

服务器使用指南

刘雷轶男 智朝晖

2023 年 5 月 17 日

摘要

这是一篇介绍组内服务器如何使用的指南。本文的源代码放在了Overleaf上，有错误或过时的信息请大家及时更新。如果有不详尽的地方，请及时指出。希望大家多提建议。

目录

1	服务器基本信息	3
2	如何连接服务器	3
2.1	MacOS 系统	3
2.2	Windows 系统	4
3	Linux 系统的基本操作	5
4	文件管理与传输	6
4.1	MacOS 系统	6
4.2	Windows 系统	7
5	Matlab 使用指南	7
6	Python 使用指南	8
6.1	导入 conda 路径	8
6.2	运行 python 文件	9
6.3	注意事项	9
6.4	终止 python 程序	10
7	Julia 使用指南	10
7.1	运行 Julia	10
7.2	将 Julia 链接到 MKL 库上	10
8	使用 GPU 加速计算	11
9	硬盘使用	11

1 服务器基本信息

目前组内一共有四台服务器，它们都配备了双路 6238R，Centos7 系统，基本信息如下所示，其中编号 2 配有 GPU。

编号	所在楼层	IP 地址	端口	配置的软件	管理人	内存大小
1	8 楼	172.17.176.141	88	Matlab, Python	崔健	188T
2	8 楼	172.17.176.141	2	Matlab, Python	崔健	1.5T
3	8 楼	172.17.177.238	22	Matlab, Python	王京京	188G
4	6 楼	172.17.162.3	22	Matlab, Python, Julia	刘雷轶男	754G

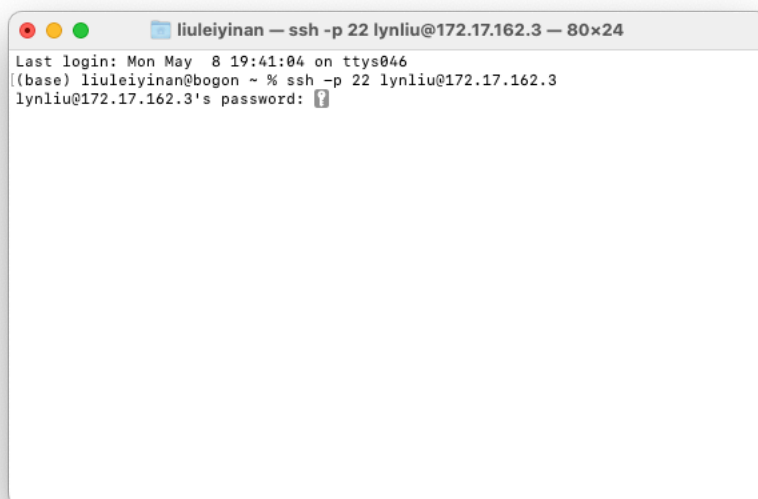
目前服务器的管理员是刘雷轶男。如果对在 Mac OS 系统上使用服务器，或对 linux 系统，亦或是在服务器上使用 Python 有疑问，可以找刘雷轶男¹讨论。如果对在 Windows 系统上使用服务器，或者是对服务器上 Matlab、Julia 使用有疑问，可以找智朝晖²讨论。

2 如何连接服务器

2.1 MacOS 系统

对于 MacOS 系统，可以通过以下方式连接服务器：


1. 同时按一下 command 键和空格键打开聚焦搜索，输入 terminal，按回车，然后会出现终端界面
2. 输入 `ssh -p 端口号 用户名@IP地址` 后回车



¹微信: liuleiyinan_buaa

²微信: z2276780832

3. 输入密码后，回车，便成功登录上了服务器



```
liuleiyinan — lynliu@localhost:~ — ssh -p 22 lynliu@172.17.162.3 — 80x24
Last login: Mon May  8 19:48:05 on ttys046
((base) liuleiyinan@bogon ~ % ssh -p 22 lynliu@172.17.162.3
lynliu@172.17.162.3's password:
Last login: Mon May  8 19:47:44 2023 from 10.23.218.101
(base) [lynliu@localhost ~]$
```

2.2 Windows 系统

对于 Windows 系统，可以通过和上文 MacOS 系统一样，通过 ssh 命令连接服务器，但这里我们更推荐用具有 ssh 服务同时还可以进行文件管理和传输的软件，比如 Xshell 和 MobaXterm。下面我们 MobaXterm 为例，其官网为<https://mobaxterm.mobatek.net/>:

1. 在左上角部分菜单栏选择 Session，即建立一个联系/对话。如图1



图 1: MobaXterm 选择页面

2. 选择 SSH 连接方式，在 Basic SSH settings 部分填入远程 ip(内网 ip 或者是校园网 ip)，然后填入用户名，选择相应端口，（具体信息参见1），点 OK 就可以了。

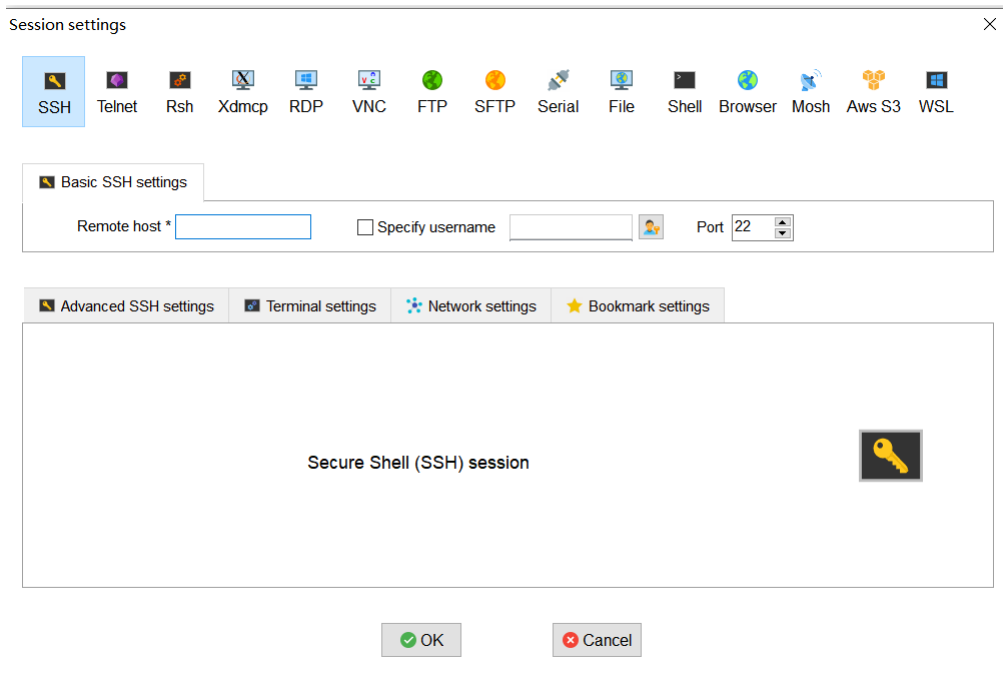
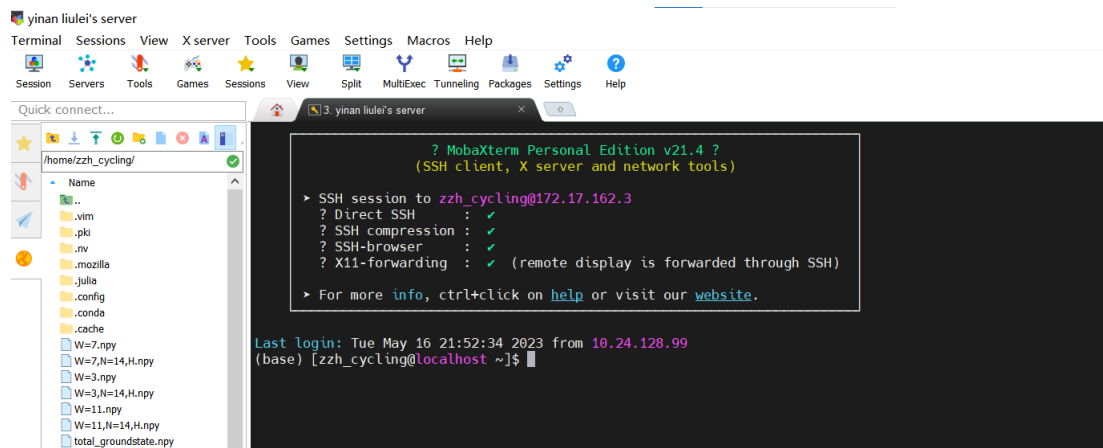


图 2: 新建 ssh 会话

3. 会显示login as:，输入用户名后，敲回车即成功进入。



3 Linux 系统的基本操作

常用的命令，包括：

- 切换目录 `cd`；切换到上一级目录 `cd ..`；回到当前账号的工作目录 `cd ~`
- 查看目录 `ls`；列出文件详细信息 `ls -l`；列出当前目录下所有文件（包括隐藏的）`ls -a`

- 创建目录 `mkdir`
- 查看文件 `cat`
- 删除文件 `rm` ; 删除目录 `rm -r` ; 强制删除 `rm -f`
- 显示当前目录 `pwd`
- 动态显示当前耗费资源最多进程信息 `top`
- 显示瞬间进程状态 `ps` , 比如查看所有 `python` 程序的运行情况

```

(base) [lynliu@localhost home]$ ps -aux | grep python
root      2875  0.0  0.0 363816 38812 ?        Ssl  2022   1:07 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid
root      2764  0.0  0.0  574288 15520 ?        Ssl  2022  38:17 /usr/bin/python2 -Es /usr/sbin/tuned -l -p
lynliu    154767 0.0  0.0  112828   980 pts/0    S+   20:10   0:00 grep --color=auto python
zzh_cyc+  283916 397  2.6 24547968 18598460 ?      Rl   3月27 241662:42 python Rydbergssp.py
lynliu    278654 393  0.1  997748 398924 ?        Rl   2月28 436421:57 python -u crab_init1.py
lynliu    278755 395  0.0  922164 288680 ?        Rl   2月28 437661:04 python -u crab_init2.py
lynliu    278768 394  0.0  985764 387776 ?        Rl   2月28 437582:41 python -u crab_init3.py
lynliu    278764 394  0.1  998192 402540 ?        Rl   2月28 436639:25 python -u crab_init4.py
(base) [lynliu@localhost home]$
  
```

图 3: 查看所有 `python` 程序的运行情况, `grep` 表示筛选有 `python` 关键字的进程名

- 命令帮助 `man` , 比如`grep`命令不会用, 输入`man grep`看文档
- 清屏 `clear`
- 杀死进程 `kill`
- 查看硬盘使用情况 `df -h`
- 移动文件位置 `mv /xxx/xxx /xxx/`

4 文件管理与传输

4.1 MacOS 系统

对于 MacOS 系统, 可以使用 Cyberduck 等其他远程文件管理软件。Cyberduck 的官网下载地址为<https://cyberduck.io/download/>。

下面介绍 Cyberduck 如何使用。

1. 在书签页面, 右键点击, 选择新建书签

2. 如下图一样配置即可，**第一栏一定要选成 SFTP**，名称和备注名可任选，服务器、端口按上面的表填写即可。



图 4: 配置书签

3. 创建完成后，双击创建的书签，就可以连接到服务器了，后续的操作都是可视化的。

4.2 Windows 系统

Windows 系统下的远程文件管理和传输也可以用 Xshell 和 MobaXterm，这里接着以 MobaXterm 为例：

在 MobaXterm 里上传与下载文件十分方便，只需要在本地和 MobaXterm 页面之间进行拖拽即可，或者采用鼠标右键选择相应的下载功能。

5 Matlab 使用指南

如果用 MobaXterm 连接服务器的话，直接在命令行输入 matlab，然后根据 xmanager 工具可以在本地打开 matlab 的界面工具，实现本地直接编写 matlab 代码并运行。

如果想用命令行操作的话，假设文件名为 matlabfile.m，进入 m 文件所在目录后，运行

```
matlab -nodesktop -nosplash -rmatlabfile
```

6 Python 使用指南

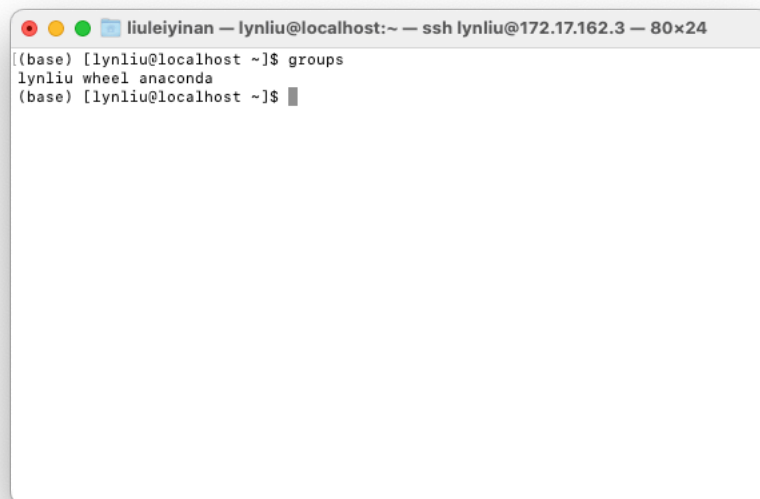
所有的服务器上都装有 anaconda, python 的版本都在 3.9 及以上, 并且 base 环境装了常用的库, 包括:

- numpy: <https://numpy.org>
- scipy: <https://scipy.org>
- matplotlib: <https://matplotlib.org>
- qutip: <https://qutip.org>

如果需要可以自行安装其他的包, 如果安装失败请联系管理员安装。由于只有root权限才可以新建环境, 如果需要创建新环境也需要联系管理员。

6.1 导入 conda 路径

在拿到账号密码后, 要和管理员确认是否将该账号加入了 anaconda 的用户组, 如果不在该用户组里, 以下的操作是无效的。可以使用groups命令查看目前账号是否在 anaconda 用户组中。



```
liuleiyinan — lynliu@localhost:~ — ssh lynliu@172.17.162.3 — 80x24
(base) [lynliu@localhost ~]$ groups
lynliu wheel anaconda
(base) [lynliu@localhost ~]$
```

图 5: 如果在 anaconda 用户组里, 会出现 anaconda 名称

确认之后, 登录账号, 输入`cd ~`, 确保当前在该账号的主目录下, 更改`.bashrc`文件的配置。将`.bashrc`文件的内容改为


```

if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi
__conda_setup="$('/usr/local/anaconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/usr/local/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/usr/local/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/usr/local/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup

```

或者在本地创建一个包含上述内容的 txt 文件，通过 Cyberduck、Xshell 等工具上传到服务器上再改名为 `.bashrc` 也可。修改完成后，输入 `logout` 退出登录，再次登录时，就可以直接使用 `conda` 和 `python` 命令了。

更改完后可以使用 `cat .bashrc` 查看是否更改成功。

6.2 运行 python 文件

将 `python` 上传到服务器上后，进入该文件所在目录，输入 `python xxx.py` 即可运行该程序。

如果程序运行时间较长，想让程序在退出登录后继续运行，可以使用 `nohup` 命令

```
nohup python -u xxx.py >output.log 2>&1 &
```

其中 `xxx.py` 是所运行的文件，`output.log` 是 `xxx.py` 运行时输出内容的存放文件。如果 `xxx.py` 出现错误，或者意外退出，信息都会保存在 `output.log` 中。

服务器可以一次提交多个 `python` 程序。

6.3 注意事项

`Python` 程序本身是只能单核运行的，但是如果使用 `numpy` 和 `scipy` 库，此时它们调用的 `mkl` 会多线程运行，从而占据许多 CPU 资源，并且此时的程序运行效率大大下降。因此，在服务器上运行 `python` 程序时，务必在运行的 `python` 脚本开头加上如下代码

```

import os
os.environ["MKL_NUM_THREADS"] = "2"

```

这样，程序的运行就不会有很高的 CPU 占用率了，如果设置为“2”，则最高应只有 200% 的占用率。此处的具体数字可以根据程序的运算量灵活设置，最好不要一次性占用太多 CPU 资源。

6.4 终止 python 程序

如果想要中断某个 python 程序的运行，可以使用

```
ps -aux | grep python
```

列出所有的 python 程序，找到你想终止的程序的 PID 后，输入

```
kill xxxxxxxx
```

回车即可。(xxxxxxx表示某一 PID)

7 Julia 使用指南

由于 Julia 仍属于十分小众的语言，所以只在一台服务器上安装了 Julia（具体见服务器基本信息），版本号是1.8.5。如果要使用 Julia，请在.bashrc文件中加入以下路径：

```
export PATH="/data/julia-1.8.5/bin:$PATH"
```

更改.bashrc文件后，重新登录账号或者用source .bashrc刷新一下后，就可以用julia 直接进入 Julia REPL 了。

7.1 运行 Julia

运行 Julia 程序的步骤与 python 是类似的

```
nohup julia xxx.jl >output.log 2>&1 &
```

即可在后台持续运行xxx.jl程序，并将输出或报错信息记录在output.log中。服务器可以一次提交多个 Julia 程序。

7.2 将 Julia 链接到 MKL 库上

因为 Julia 默认是链接到 OpenBLAS 库上，而在 Intel 芯片上 MKL 运行更有效率更高效，所以你可以选择将 Julia 的 LinearAlgebra 的线性代数库换成 MKL，操作如下：

1. 进入 Julia REPL 后，`using Pkg; Pkg.add("MKL")`
2. 然后在程序导入库的**第一行**写上：`using MKL`，就能愉快地用 MKL 来计算了。
(虽然这时候你输入`using LinearAlgebra`；然后 `LinearAlgebra.BLAS.get_config()`；或者`LinearAlgebra.BLAS.vendor()`还是显示的 OpenBLAS)

8 使用 GPU 加速计算

使用 GPU 加速计算基本都是利用商业公司已经写好的计算架构和 GPU 计算库来编程，使用 GPU 加速计算有两种方法：

1. 直接使用 CUDA 的 C/C++ 版本进行编程。
2. Python 使用 Numba 库调用 CUDA，或者使用 cupy，或者是 Pytorch 等已经生态很良好的深度学习框架来编写程序。

9 硬盘使用

如果有很大的文件需要储存，建议移动到硬盘中储存。目前一般在 `/data` 或 `/data1` 下储存大文件。下面是操作步骤。（如果有服务器没有开启权限，请联系管理员。）

1. 如果需要将文件存到硬盘中，需先创建一个以自己用户名为文件名的文件夹。切换到 `/data` 路径下：
`cd /data`，然后创建文件夹：`mkdir 用户名`。
2. 将文件移动到该路径下：

`mv 原先文件路径 新的路径`

比如：

`mv /home/lynliu/xxx.txt /data/lynliu/`