user like me, the non-graphical interface of Linux system is quite a new experience. The CEDA website has a good summary of the basic commands in Linux (<http://jasmin.ac.uk/how-to-use-jasmin/basic-linux/>). The CEDA helpdesk (support@ceda.ac.uk) had a great suggestion for me: to replace Putty with a software called MobaXterm. The Personal Edition of MobaXterm is free to download and use (<http://mobaxterm.mobatek.net/download.html>).

I add the ssh sessions into the session surface and save them, or directly import from the saved sessions. Then every time I want to login to the JASMIN system I just double click on the session without having to type in my user name or host address. Having logged into the system using MobaXterm, the interface on the left automatically goes to SFTP, allowing me to download or upload files by clicking and dragging like windows.



Another windows-based software I find useful for managing lots of files is WinSCP, which has a similar interface to most FTP software. It allows me to easily check my files, move them from

one place to another, or download/upload files to/from the local machine. I need to login xxx@cems-xfer1.cems.rl.ac.uk to use it.

2.3 Downloading remote sensing data from NOAA FTP

As I already mentioned, I need to download a huge amount of data from NOAA's FTP service. I am hoping to take the advantage of JASMIN's super-fast broadband speed and its large data storage space. Once I have ordered remote sensing data from NOAA I get an email with the ftp server name and the file locations. Normally, I first login to my xxx@cems-xfer1.cems.rl.ac.uk account. Then I open the directory where I want to store the data, which is located in my group space. It looks something like:

```
$ cd /group_workspaces/xgroup/xxsubgroup/users/xxx/
```

Then I create a directory (\$ mkdir xxfile) and then open it (\$ cd xxfile). Then I login to NOAA's FTP server by typing:

```
$ ftp -i ftp.class.ngdc.noaa.gov
```

Here, ftp.class.ngdc.noaa.gov is the host I am using to download my data, and the '-i' keyword turns off the interactive mode, which will later force the system to automatically download all the files without asking my permission for each one.

Then I type in my name/password to get into the ftp server. If anything goes wrong, such as making a typing error, I just simply use the 'bye' command to close the ftp server connection, and link again.

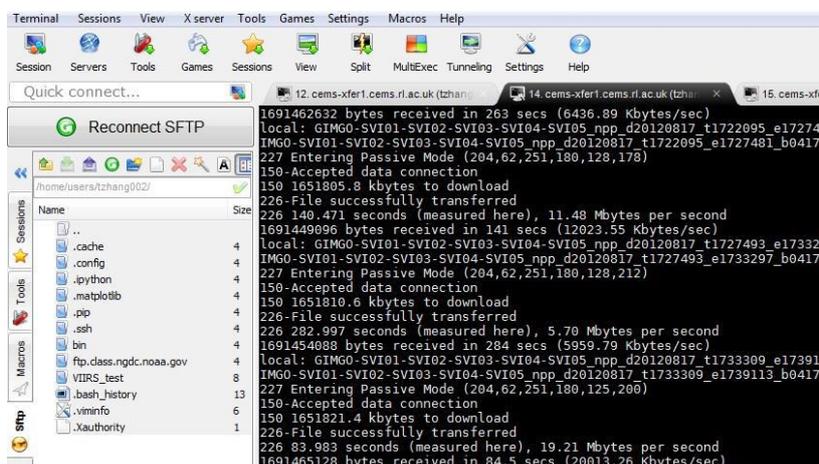
In the FTP server I open the directory where my target files are located. I then check the files by typing 'ls -l'. To copy all the files here to the folder I just created on the JASMIN system, I simply use:

```
$ mget *.*
```

Or if you just download some of the files (for example with 201507, 201508 within the file names), you can change the command to:

```
$ mget *201507*.* *201508*.*
```

Then you can marvel at JASMIN's download speed:



2.4 Running python code on JASMIN

To run python code I need to login to a Virtual Machine. Here I can use my account, the one like xxx@cems-login1.cems.rl.ac.uk, to login to the JASMIN system first, and I come to an interface that looks like this:

```
For support please contact CEDA Helpdesk: support@ceda.ac.uk
*****
** JASMIN Shared VM status at 2015-08-17 12:45:01.278295 **
*****
Average load on each VM over the last hour:
=====
Host                Users  Free memory  CPU
-----
jasmin-sci1.ceda.ac.uk  23    26.7G      40.0%
jasmin-sci2.ceda.ac.uk   8    27.5G      13.0%
cems-sci1.cems.rl.ac.uk  11    14.0G       0.0%
=====
-bash-4.1$ █
```

Then I can login to the Virtual Machine:

```
$ ssh xxx@jasmin-sci1.ceda.ac.uk
```

where I should be able to run my code.

To run the Python code, related packages need to be installed simply by following the FAQ 20 (<http://jasmin.ac.uk/faq/#python>).

To summarise, first I run virtualenv to get the inherited site packages, which only need to run the first time I install them:

```
$ virtualenv --system-site-packages $PATH_TO_VENV_DIRECTORY
```

Then I activate it, which needs to be done every time I run python code:

```
$ source $PATH_TO_VENV_DIRECTORY/bin/activate
```

To install/upgrade other packages in pip, I need to check the network, following FAQ 6.

After installing all the packages I need, I use the command:

```
$ python xxxx.py
```

To run my own python code. The only difference between my windows version of python code and the one I run on the VM machine is to add a head at the beginning of the code to let the Linux machine identify it:

```
#!/usr/bin/python2.7
```

Another tip for running python code on VM is take the advantage of its multi-core CPU by using multiprocessing (<https://docs.python.org/2/library/multiprocessing.html>).