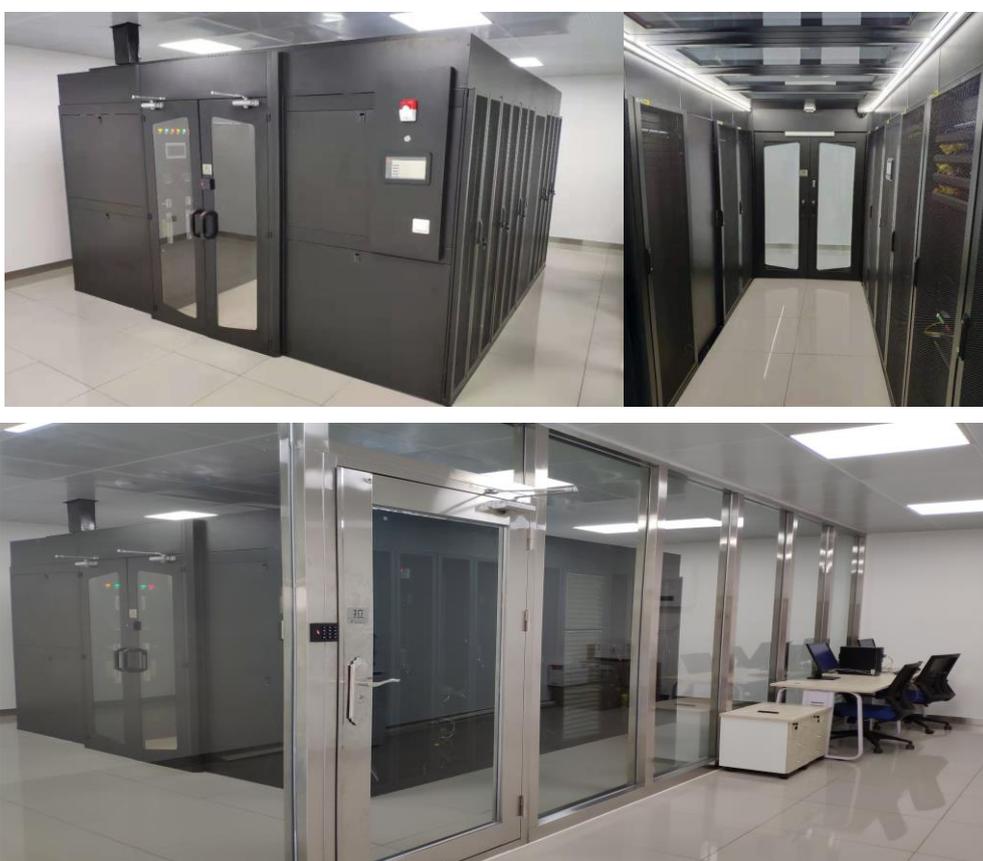


“凤凰超算平台” 使用手册

PHOENIX SUPERCOMPUTING PLATFORM

User Manual

版本 1.0



大样本恒星演化研究团组和恒星物理研究团组共同管理

1、帐号申请和使用

帐号的申请见帐号申请表。

2、登录、退出系统连接

2.1 登录方式：

Linux 用户，使用命令登录：`ssh <username>@192.168.79.3`

例如：用户 A 申请到超算平台账号用户名为 phoenix，密码为 123456。在终端中输入 `ssh phoenix@192.168.79.3`，回车后输入密码 123456 即可登陆平台。

2.2 退出系统连接：

ssh 访问必需使用 `exit` 命令一到多次确保最终结束连接。

2.3 文件上传与下载（目前适用 Linux 用户）：

① 从本地机子上上传数据到超算平台(Linux 系统终端输入)：

`scp -r <本地文件或文件夹> <username>@192.168.79.3:<用户目录>`

例如，用户 A 在超算平台上的账号用户名为 phoenix，密码为 123456。用户 A 将存储在本本地计算机/home/data/目录下的 code 文件夹上传到超算平台上，并储存到超算平台自己目录下 (/home/phoenix/)，只需在终端输入以下命令，并输入用户密码确认即可：

```
scp -r /home/data/code/ phoenix @192.168.79.3:/home/phoenix/
```

② 从超算平台下载数据到本地：

```
scp -r <username>@192.168.79.3: <用户文件或文件夹下需下载的文件  
或文件夹> <本机保存位置>
```

例如，用户 A 超算平台账号用户名为 phoenix，密码为 123456。用户 A 想将存储在超算平台 /data/phoenix/ 目录下的 code 文件夹下载到本地计算机/home/目录下：

```
scp -r phoenix@192.168.79.3:/data/phoenix/code /home/data/
```

3、存储空间使用

① 用户自申请开通“凤凰超算平台”个人账号开始，即默认同意签署该超算平台提供的“系统每个月 1 号定时自动删除临时存储空间内个人数据”服务的免责协议。

② 超算平台为每个用户提供长期存储空间（目录为：/home/<用户名>/）和临时存储空间（目录为：/data/<用户名>/）。

例如：用户 A 超算平台账号用户名为 phoenix；

其长期存储空间目录为/home/phoenix /；

其临时存储空间目录为/data/phoenix/;

说明:

- a. 用户登陆超算平台，默认登陆到用户长期存储空间的用户目录;
- b. 用户在/home/或者 data/的主目录下不能自己创建文件夹;
- c. 但是在自己的用户目录里可以随意创建文件夹。

③ 长期存储 (/home/)：每个用户在长期存储空间最多可以保存 1T 大小的数据；该长期存储空间内的所有文件从用户开通账号开始到用户注销前可一直保存。

④ 短期临时存储 (/data/)：不对用户在临时存储空间的可用大小进行限制。但是，每月 1 号 00 时 00 分开始，系统将自动删除所有用户储存在临时存储空间内的所有数据（**所有用户开通账号即默认同意该临时存储中的数据会被系统定时删除的免责声明!**）。

4、软件安装与使用

4.1 平台已安装软件:

超算平台已经安装了用户需求的大部分编译器和软件并通过 Environment Modules (<http://modules.sourceforge.net/>) 对这些软件进行管理:

- ① 包括 gnu、intel 以及 pgi 三类编译器；
- ② blas、fftw、hdf5、hypre、lapack、scalapack 以及 mkl 软件库；
- ③ anaconda、ffmpeg、matlab、nmap 以及 pgplot 核心包。

4.2 用户自行安装软件：

用户可以在超算平台上自行安装所需其它软件；但是，安装软件必须指定路径安装（可以根据不同的软件，百度具体操作命令），仅允许安装在用户自己的长期存储空间（即，`/home/<username>/`）或临时存储空间内（即，`/data/<username>/`）的用户目录下。

4.3 平台已有软件的使用方法：

用户可在登陆后使用 `module avail` 命令可以查看超算平台提供的所有软件信息。

例如：用户 test 在终端中输入 `module avail` 后会显示如下：

```
[test@mu01 ~]$ module avail
```

```
----- /opt/software/modulefiles/compilers -----
```

```
gnu/gcc-3.4.6_g77-f77      gnu/mvapich2-2.2      intel/2015
gnu/gcc-7.2.0             gnu/openmpi-1.10.6   intel/2018
gnu/mpich2-1.5            gnu/openmpi-3.1.2    pgi/v10.6
gnu/mpich-3.2             gnu/openmpi-4.0.0
```

```
----- /opt/software/modulefiles/libraries-gnu -----
```

```
blas-gcc4.8.5-gfortran/3.8.0      hdf5-mvapich2-2.2/1.10.5-parallel
fftw-mpich3.2/2.1.5-parallel      hdf5-mvapich2-2.2/1.10.5-serial
fftw-mpich3.2/3.3.6-parallel      hdf5-mvapich2-2.2/1.6.9-parallel
```

fftw-mvapich2-2.2/2.1.5-parallel	hdf5-mvapich2-2.2/1.8.20-parallel
fftw-mvapich2-2.2/3.3.6-parallel	hdf5-mvapich2-2.2/1.8.20-serial
fftw-openmpi4.0.0/2.1.5-parallel	hdf5-openmpi3.1.2/1.6.9-parallel
fftw-openmpi4.0.0/3.3.6-parallel	hdf5-openmpi3.1.2/1.8.20-parallel
hdf5-gcc4.8.5/1.10.5-serial	hdf5-openmpi4.0.0/1.10.5-parallel
hdf5-gcc4.8.5/1.6.9-serial	hdf5-openmpi4.0.0/1.10.5-serial
hdf5-gcc4.8.5/1.8.20-serial	hdf5-openmpi4.0.0/1.8.20-serial
hdf5-mpich3.2/1.10.5-parallel	hypr-gcc4.8.5/2.14
hdf5-mpich3.2/1.10.5-serial	hypr-mpich3.2/2.14
hdf5-mpich3.2/1.6.9-parallel	hypr-mvapich2-2.2/2.14
hdf5-mpich3.2/1.8.20-parallel	hypr-openmpi4.0.0/2.14
hdf5-mpich3.2/1.8.20-serial	lapack-gcc4.8.5-gfortran/3.8.0

----- /opt/software/modulefiles/libraries-intel -----

blas-intel2018-ifort/3.8.0	hdf5-intel2018/1.8.20-parallel
fftw-intel2018/2.1.5-parallel	hdf5-intel2018/1.8.20-serial
fftw-intel2018/3.3.6-parallel	hypr-intel2018/2.14
hdf5-intel2018/1.10.5-parallel	lapack-intel2018-ifort/3.8.0
hdf5-intel2018/1.10.5-serial	mkl/intel2015
hdf5-intel2018/1.6.9-parallel	mkl/intel2018
hdf5-intel2018/1.6.9-serial	scalapack-intel2018/2.0.2

----- /opt/software/modulefiles/core-packages -----

anaconda/2-2019.07-python2.7	ffmpeg/4.1.3
anaconda/2-4.1.1-python2.7	matlab/R2018a
anaconda/3-2019.07-python3.7	nmap/7.70
anaconda/3-4.3.0-python3.6	pgplot/5.2-gfortran

① 命名和编译器的命名方式：

<哪类编译器>/<编译器及版本>

例如：intel/2018 为 2018 版的 intel 编译器

② 软件库的命名方式：

<软件名>-<使用哪种编译器对该软件进行编译>/<软件版本>-<编译时是否支持了并行>。

例如：

/opt/software/modulefiles/libraries-gnu 下的 fftw-mpich3.2/2.1.5-parallel 为使用 gnu/mpich-3.2 编译器编译的 fftw，该 fftw 的版本为 2.1.5，并且编译时支持了并行。其中，软件库又根据编译时使用的是 gnu 编译器（或者 intel 编译器）进行了分类。

③ 核心包的命名方式： <核心包名称>/<软件包的版本>

例如： matlab/R2018a 为 R2018a 版本的 matlab

④ 用户根据自己的需求，可以通过 module load 和 module unload 命令对平台已经安装的软件进行调用及卸载：

⑤ 软件调用： module load <软件名称>， 软件名称即为通过以上的 module avail 命令后看到的具体软件名称

例如，用户 test 需要调用 intel 2018 版本的编译器，用户首先通过上述 module avail 命令获得了 intel 2018 版编译器的名称为 intel/2018，调用只需： module load intel/2018

⑥ 软件卸用： module unload <软件名称>

例如，用户 test 需要卸载已经调用的 intel 2018 版本的编译器：

module unload intel/2018

⑦ 采用 `module list` 可以查看用户自己已经导入的所有软件。

5、系统状态查询

查询作业

命令	介绍
<code>qstat</code>	查看本用户提交的任务
<code>qstat -an</code>	同上，输出内容稍有不同，包含有此作业运行机器名称等

查看作业队列

命令	介绍
<code>qstat -q</code>	系统中所有的队列，以及每个队列中任务的运行和等候情况

查看运行中任务

命令	介绍
<code>showq</code>	查看系统中所有运行的任务

删除作业

命令	介绍
<code>qdel id</code>	删除 JOBNAME 为 id 的任务

查看计算节点运行状态

命令	介绍
<code>pbsnodes -l all</code>	列出所有计算节点的运行状态

5.1 作业管理

超算平台通过“PBS 作业管理系统”对平台上用户的所有作业操作进行管理和分配，用户必须通过编辑 PBS 脚本文件来进行作业的提交和管理。

注意：

- a. 用户使用超算平台必须一律通过 PBS 作业管理系统来进行作业的提交和管理；
- b. 用户不能以任何绕过作业管理系统的方式使用超算平台（包括测试程序）；
- c. 用户应尽量避免使用队列、指定节点等方式提交作业。（更多使用规范详见《超算平台规章制度》）

超算平台共有 31 个计算节点，其中 30 个普通计算节点（cu01-cu30），1 个胖节点（fat01）。为了更加合理利用计算资源，超算平台采用队列的方式对计算节点进行划分：

- ① 30 个普通计算节点作为一个队列，队列名称为 batch；
- ② 1 个胖节点单独作为一个队列，队列名称为 batchfat。

说明：鼓励处理观测数据的用户或者进行单核心（串行）任务计算的用户，通过在 PBS 脚本中区分队列的方式将计算任务提交到胖节点（任务脚本见后）。

5.2 PBS 脚本模板：

5.2.1 PBS 脚本文件由脚本选项和运行脚本两部分组成：

脚本选项：PBS 作业脚本选项是以 ‘#PBS’ 开头的区域（**注意** ‘#PBS’ 中的#并非注释），后面添加不同选项参数。

例如，用户 test 创建编辑 job.pbs 文件如下：

```
[test@mu01 ~]$ vim job.pbs
```

编辑内容如下

```
#PBS -N MESA_test
#PBS -l nodes=3:ppn=36
#PBS -q batch
#PBS -V
#PBS -S /bin/bash
```

参数描述如下：

- ① -N 作业名称 设定作业名称
- ② -l 设定作业所需资源 nodes 为设置作业所需节点及每个节点所需核心数。使用方式：nodes=<节点数目>:ppn=<每个节点调用多少个核心数>。上述示例调用 3 个计算节点，每个计算节点使用 36 个核心。

- ③ -q 队列名称 设定作业队列名称
- ④ -V 表明 qsub 命令的所有环境变量都 export 到此作业
- ⑤ -S 表明执行运行脚本所用的 shell，须包含全路径

5.2.2 运行脚本根据“串行”或“并行”的需求有所不同。

以下，提供超算平台 3 种提交作业的 PBS 脚本模板。用户可根据任务需求，选择相应的模板进行调整。（模板中的括号中蓝色或者红色字体的内容为相应参数和命令的解释）

(1) 串行任务模板（serial.pbs，建议提交到胖节点）：

```
#PBS -q batchfat          (batchfat 为 PBS 队列名称，用户  
暂无需修改)  
  
#PBS -N jobname          (“jobname” 用户自己作业的名称，  
作为用户识别的标志，用户可根据需要自行修改)  
  
#PBS -V                  (该行不用修改)  
  
#PBS -S /bin/bash        (该行不用修改)  
  
module load <运行程序所需的软件名> (加载程序所需的软件  
和函数库。如果没有去掉这行)  
  
cd /home/test/run (此处修改为用户可执行程序文件所在目  
录，必须修改)  
  
date
```

./evol > out (“evol”为用户可执行程序文件名,必须修改)

(2) 并行单节点任务模板 (job.pbs, 建议提交到胖节点上):

#PBS -q batchfat (队列可选为 batch 或 batchfat)

#PBS -N jobname (用户可自定义)

#PBS -l nodes=1:ppn=18 (不需改节点 nodes 数目,只需更改每个节点所需核心数,当前设置为 18 计算核心)

#PBS -V (该行不用修改)

#PBS -S /bin/bash (该行不用修改)

###module load <运行程序所需的软件名> (加载程序所需的软件和函数库。如果没有去掉这行。MESA 无需加入)

export OMP_NUM_THREADS=18 (根据用户上面设置的总核心数调整)

cd \$PBS_O_WORKDIR (进入工作目录。注意:请在所需执行的程序的目录下提交作业,否则用户需自行指定程序所在目录,如下)

cd /home/test/MESA (此处修改为用户可执行程序文件所在目录,必须修改)

EXEC='./rn' (“rn”为可执行文件,根据用户自己程序的可执行文件修改)

date

\$EXEC

(3) 并行多节点任务模板 (mpi.pbs):

#PBS -q batch (“batch” 为普通计算节点队列名称，用户无需修改)

#PBS -N jobname (用户可自己命名作业)

#PBS -l nodes=3:ppn=18 (改节点数以及每个所需核心数，当前为使用 3 个节点，每个节点使用 18 个核心，一共使用 54 个核心进行计算)

#PBS -V (该行不用修改)

#PBS -S /bin/bash (该行不用修改)

module load <运行程序所需的软件名或者函数库> (加载程序所需的软件及程序所需设置)

cd \$PBS_O_WORKDIR (进入工作目录。注意：请在所需执行的程序的目录下提交作业，否则用户需自行指定程序所在目录)

mpdboot -n \$NN -f /tmp/nodes.\$\$ -r rsh

date

mpirun -machinefile \$PBS_O_WORKDIR/nodefile -np

\$NP ./WIND parameter.file (“WIND” 为程序可执行文件，必须修改; “parameterfile” 为程序运行需要读入的初始参数文件，没有的话无需加。当前展示了 mpirun 的使用方法; mpiexec 需要调整，用户需要可以咨询)

date

```
mpdallexit  
rm -rf $PBS_O_WORKDIR/nodefile  
rm -fr /tmp/nodes.$$  
rm -fr /tmp/nodefile.$$
```

PBS 脚本编辑完成后，使用 `qsub` 命令进行作业提交。使用方式：`qsub <PBS 脚本>`。例如：提交单核心任务，使用 `serial.pbs` 脚本的命令如下：`qsub job.pbs`

6、科研产出中致谢:

① 中文：感谢中国科学院云南天文台“大样本恒星演化研究团组”和“恒星物理研究团组”共同管理的“凤凰超算平台”提供的机时。

② 英文：The authors gratefully acknowledge the “PHOENIX Supercomputing Platform” jointly operated by the Binary Population Synthesis Group and the Stellar Astrophysics Group at Yunnan Observatories, Chinese Academy of Sciences.