

# Tips of Using Servers

Yanwei Fu

December 2, 2016

To connect the servers located in Fudan, you need the VPN which is downloaded from

<http://www.ecampus.fudan.edu.cn/2442/list.htm>

## 1 SSH in Linux or Mac OS X

To connect to the server from home you must connect.

To connect via ssh:

**\$ssh username@10.141.208.18**

you need to input the password.

Note that you need to connect the server using VPN if your computer is not connected within Fudan Internet.

### 1.1 Using Graphical Interface on the server

You should connect the server via ssh:

**\$ssh -X username@10.141.208.18**

then, try

**\$matlab**

or

**\$nautilus**

or

Note that

1. for mac users, you need to install X11: <https://www.xquartz.org>, since <https://support.apple.com/zh-cn/HT201341>
2. for linux users, you should be OK directly running that command;
3. for windows users, I am still googling...

## 2 Top command

try input:

**\$top**

top command lets you see how much CPU, memory etc are being used by your processes.

In top the following can help you at times.

- 1) To see number of CPUs on the machine: ctrl+l
- 2) To see only your processes: u followed by your username
- 3) To reorder rows by memory, cpu etc: O

You can also try

**\$man top**

to learn more about top;

you can also try

**\$htop.**

### 3 Copying data crossing servers

you need to copy the data from you local machine to server; or inverse direction.

Then, you need scp command

**\$scp source-file username@10.141.208.18:/home/username/target-file**

If you are copying the folder from one machine to another machine, just using

**\$scp -r source-file-folder username@10.141.208.18:/home/username/target-file-folder**

here -r, is for recursively copying.

### 4 Using screen

Running your experiments using screens allows you to close your terminal window without stopping your experiments.

- 1) Start a new screen using command: `screen -S <name> -t <name_of_tab>`
- 2) Run your program (like matlab) in the screen.
- 3) Detach from the screen using command: `Ctrl+a+d` ,

or,

`screen -d <name>`

- 4) At this point your program is still running. If you want to connect back to the screen to see your program output.

Use command: `screen -r <name>`

Some other commands that may be of help are:

- 1) To see a list of screen you have created, use command: `screen -ls`
- 2) To remotely detach an attached screen, use command `screen -d <name>`

### 5 Using Anaconda and Matconvnet

Xuelin had installed Anaconda on the servers. You can try to add it to your own path. you do as what instructed in Fig. 1.

```
1、命令行输入 vim ~/.bashrc
2、按'i'在最后输入下面两条命令

export PATH=/usr/local/Anaconda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/Anaconda/lib:$LD_LIBRARY_PATH

3、'ESC' + ':' + 'wq' + 'ENTER'    (单引号里为要键入的字母或按键)
4、命令行输入 source ~/.bashrc
```

Figure 1: How to add anaconda to your path.

## 6 Running Programs

We share the servers among all researchers as a result we want to ensure that our programs don't take up too much resources and slows the servers down to a crawl. Therefore when you start matlab or other programs it is advisable to use the nice command.

nice +PRIORITY matlab

Where PRIORITY is a number greater than 0. If you don't use nice, the program by default gets a PRIORITY of 0.

For Matlab users, some programs you write might take a number of CPUs to execute. This may not be fair since you are sharing the servers with others. You can control the CPU usage by simply putting this command at the beginning of your Matlab script:

maxNumCompThreads(<num\_cpu>)