

高性能计算平台二期平台 使用手册

南京航空航天大学高性能计算中心

2022 年 8 月

目 录

1 平台资源概述.....	1
1.1 二期平台集群.....	1
1.2 应用软件资源.....	1
2 使用流程及规范.....	2
2.1 使用流程.....	2
2.2 使用规范.....	5
3 快速入门.....	6
4 登录集群和文件上传下载.....	7
4.1 平台管理系统注册.....	7
4.2 Windows 系统.....	9
4.3 Linux/Mac 系统.....	10
4.4 校外用户登录 VPN.....	11
4.5 Windows 系统文件上传下载.....	12
4.6 类 Unix 系统文件上传下载.....	13
5 提交作业.....	15
5.1 Slurm 作业调度系统使用说明.....	15
5.1.1 sbatch 提交作业.....	15
5.1.2 salloc 交互式运行作业.....	17
5.1.3 sinfo 查看资源空闲状态.....	18
5.1.4 squeue 查看作业队列.....	18
5.1.5 scancel 取消作业.....	19
5.2 常用科学计算软件使用.....	19
5.2.1 GNU 编译器套件.....	19
5.2.2 Gromacs.....	21
5.2.3 vasp.....	22
5.3 可视化作业.....	24

5.4 Windows 作业	28
6 常见问题及注意事项	32
6.1 xshell 工具在哪下载?	32
6.2 我想用的计算软件没有怎么办?	32
6.3 我要用的计算软件作业脚本不会写怎么办?	32
6.4 提交作业报错如下错误.....	32
6.5 作业没有运行, 并且显示 QOSGrpCpuLimit.....	32
6.6 怎么修改账号密码	32
附录 1 Linux 常用命令	33
附录 2 常用脚本	34
附录 3 modulefile (使用场景)	36

1 平台资源概述

南京航空航天大学高性能计算平台（以下简称平台）是依托工信部十三五信息化专项--南京航空航天大学智慧校园建设的校级科学仪器公共平台，平台包含了一期平台集群、二期平台集群、人工智能集群三部分，可提供 CPU 计算和 GPU 计算所需资源。

1.1 二期平台集群

高性能计算平台二期平台集群硬件资源如下表所示，用户需要根据算例情况和计算需求确定 CPU 核数等参数编写脚本才能提交作业。

表 1 二期平台计算集群配置

队列名	CPU 类型	GPU 类型	内存	节点数
cpu	2 颗 Intel®Xeon®Platinum 8358 Processor, 2.6GHz, 32 核	/	256G	256

表 2 二期平台存储配置

型号	主要规格	套数
联想 DSS-G	硬盘 334*6T HDD 2*800G SSD 千兆网口 8*10G IB 网口 8*100GB InfiniBand 端口	1
存储容量合计：2PB 读写带宽 20GB/s		

1.2 应用软件资源

平台上已经安装常用应用软件，所在目录为/fs0/software，用户可进入该目录查看可用软件，编译或运行程序需设置正确环境变量。此外，附录 1 提供了常用的 Linux 系统命令，用户可在命令行界面登陆成功后参考并操作，附录 2 提供 vasp、lammmps、fluent、lsdyna 脚本样本，用户可以直接复制使用脚本样本，附录 3 提供了 module 常见的指令和几个 module 的使用案例，并演示了如何编写 modulefile 来管理自己的软件环境，方便加载应用软件所需的环境变量。

针对需要使用 python 的用户特别说明：集群无法连接公网下载第三方模块，集群提供了 anaconda3 环境供用户使用，可使用 module load 命令加载 anaconda3 的环境。如 anaconda3 中缺少第三方模块，请联系管理员协助安装所需的第三方模块。

2 使用流程及规范

2.1 使用流程

1) 开户：用户在网上办事大厅办理《高性能计算平台用户申请》。通过审批后由平台管理员开户，开户通过后，用户使用统一身份认证账号登录高性能计算二期平台管理系统（登录地址为 <https://hpcpt2.nuaa.edu.cn>），自行设置二期平台节点登录密码（详见 4.1 节），该密码用于登录集群计算节点提交作业。



图 1 用户平台申请

用户需输入个人邮箱用于接收开户信息和通知信息，点击提交按钮开始申请流程。

高性能计算平台用户申请

申请人信息			
姓名	<input type="text"/>	单位	<input type="text"/> * 此项必
学/工号	<input type="text"/>	邮箱	<input type="text"/>
固定电话	<input type="text"/>	移动电话	<input type="text"/>
是否为项目负责人		<input checked="" type="radio"/> 是 <input type="radio"/> 否	
项目信息			
项目名称	<input type="text"/>	项目编号	<input type="text"/>
所需软件及版本	<input type="text"/>		
项目简介	<input type="text"/>		
集群类型			
开户集群	<input type="checkbox"/> CPU1 集群 <input checked="" type="checkbox"/> CPU2 集群 <input type="checkbox"/> GPU 集群		
<input type="checkbox"/> 本人同意遵守《南京航空航天大学高性能计算平台管理条例（试行）》			
<input type="checkbox"/> 本人承诺，计算内容不涉及国家秘密信息，如有违反，愿承担相应责任。			

图 2 申请页面

2) 签订服务协议：未签订协议的用户，登录智慧门户“高性能计算平台用户技术服务协议签订”流程，填写项目内容、所需计算资源等信息，通过电子签名签章完成协议签订。学生用户在明确绑定的教师账户后无需签署技术服务协议。

3) 缴费: 计算任务完成后, 用户应根据使用情况缴纳计算费用。高性能计算中心定期向未完成缴费的用户发送缴费提醒, 并暂停未及时缴费用户的平台使用权限。基本流程为: a)系统登录—>b)使用确认—>c)费用缴纳。详情请参考计算中心主页的仪器设备使用预约及参考操作手册。

a)系统登录: 首先用户需要登录南京航空航天大学大型仪器设备共享管理平台, 校内用户通过统一身份认证登录, 如图 3 所示:



图 3 统一身份认证登录大仪平台

校外用户进入大仪平台官网后点击左上角“校外”按钮, 完成注册后输入账户密码登录, 如图 4 所示:



图 4 校外用户大仪入口

b)使用确认: 在“结算中心”模块中点击“个人消费记录”, 进入个人消费记录

页面。用户核对使用及消费信息，确认无误即点击“确认”按钮。



图 5 使用确认

c)费用缴纳：仪器管理员生成结算单并提交财务后，用户可在“结算中心--待缴费用”模块，打印结算单。结算单由财务账号负责人签字并加盖印鉴后，提交至高性能计算中心（本部提交至综合楼 710，江宁提交至 1 号楼 8 楼办公室），由高性能计算中心汇总至国资处实验室管理科（本部综合楼 326），并统一提交财务处结算。



图 6 校内用户费用缴纳

校外用户可在“结算中心--待缴费用”模块，查看待缴费情况，并可通过扫描缴费和对公汇款进行费用结算。

(1) 扫描缴费：点击缴费按钮，弹出缴费二维码，扫码即可进行缴费。

(2) 对公汇款：

账户名称：南京航空航天大学

开户银行：交通银行南京御道街支行

银行账号：320006639010149000354

交行行号：301301000394

4) 使用：用户签署平台服务协议后，用户可登录平台确定软硬件环境（详见第 3、4 章），校外用户登录需要先登录校园 VPN。登录集群和文件上传下载详见第 4 章；提交计算作业详见第 5 章；用户常见问题和注意事项详见第 6 章。注意，学生用户使用记录会关联到指导老师账户。

高性能计算平台鼓励用户在发表依托平台产出的成果时标注平台信息，在正

文或致谢中注明“研究工作得到南京航空航天大学高性能计算平台支持”（This work is partially supported by High Performance Computing Platform of Nanjing University of Aeronautics and Astronautics）。

5) 销户：若不再需要继续使用高性能计算平台资源，可主动联系管理员进行注销账户。

2.2 使用规范

1) 用户应遵守《中华人民共和国保守国家秘密法》等国家保密相关法律法规，禁止使用平台处理涉密数据或信息。

2) 用户应遵守《中华人民共和国网络安全法》等网络安全相关法律法规，禁止使用平台从事违法犯罪活动。

3) 用户使用平台时，一律通过安装在相应集群上的作业管理系统提交作业，进行计算和获取结果，不可绕过作业管理系统使用平台资源。

4) 不得擅自转让、出借账号，将口令随意告诉他人；也不得借用他人账号使用计算资源。

5) 不得窃取他人账号或口令，非法入侵他人账号，阅读他人文件，窃取他人计算和研究成果或受法律保护的资源。

6) 用户不得利用平台制造和传播计算机病毒；禁止破坏数据、破坏程序或其他恶意行为。

7) 用户应高度重视作业和结果的安全，及时收取计算结果，防止数据损坏、丢失或泄露。

8) 增强自我保护意识，及时反映违反用户行为规范的人和事。平台有权停止账号的使用权限，并对违规用户做出相应的处罚。

9) 用户有义务及时反馈平台存在的问题，有义务积极配合平台进行的统计、测试等各项工作。

3 快速入门

本章让用户快速掌握如何使用二期平台集群提交作业。

用户在获取上机账号后，Linux/Mac 用户可直接使用系统通过 ssh 命令：`ssh user_name@sc2.nuaa.edu.cn` 登录集群，Windows 用户则可以通过 ssh 客户端（例如 Xshell）登录集群（注意，集群首次登录详细步骤见第 4 章），为了方便文件传输，可同时下载并安装 Xftp，安装完后点击软件左上角新建连接，输入 IP 和用户名密码即可登录。

登录集群后，编写作业脚本，并通过 `sbatch` 指令将作业提交到计算节点上执行；此外，集群上安装了常见的计算软件，通过 `module` 指令导入计算环境。假设我们的计算过程为：在计算节点上运行 `hostname` 指令，那么就可以这么编写作业脚本：

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
#SBATCH --output=%J.out

Hostname
```

假设上面作业脚本的文件名为 `job.sh`，通过以下命令提交：

```
sbatch job.sh
```

集群安装了常见的计算软件，可以通过 `module` 指令导入计算环境；可以通过 `module` 加载平台上装有的软件环境，也可以自行安装配置需要的计算环境，下面的作业脚本加载了 `intel/2017u5` 的软件环境，具体可用的软件环境可使用命令 `module avail` 指令进行查看。

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
#SBATCH --output=%J.out

module purge
module load intel /2017u5
```

4 登录集群和文件上传下载

高性能计算平台二期平台管理系统为校内用户提供了统一身份认证登录渠道，用户可自行选择通过二期平台管理系统登录或使用 xshell 等软件登录进行操作。二期平台管理系统具体操作说明可参考高性能计算中心主页《高性能计算二期平台管理系统用户端操作手册》

4.1 平台管理系统注册

用户首次登录前，需在二期平台管理系统进行注册，初始化节点登录密码(非统一身份认证密码)。

1) 二期平台管理系统登录

打开浏览器，输入 <https://hpcpt2.nuaa.edu.cn> 进入登录入口，校外人员通过管理员提供的账户密码从图 7 页面入口登录，校内人员点击右下角统一身份认证入口统一身份认证账户密码登录，如图 8 所示。



图 7 二期平台管理系统入口



图 8 统一身份认证入口

2) 用户信息完善

校内人员统一身份认证后,页面跳转至平台用户信息完善页面(如图 9 所示),用户需自行设置高性能计算平台的节点登录密码,点击提交后等待管理员审核通过,下次登录即可通过统一身份认证直接登入平台。



图 9 用户信息完善



图 10 注册成功页面

4.2 Windows 系统

以 Xshell 为例, 命令: `ssh user_name@#@10.12.6.11@sc2.nuaa.edu.cn`, 回车。

如图 11 所示:

```
[D:\~]$ ssh zhouhao@#@10.12.6.11@sc2.nuaa.edu.cn

Host 'sc2.nuaa.edu.cn' resolved to 172.18.101.68.
Connecting to 172.18.101.68:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Last login: Thu Aug 25 14:56:18 2022 from 10.12.5.115

正在进行认证,请稍后 .....

Using username "zhouhao".
Last login: Thu Aug 25 10:35:08 2022
[zhouhao@login1 ~]$
```

图 11 远程登录

如遇下述情况需要保留主机秘钥, 选择“接受并保存”, 如图 12 所示:

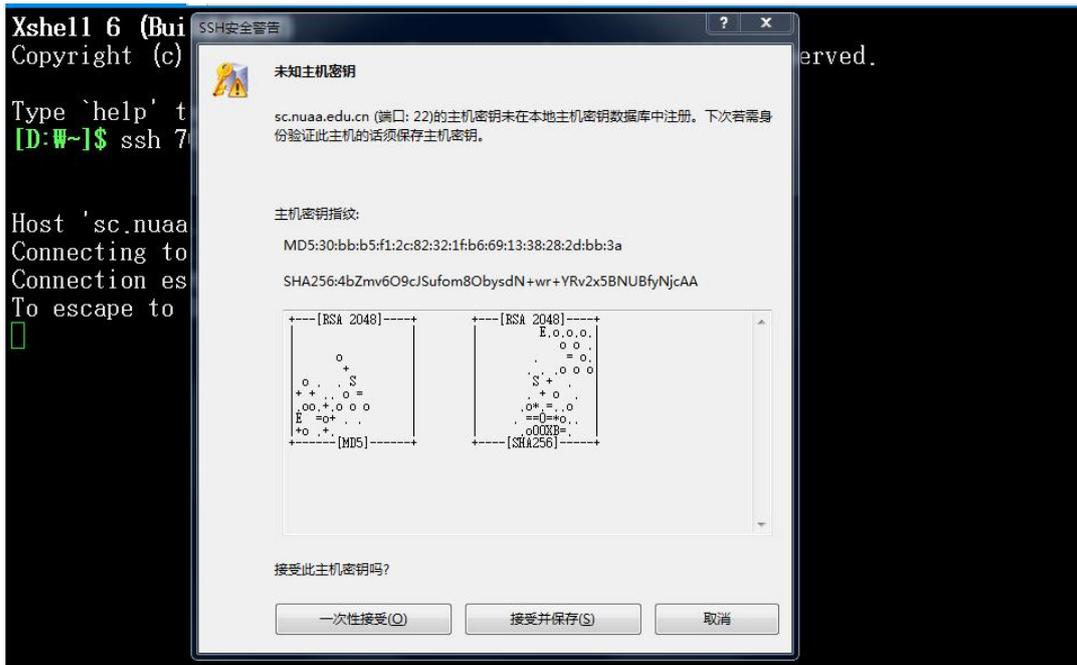


图 12 接收密钥

如图 13 所示表示登录成功:

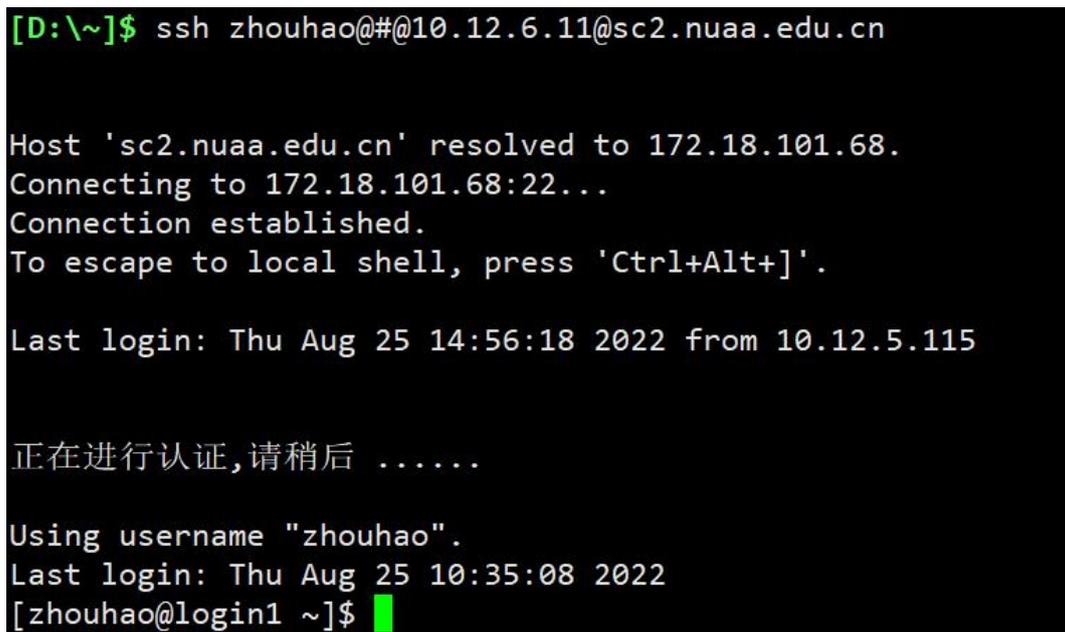


图 13 登录成功

4.3 Linux/Mac 系统

注意: 因不同发行版 (包括 Mac/CentOS/Ubuntu 等) 命令行的差异, 下述的 `-p` 参数可能是 `-P`, 或其他可能的参数, 请自行确定后再次尝试。

Linux/Mac 等系统, 命令: `ssh user_name@#@10.12.6.11@sc2.nuaa.edu.cn,`

回车。详细操作代码如图 14 所示：

```
C:\Users\zh>ssh zhouhao@#10.12.6.11@sc2.nuaa.edu.cn
The authenticity of host 'sc2.nuaa.edu.cn (172.18.101.68)' can't be established.
RSA key fingerprint is SHA256:hPAPhRVEOhMqVzvZ1liJb1EYOX760560qMvaIV+qBvQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'sc2.nuaa.edu.cn,172.18.101.68' (RSA) to the list of known hosts.
zhouhao@#10.12.6.11@sc2.nuaa.edu.cn's password:
Last login: Thu Aug 25 15:28:46 2022 from 10.100.246.72

正在进行认证,请稍后 .....

Using username "zhouhao".
Last login: Thu Aug 25 15:30:04 2022
[zhouhao@login1 ~]$
```

图 14 Linux/max 系统登录命令

4.4 校外用户登录 VPN

校外用户打开浏览器，在地址栏中输入 <https://v.nuaa.edu.cn>



图 15 校外访问 VPN

输入用户名和密码点击登录。

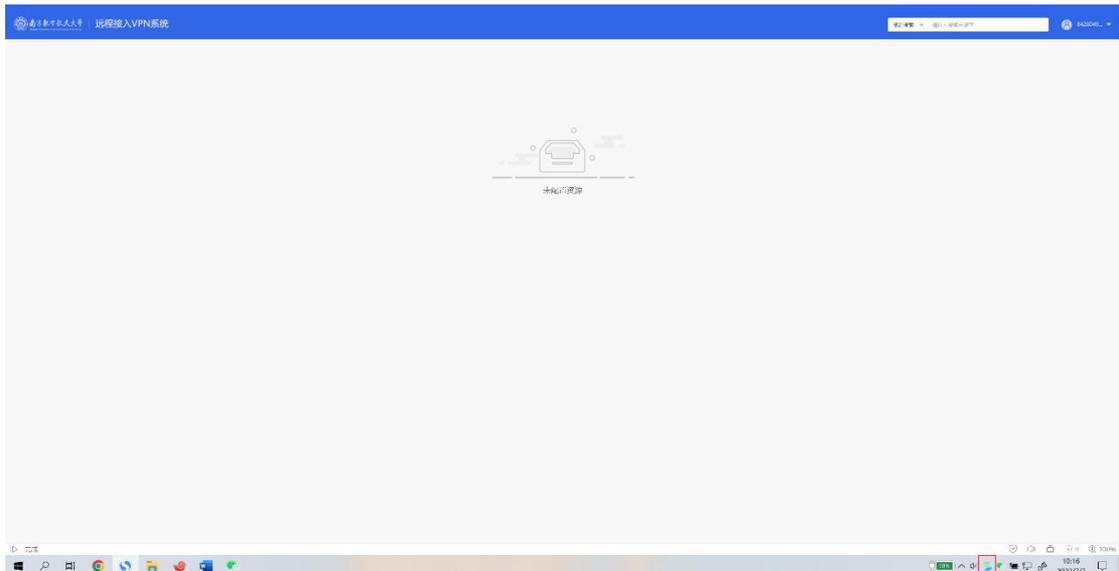


图 16 登录成功

4.5 Windows 系统文件上传下载

Windows 系统文件上传/下载需要参照 4.2 节的步骤通过 Xshell 远程登录软件使用 ssh 登录集群的登录节点。ssh 登录成功后点击 Xshell 工具栏中的“新建文件传输”图标，如图 17 所示，在提示窗口输入密码。在 Xftp 登录成功后即可用拖拽的方式从左侧本地电脑窗口将所需要上传的文件拖拽至右侧代表集群空间的窗口从而实现文件的上传。如图 18 所示。如需从集群空间下载文件，则请反向操作。

注意：如果出现大文件上传失败请联系管理员以协助定位并解决问题。

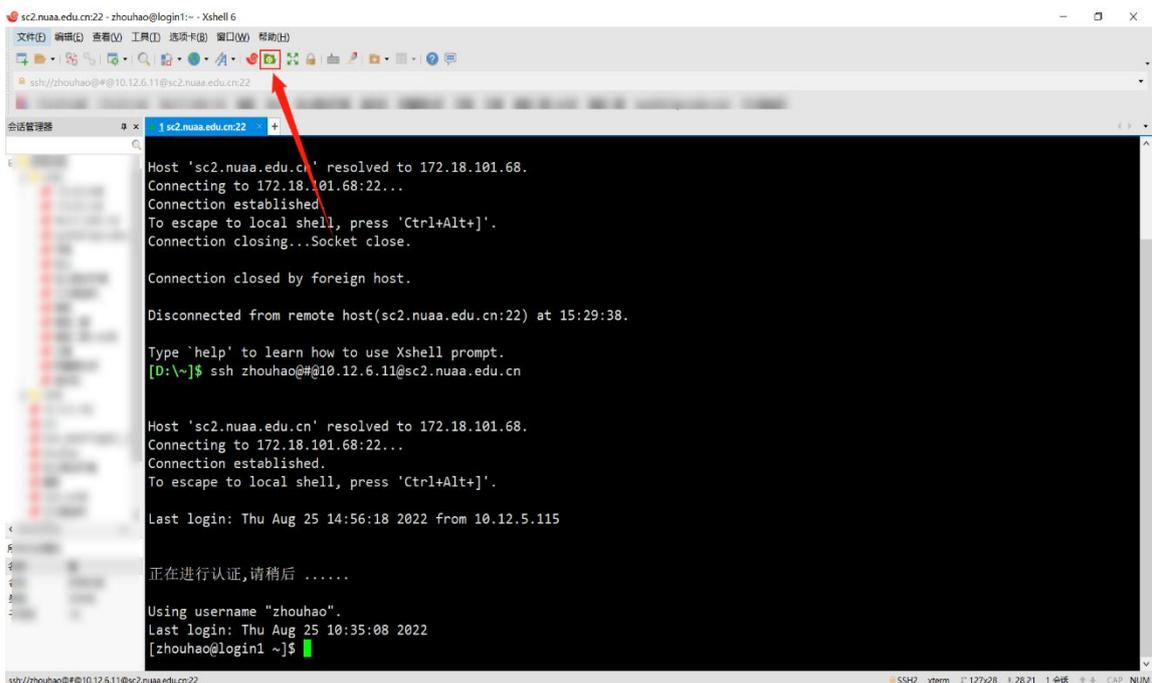


图 17 文件下载

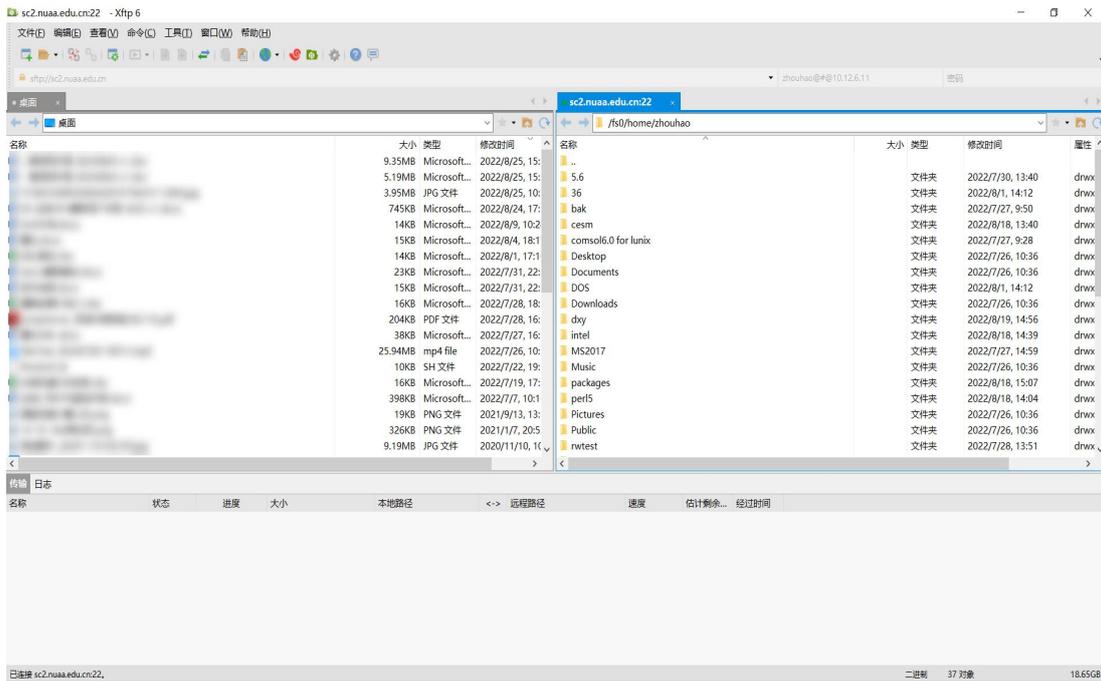


图 18 文件下载

4.6 类 Unix 系统文件上传下载

针对 Mac/CentOS/Ubuntu 等发行版的 Linux 系统，如果您已安装了对应系统的 Xshell 软件则参照 4.3 节中的步骤。如果没有，则需要使用命令行的方式进行文件传输。命令为：`sftp user_name@sc2.nuaa.edu.cn` 回车。如图 19 所示：

```
[root@node08 ~]# sftp zhouhao@172.18.101.68
zhouhao@172.18.101.68's password:
Connected to 172.18.101.68.
sftp>
```

图 19 类 Unix 文件下载

Sftp 登录后的命令行中可以使用部分当前 Linux 系统所提供的部分命令（如 `ls/cd` 等命令），通过输入问号(?)获取 Sftp 命令行界面所支持的所有命令。如图 20 所示：

```
File Edit View Search Terminal Tabs Help
root@systemtag:~
sftp> ?
Available commands:
bye                               Quit sftp
cd path                           Change remote directory to 'path'
chgrp grp path                    Change group of file 'path' to 'grp'
chmod mode path                   Change permissions of file 'path' to 'mode'
chown own path                    Change owner of file 'path' to 'own'
df [-hi] [path]                  Display statistics for current directory or
                                  filesystem containing 'path'
exit                               Quit sftp
get [-afPpRr] remote [local]     Download file
reget [-fPpRr] remote [local]    Resume download file
reput [-fPpRr] [local] remote    Resume upload file
help                               Display this help text
lcd path                           Change local directory to 'path'
lls [ls-options] [path]]         Display local directory listing
lmkdir path                       Create local directory
ln [-s] oldpath newpath          Link remote file (-s for symLink)
lpwd                               Print local working directory
ls [-lafhlnrSt] [path]           Display remote directory listing
lumask umask                      Set local umask to 'umask'
mkdir path                         Create remote directory
progress                           Toggle display of progress meter
```

图 20 命令汇总

上传下载命令如下:

从本地电脑上传文件到集群空间:

#上传单个文件: sftp>put/opt/testfile

#上传文件夹: sftp>put-r/opt/testdir

从集群空间下载文件到本地电脑:

#下载单个文件: sftp>get/data/home/zhouhao/testfile

#下载文件夹: sftp>get-r/data/home/zhouhao/testdir

5 提交作业

5.1 Slurm 作业调度系统使用说明

二期平台使用 Slurm 作业调度系统管理所有计算作业，该系统接受用户的作业请求，并将作业合理的分配到合适的节点上运行。下图为用户提交作业的示意图：

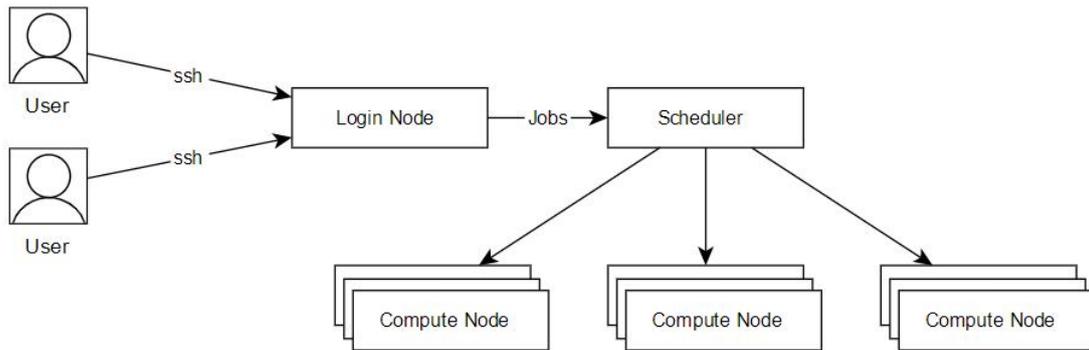


图 21 提交作业示意图

本节将介绍运行作业的两种方式，一种是将计算过程写成脚本，通过 `sbatch` 指令提交到计算节点执行，另一种是通过 `salloc` 申请到计算节点，再 `ssh` 连接到计算节点进行计算本节还将介绍如何 `sinfo`、`squeue`、`scancel` 等命令具体操作。

5.1.1 sbatch 提交作业

用户使用 `sbatch` 命令向作业调度系统提交作业，`sbatch` 可用参数十分丰富，可对作业进行非常细致的控制，这里简要介绍常用参数和方法。

运行作业的第一种方式是将整个计算过程，写到脚本中，通过 `sbatch` 指令提交到计算节点上执行。首先介绍一个简单的例子，假设我们的计算过程为，在计算节点上运行 `hostname` 指令，那么就可以如下编写作业脚本：

```
#!/bin/bash
#!/bin/bash
#SBATCH-J test
#SBATCH-p cpu
#SBATCH-n 64
#SBATCH--error=%J.err
#SBATCH--output=%J.out
hostname
```

假设上面作业脚本的文件名为 `job.sh`，通过以下命令提交：

```
sbatch job.sh
```

随后我们介绍脚本中涉及的参数：

```
-J test#作业在调度系统中的作业名为 test;
-p cpu##作业提交的指定分区为 cpu
-n 64#这个作业使用 64 核运行，如果程序不支持多线程(如 openmp)，这个数不应该超过 1;
--error=%J.err#脚本执行的错误输出将被保存在当%j.err 文件下，%j 表示作业号;
--output=%J.out#脚本执行的输出将被保存在当%j.out 文件下，%j 表示作业号;
```

除此之外，还有一些常见的参数：

```
--help#显示帮助信息;
-D,--chdir=<directory>#指定工作目录;
--get-user-env#获取当前的环境变量;
--gres=<list>#使用 gpu 这类资源，如申请两块 gpu 则--gres=gpu:2
-J,--job-name=<jobname>#指定该作业的作业名;
--mail-type=<type>#指定状态发生时，发送邮件通知，有效种类为
(NONE,BEGIN,END,FAIL,REQUEUE,ALL);
--mail-user=<user>#发送给指定邮箱;
-n,--ntasks=<number>#SBATCH 并不会执行任务，当需要申请相应的资源来运行脚本，默认情况下一个任务一个核心，--cpus-per-task 参数可以修改该默认值;
-c,--cpus-per-task=<ncpus>#每个任务所需要的核心数，默认为 1;
--ntasks-per-node=<ntasks>#每个节点的任务数，--ntasks 参数的优先级高于该参数，如果使用--ntasks 这个参数，那么将会变为每个节点最多运行的任务数;
-o,--output=<filename pattern>#输出文件，作业脚本中的输出将会输出到该文件;
-p,--partition=<partition_names>#将作业提交到对应分区;
-q,--qos=<qos>#指定 QOS;
-t,--time=<time>#允许作业运行的最大时间，目前未名一号和生科一号为 5 天，教学一号为两天;
-w,--nodelist=<node name list>#指定申请的节点;
-x,--exclude=<node name list>#排除指定的节点;
```

接下来是一个 GPU 作业的例子，假设我们想要申请一块 GPU 卡，并通过指令 `nvidia-smi` 来查看申请到 GPU 卡的信息，那么可以这么编写作业脚本：

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p gpu4
#SBATCH -n 4
#SBATCH --gres=gpu:1
#SBATCH --error=%J.err
#SBATCH --output=%J.out
nvidia-smi
```

脚本中的一些参数说明如下

```
#SBATCH --gres=gpu:1#每个节点上申请一块 GPU 卡
```

最后是一个跨节点多核心的例子，假设我们想用两个节点，每个节点 40 个核心来运行 `vasp`，那么可以这么编写作业脚本：

```
#!/bin/bash
#SBATCH-J test
#SBATCH-p cpu
#SBATCH-N 2
#SBATCH--ntasks-per-node=40
#SBATCH--error=%J.err
#SBATCH--output=%J.out
#导入 MPI 运行环境
module load intel/2017u5
#导入 VASP 运行环境
module load vasp/5.4.4
#执行 VASP 并行计算程序
mpirun-n 80 vasp_std
scontrol show job$SLURM_JOBID
```

5.1.2 salloc 交互式运行作业

运行作业的第二种方式是通过 salloc 交互式运行作业，首先需要申请计算节点，然后登录到申请到的计算节点上运行指令。salloc 的参数与 sbatch 相同，以下提供申请一个节点 6 个核心，并跳转到该节点上运行程序示例：

```
salloc-p cpu-N1-n6
#salloc 申请成功后会返回申请到的节点和作业 ID 等信息,假设申请到的是 cu01 节点,
作业 ID 为 1078858
ssh cu01#直接登录到刚刚申请到的节点 cu01 调式作业
scancel 1078858#计算资源使用完后取消作业
squeue-j 1078858#查看作业是否还在运行, 确保作业已经退出, 避免产生不必要的费用
```

随后是一个 GPU 节点的使用案例；

申请一个 GPU 节点，6 个核心，1 块 GPU 卡，并跳转到节点上运行程序；

```
salloc -p gpu4 -N1 -n6 --gres=gpu:4
# 假设申请成功后返回的作业号为 1078858, 申请到的节点是 gpu05
ssh gpu05 # 登录到 gpu05 上调式作业
scancel 1078858 # 计算结束后结束任务
squeue -j 1078858 # 确保作业已经退出
```

最后介绍一个跨节点使用案例；

申请两个节点，每个节点 12 个核心

```
salloc -p cpu -N2 --ntasks-per-node=80
# salloc 申请成功后会返回申请到的节点和作业 ID 等信息, 假设申请到的是
a8u03n[05-06]节点, 作业 ID 为 1078858
# 这里申请两个节点, 每个节点 12 个进程, 每个进程一个核心
```

```

# 根据需求导入 MPI 环境
module load intel/2017u5

# 根据以下命令生成 MPI 需要的 machine file
srun hostname -s | sort -n > slurm.hosts

mpirun -np 80 -machinefile slurm.hosts hostname

# 结束后退出或者结束任务
scancel 1078858

```

5.1.3 sinfo 查看资源空闲状态

sinfo 可查询各分区节点的空闲状态，输入 sinfo 命令，返回状态显示 idle 为空闲，mix 为节点部分核心可以使用，alloc 为已被占用，maint 为维护中，如下所示：

```

[root@cu60 ~]# sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
cpu*      up    infinite   2    maint cu[02,60]
cpu*      up    infinite   3     mix  cu[01,03,62]
cpu*      up    infinite   7    alloc cu[04,17-21,61]
cpu*      up    infinite  50    idle  cu[05-16,22-59]
gpu4      up    infinite   1     mix$ gpu07
gpu4      up    infinite   1    alloc$ gpu04
gpu4      up    infinite   1     maint gpu08

```

sinfo 的一些常用参数：

```

--help      # 显示 sinfo 命令的使用帮助信息；
-d          # 查看集群中没有响应的节点；
-i <seconds> # 每隔相应的秒数，对输出的分区节点信息进行刷新
-n <name_list> # 显示指定节点的信息，如果指定多个节点的话用逗号隔开；
-N         # 按每个节点一行的格式来显示信息；
-p # <partition> 显示指定分区的信息，如果指定多个分区的话用逗号隔开；
-r        # 只显示响应的节点；
-R        # 显示节点不正常工作的原因；

```

5.1.4 squeue 查看作业队列

用户可以通过 squeue 命令查看提交作业的排队情况，如下所示输入 squeue 命令：

```

squeue#二期平台集群使用命令

```

默认情况下 squeue 输出的内容如下，分别是作业号，分区，作业名，用

户，作业状态，运行时间，节点数量，运行节点（如果还在排队则显示排队原因）。

```
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

squeue 的常见参数：

```
--help#显示 squeue 命令的使用帮助信息；  
-A<account_list>#显示指定账户下所有用户的作业，如果是多个账户的话用逗号隔  
开；  
-i<seconds>#每隔相应的秒数，对输出的作业信息进行刷新  
-j<job_id_list>#显示指定作业号的作业信息，如果是多个作业号的话用逗号隔开；  
-n<name_list>#显示指定节点上的作业信息，如果指定多个节点的话用逗号隔开；  
-t<state_list>#显示指定状态的作业信息，如果指定多个状态的话用逗号隔开；  
-u<user_list>#显示指定用户的作业信息，如果是多个用户的话用逗号隔开；  
-w<hostlist>#显示指定节点上运行的作业，如果是多个节点的话用逗号隔开；
```

5.1.5 scancel 取消作业

用户可以通过 scancel 命令取消账号中已提交的作业，如下所示：

```
#取消作业 ID 为 123 的作业  
scancel 123
```

也可通过 scancel 命令取消自己账号上所有作业：

```
#注意 whoami 前后不是单引号  
scancel-u user_name
```

scancel 常见参数：

```
--help#显示 scancel 命令的使用帮助信息；  
-A<account>#取消指定账户的作业，如果没有指定 job_id,将取消所有；  
-n<job_name>#取消指定作业名的作业；  
-p<partition_name>#取消指定分区的作业；  
-q<qos>#取消指定 qos 的作业；  
-t<job_state_name>#取消指定作态的作业，"PENDING","RUNNING"或  
"SUSPENDED";  
-u<user_name>#取消指定用户下的作业；
```

5.2 常用科学计算软件使用

5.2.1 GNU 编译器套件

GCC（GNU Compiler Collection，GNU 编译器套件）是由 GNU 开发的编程语言译器。GNU 编译器套件包括 C、C++、Objective-C、Fortran、Java、Ada 和 Go 语言前端，也包括了这些语言的库（如 libstdc++，libgcj 等）。

安装 GCC 操作方法如下（因二期平台是内网环境，无法连接互联网，可以通过下载安装包后按照 4.6、4.7 节的方法进行上传，再进行安装操作）：

```

# 获取源码
export VERSION=9.3.0
wget https://mirrors.aliyun.com/gnu/gcc/gcc-${VERSION}/gcc-${VERSION}.tar.gz

# 解压
tar xf gcc-${VERSION}.tar.gz
cd gcc-${VERSION}

# 编译安装
./contrib/download_prerequisites
# 如果没有办法下载依赖，可以尝试修改./contrib/download_prerequisites 文件
# 把 base_url='ftp://gcc.gnu.org/pub/gcc/infrastructure/' 改成
# base_url='http://gcc.gnu.org/pub/gcc/infrastructure/'
# 定义安装路径
export PREFIX=/gpfs/share/software/gcc/9.3.0
./configure --prefix=${PREFIX} --enable-languages=c,c++,fortran --build=x86_64-linux
--disable-multilib
make
make install

```

环境变量设置如下：

```

PATH
CPATH
LIBRARY_PATH
LD_LIBRARY_PATH

```

GCC 使用方法如下：平台上安装了多个版本的 gcc，可以直接通过 module 加载使用；加载 gcc 编译器代码如下：

```
module load gcc/7.3.0
```

测试代码 test.c

```

#include <stdio.h>
int main(void)
{
    printf("Hello world!\n");
    return 0;
}

```

使用 gcc 编译器编译连接

```
gcc test.c -o test
```

运行编译好的程序

```
./test
```

5.2.2 Gromacs

Gromacs 是用于研究生物分子体系的分子动力学程序包。它可以用分子动力学、随机动力学或者路径积分方法模拟溶液或晶体中的任意分子，进行分子能量的最小化，分析构象等。同时它是一个功能强大的分子动力学的模拟软件，其在模拟大量分子系统的牛顿运动方面具有极大的优势。它的模拟程序包包含 Gromacs 力场(蛋白质、核苷酸、糖等)，研究的范围可以包括玻璃和液晶、到聚合物、晶体和生物分子溶液。

Gromacs CPU 版本安装方法如下（因二期平台是内网环境，无法连接互联网，可以通过下载安装包后按照 4.6、4.7 节的方法进行上传，再进行安装操作）：

```
# 获取源码
export VERSION=2021.3
wget --no-check-certificate https://ftp.gromacs.org/gromacs/gromacs-${VERSION}.tar.gz

# 解压
tar xf gromacs-${VERSION}.tar.gz
cd gromacs-${VERSION}

# 编译安装
# 导入编译环境，avx512 需要高版本 gcc 支持
module load openmpi/3.1.1
module load cmake/3.18.3
module load gcc/7.3.0

# cpu 版本安装
export CC=gcc CXX=g++
mkdir build ; cd build

# -DGMX_SIMD=AVX_512 需要在支持 avx512 的 CPU 处理器上才可编译成功
PREFIX=/gpfs/share/software/gromacs/2021.3/gcc-7.2.0/openmpi-3.1.3
cmake .. -DGMX_BUILD_OWN_FFTW=ON -DGMX_MPI=ON -DGMX_SIMD=AVX_512
-DMAKE_INSTALL_PREFIX=$PREFIX
make && make install

GPU 版本安装
# gpu 版本的编译需要在 gpu 节点上执行
# 获取源码
export VERSION=2021.3
wget --no-check-certificate https://ftp.gromacs.org/gromacs/gromacs-${VERSION}.tar.gz

# 解压
tar xf gromacs-${VERSION}.tar.gz
```

```

cd gromacs-${VERSION}

# 编译安装
# 导入编译环境, avx512 需要高版本 gcc 支持
module load openmpi/3.1.1
module load cmake/3.18.3
module load gcc/7.3.0

# cpu 版本安装
export CC=gcc CXX=g++
mkdir build_gpu ; cd build_gpu

# -DGMX_SIMD=AVX_512 需要在支持 avx512 的 CPU 处理器上才可编译成功
PREFIX=/gpfs/share/software/gromacs/2021.3_gpu/gcc_7.2.0/openmpi_3.1.3
cmake .. -DGMX_BUILD_OWN_FFTW=ON -DGMX_GPU=CUDA -DGMX_MPI=ON
-DGMX_SIMD=AVX_512 -DCMAKE_INSTALL_PREFIX=$PREFIX
make && make install

```

Gromacs 提交作业步骤如下:

step 1. 创建工作目录;

```

mkdir gromacsJob1
cd gromacsJob

```

step 2. 将运行 gromacs 需要的相关文件上传到该文件夹下;

step 3. 在该文件夹下编写作业脚本, 并命名为 gromacs.sh, 脚本内容如下:

```

#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
#SBATCH --output=%J.out
module load gcc/7.3.0
module load openmpi/3.1.1
/gpfs/share/software/gromacs/2021.3_gpu/gcc_7.2.0/openmpi_3.1.3/bin/GMXRC.bash
mpirun -np 48
/gpfs/share/software/gromacs/2021.3_gpu/gcc_7.2.0/openmpi_3.1.3/bin/gmx_mpi mdrun -deffnm
md1-cluster0 -dlb yes

```

step 4. 提交作业:

```

sbatch gromacs.sh

```

5.2.3 vasp

VASP 是维也纳大学 Hafner 小组开发的进行电子结构计算和量子力学-分

子动力学模拟软件包。它是目前材料模拟和计算物质科学研究中最流行的商用软件之一。

安装 vasp 操作方法如下（因二期平台是内网环境，无法连接互联网，可以通过下载安装包后按照 4.6、4.7 节的方法进行上传，再进行安装操作）：

```
# 导入编译器
module load intel/2017u5
tar xf vasp.5.4.4.tar.gz
cd vasp.5.4.4
cp arch/makefile.include.linux_intel ./makefile.include
make all
# 编译完成后会在 vasp.5.4.4 的文件夹下的 bin 文件夹里生成 vasp_gam、vasp_ncl、vasp_std
三个可执行文件。

# gpu 版本安装
module load intel/2017u5
tar xf vasp.5.4.4.tar.gz
cd vasp.5.4.4
cp arch/makefile.include.linux_intel makefile.include
# 修改 -openmp 为 -qopenmp
make gpu
# GPU 版本使用的时候需要载入 cuda，高版本如 2018 的 intel 编译器编译时会报错
```

VASP 提交作业步骤如下：

step 1. 创建工作目录；

```
mkdir vaspJob1
cd vaspJob1
```

step 2. 将运行 vasp 需要的相关文件上传到该文件夹下；

step 3. 在该文件夹下编写作业脚本，并命名为 vaspJob1.sh，脚本内容如下：

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p cpu
#SBATCH -n 64
#SBATCH --error=%J.err
#SBATCH --output=%J.out

# 导入运行环境
module load intel/2017u5
module load vasp/5.4.4
```

```
# MPI 跨节点运行
mpirun -n 64 vasp_std
```

step 4. 提交作业:

```
sbatch vaspJob1.sh
```

5.3 可视化作业

二期平台集群支持提交图形化作业。节点上部署的图形化界面可以方便用户进行桌面查看和操作,而不需要通过命令行来查看作业。**如未能打开客户端界面,请及时联系管理员处理,以免计算资源浪费。**操作步骤主要如下:

首先打开浏览器,在地址栏中输入 <https://hpcpt2.nuaa.edu.cn>,进入网址后校外用户需要输入账号密码,登录高性能计算平台。校内用户需点击统一认证入口,输入统一身份认证账户密码。



图 22 可视化作业页面

点击“作业管理-运行作业”,进入运行作业界面:

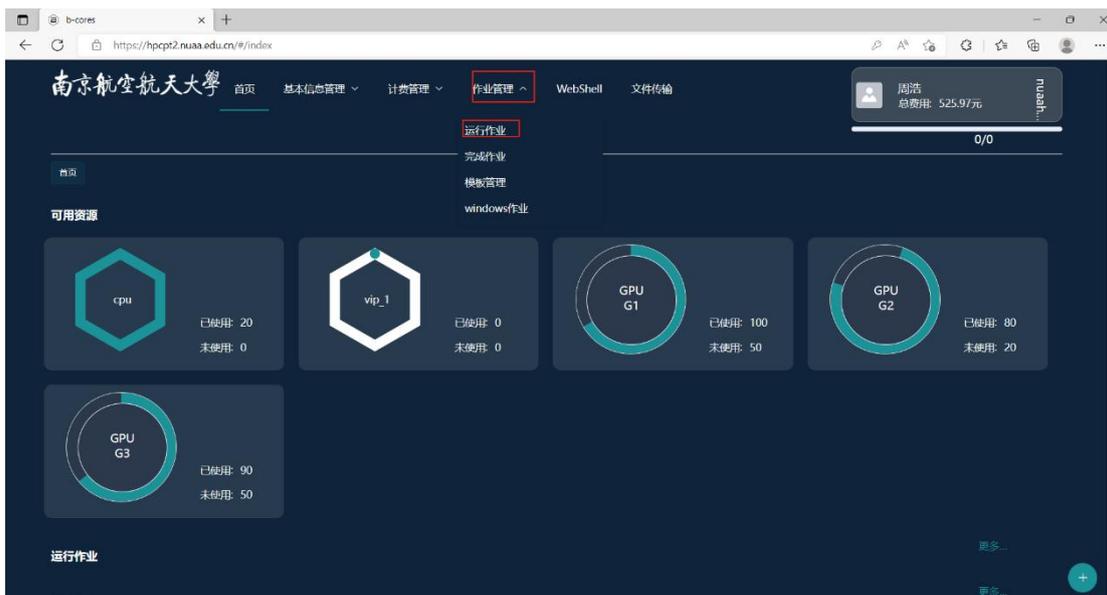


图 23 运行作业页面

点击图中“图形化作业”的选择框进入提交图形化作业界面，如图 24 所示：

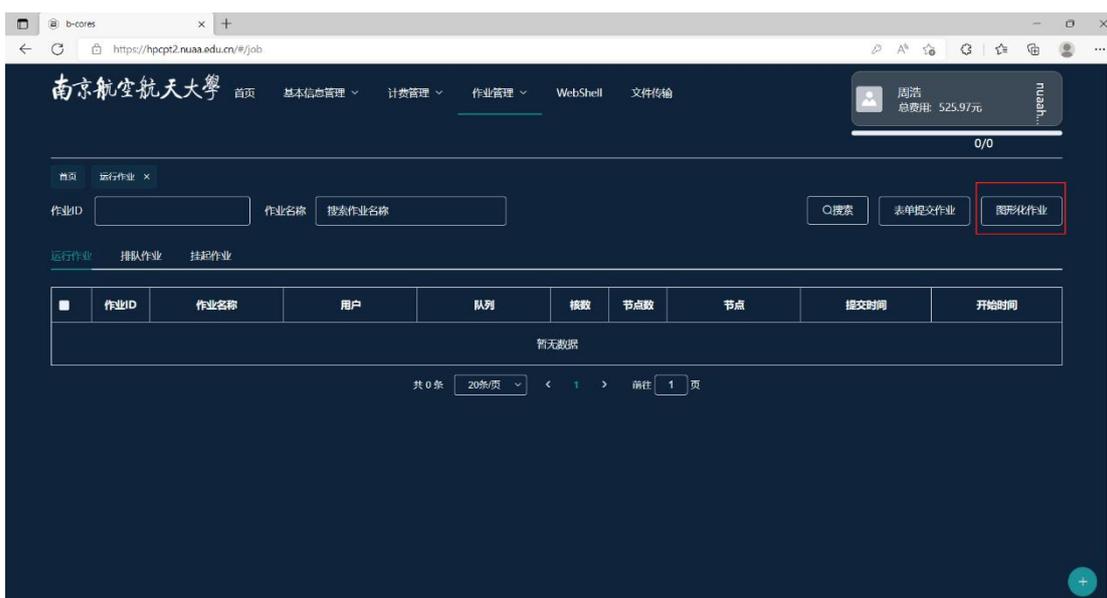


图 24 提交作业 1

输入作业名称，选择队列，输入节点数量。信息填写完成后点击提交，如图 25 所示：

注：最大运行时间非必填项，如果填写到时间后作业会被系统管理员 kill。

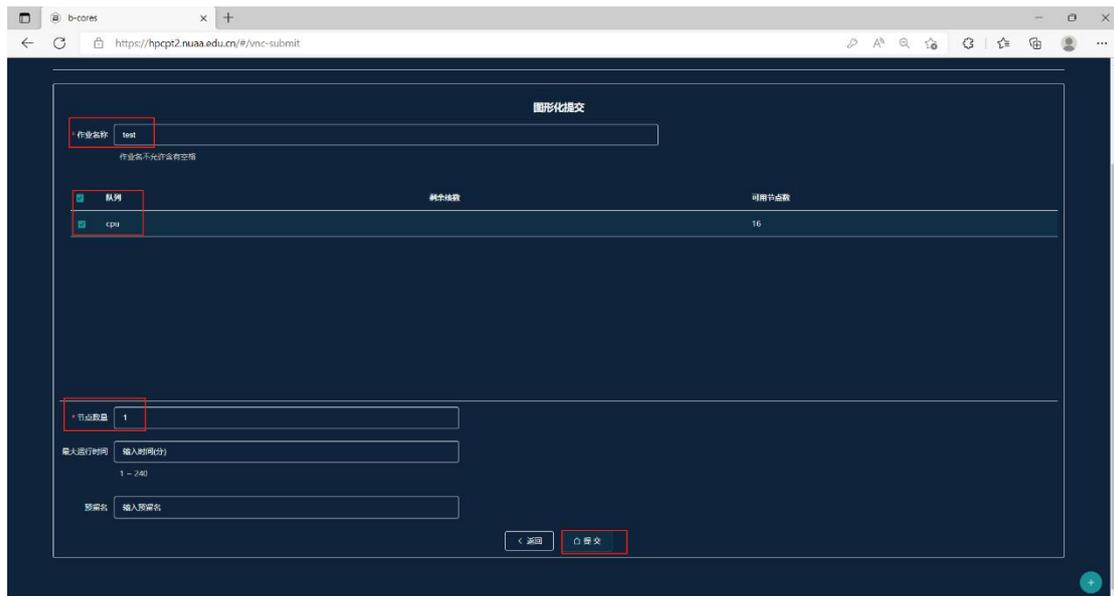


图 25 提交作业 2

作业提交后，点击运行作业，如图 26 所示：

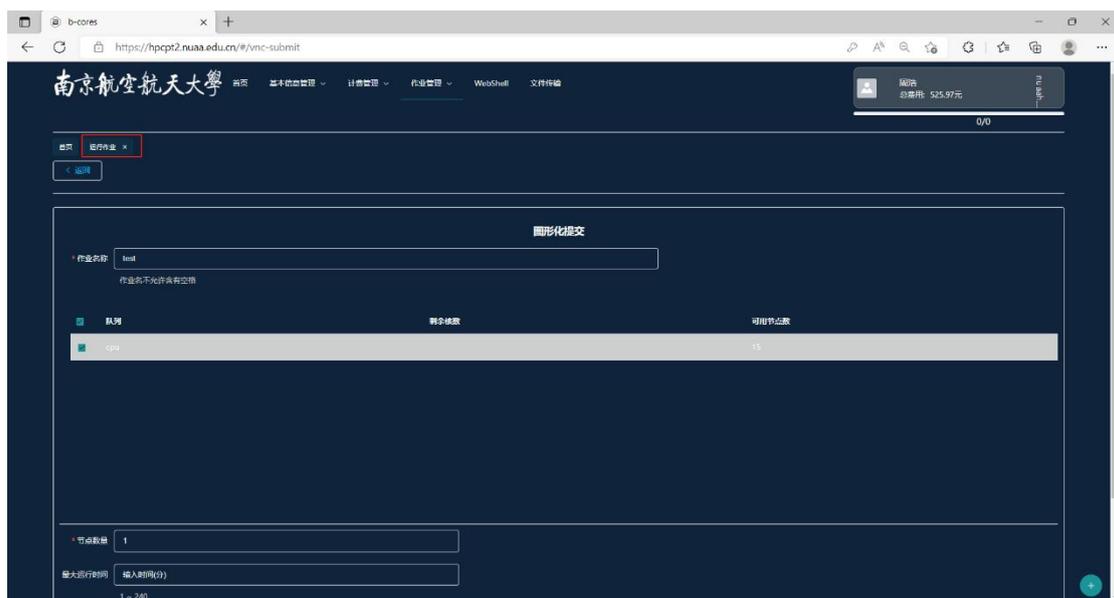


图 26 提交作业 3

如图 27 即为作业提交成功，点击查看详情即可查看作业：



图 27 提交作业成功

点击作业名称，查看作业信息，如图 28 所示：

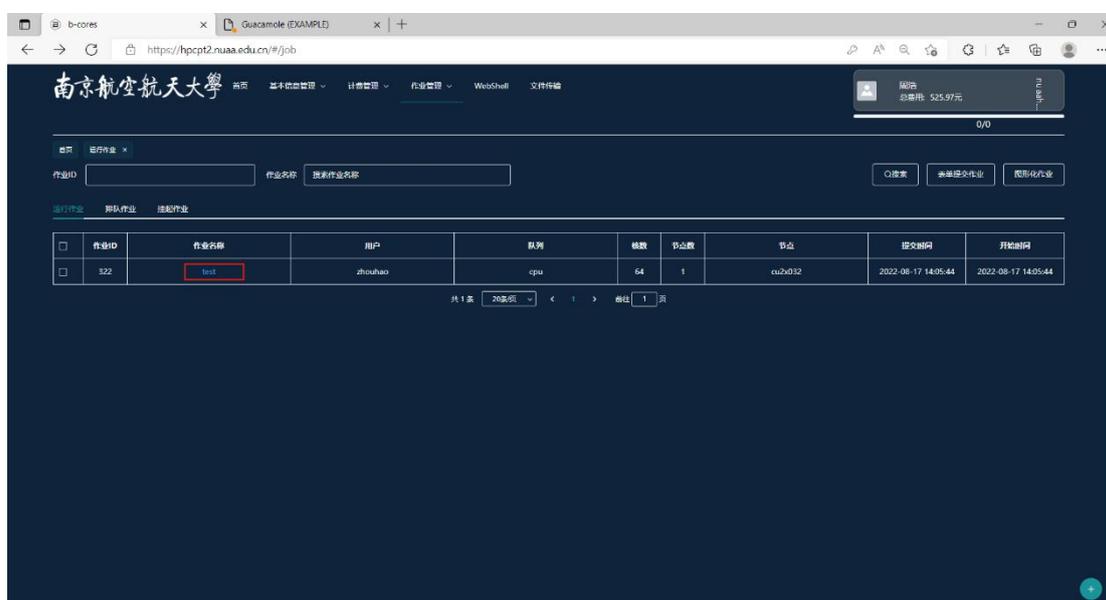


图 28 查看作业信息

点击客户端连接，进入计算节点 GUI 桌面。用户如完成作业需自行点击取消作业，以免占用节点持续计费，如图 29 所示。

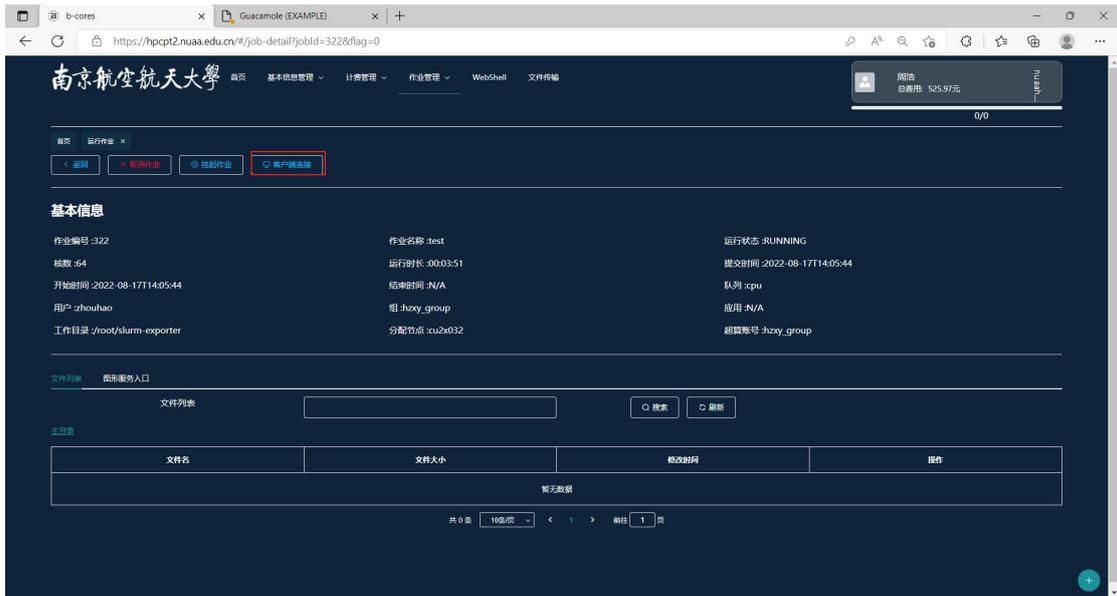


图 29 计算节点 GUI 桌面

成功打开计算节点 GUI 桌面，如图 30 所示：

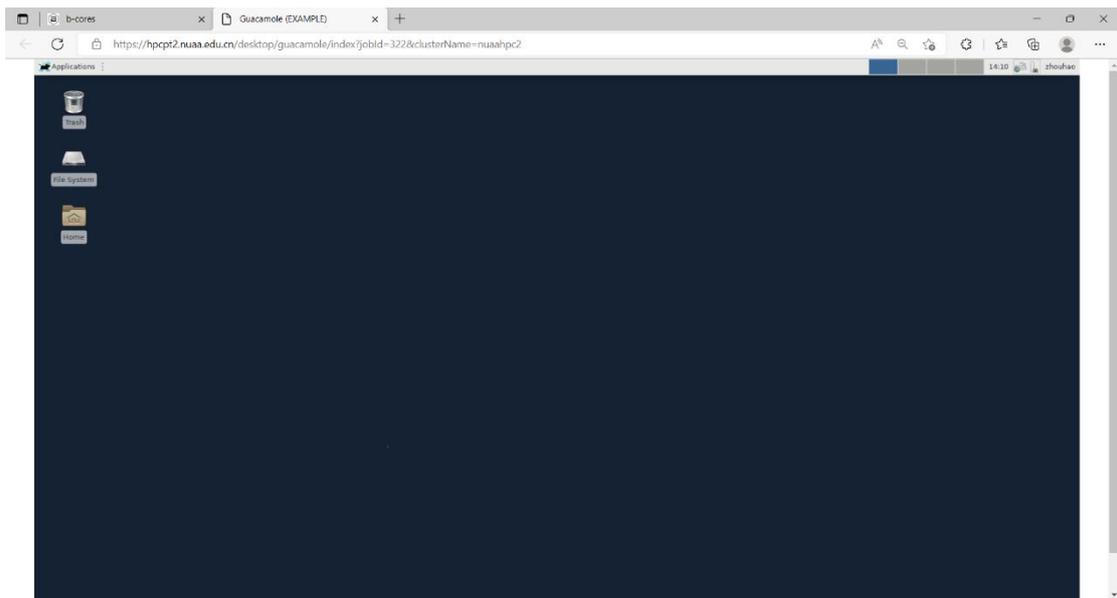


图 30 GUI 桌面

注：进入桌面后如有疑问请联系集群管理员协助操作。

5.4 windows 作业

二期平台部署了 8 台 windows 节点，支持用户提交 windows 作业。提交作业前需联系平台管理员开通权限，方可打开 windows 客户端界面。如未能打开客户端界面，请及时联系管理员处理，以免计算资源浪费。提交 windows 作业，具体操作步骤主要如下：

点击作业管理-windows 作业，进入 windows 作业列表界面，如图 31、32 所示：



图 31 windows 作业



图 32 windows 作业列表

点击图 32 中提交 windows 作业按钮进行作业提交，进入作业编辑界面，输入所需资源，如图 33 所示：



图 33 作业编辑界面

编辑完成后，点击提交按钮，提示作业提交成功即完成提交，如图 34 所示：



图 34 作业提交成功

点击查看详情，返回 windows 作业列表界面，即可查看已提交作业，点击作业名称，即可查看作业详情，点击作业名称或客户端链接都可进入 windows 客户端界面，如图 35、36 所示，用户如完成作业需自行点击取消作业，以免占用节点持续计费。

作业ID	作业名称	用户	队列	核数	节点数	节点	开始时间	操作
-9999995	test2		windows_nuaa	128	2	windows02,windows03	2022-08-31 17:10:11	
-9999980	test3		windows_nuaa	128	2	windows07,windows08	2022-09-01 09:42:38	

图 35 已提交作业列表

基本信息

作业编号 : -9999980	作业名称 : test3	运行状态 : RUNNING
核数 : 128	运行时长 : N/A	提交时间 : 2022-09-01 09:42:38
开始时间 : 2022-09-01 09:42:38	结束时间 : N/A	队列 : windows_nuaa
用户 : 82220073	组 : xj_nuaa	应用 : N/A
工作目录 : N/A	分配节点 : windows07,windows08	超算账号 : xj_nuaa

图 36 作业详情

如图 37 即为 windows 客户端界面。



图 37 windows 客户端界面

6 常见问题及注意事项

6.1 Xshell 工具在哪下载？

答：可以通过 Xshell 官方网站申请免费版 Xshell，具体申请地址如下：

<https://www.xshell.com/zh/free-for-home-school/>

6.2 我想用的计算软件没有怎么办？

答：计算软件可自行安装到宿主目录中，如不会安装请联系集群管理员协助安装。

6.3 我要用的计算软件作业脚本不会写怎么办？

答：用户所需具体脚本可通过询问集群管理员寻找目标软件的作业模板。如 ansys comsol vasp lammps 等。

6.4 提交作业报错如下错误

```
sbatch:error:Batch job submission failed:Invalid account or account/partition combination specified
```

答：导致如上报错的原因较多，大概原因有：脚本中队列设置错误，slurm 账号没有绑定等。具体问题需联系管理员逐个排查。

6.5 作业没有运行，并且显示 QOSGrpCpuLimit

答：出现此英文字符是因为账号核数有所限制，正在运行的作业达到核数限制后，其他作业状态则变为正在排队。

6.6 怎么修改账号密码

答：登录高性能计算平台二期平台管理系统修改密码，具体步骤可参考《高性能计算平台二期平台管理系统用户端操作手册》。

附录 1 Linux 常用命令

在 4.3 节 Linux 系统下命令行界面登陆成功后，即可进行操作，下面列举了常用的 Linux 系统命令：

1. 查看目录下有哪些文件：ls
2. 查看文件属性：ls-l
3. 切换目录：cd 目录名称，如 cd/data/software
4. 查看文件：cat 文件名，如 cat/data/home/zhouhao/testfile
5. 查看文件：more 文件名，如 more/data/home/zhouhao/testfile，空格键向下翻页
6. 删除文件：rm 文件名，如 rm/data/home/zhouhao/testfile
7. 删除文件夹：rm 文件夹名，如 rm-r/data/home/zhouhao/testdir
8. 编辑文件：vim 命令，vim 常用有两种模式：
 - 1) 阅读模式，用 vim 打开一个文件后就进入到阅读模式，在阅读模式下可以通过上下键调整页面阅读所打开的文件内容。
 - 2) 编辑模式，需要输入特定的键才能对文件内容进行编辑。
 - 3) 阅读模式——>编辑模式：
 - a 键：当前光标所在处的后一个字符处编辑内容
 - i 键：当前光标所在处编辑内容
 - o 键：当前光标所在行的下一行编辑内容
 - O 键：当前光标所在行的上一行编辑内容
 - 4) 编辑模式——>阅读模式：Esc 键
 - 5) 退出 vim 编辑界面，需要在阅读模式下先输入冒号，冒号后面可输入以下字符
 - w 保存不退出
 - wq 保存退出
 - wq! 保存并强制退出
 - q 不保存退出
 - q! 不保存强制退出

附录 2 常用脚本

vasp 脚本试列

```
#!/bin/bash
#SBATCH-J test
#SBATCH-p cpu
#SBATCH-n 64
#SBATCH--error=%J.err
#SBATCH--output=%J.out
module load intel/2017u5
module load vasp/5.4.4
mpirun vasp_std
scontrol show job$SLURM_JOBID
```

lammmps 脚本试列

```
#!/bin/bash
#SBATCH-J test
#SBATCH-p cpu
#SBATCH-n 64
#SBATCH--error=%J.err
#SBATCH--output=%J.out
module load intel/2017u5
module load lammmps/7Aug19
mpirun-np 80 lmp_mpi<in.cutting
scontrol show job$SLURM_JOBID
```

fluent 脚本试列

```
#!/bin/bash
#SBATCH-J test
#SBATCH-p cpu
#SBATCH-n 64
#SBATCH--error=%J.err
#SBATCH--output=%J.out
srun hostname-s>hostlist
/data/software/ansys18.0/v180/fluent/bin/fluent-ssh-g 3ddp-t64-cnfg=hostlist-mpi=intel-i
fluent.jou
scontrol show job$SLURM_JOBID
```

注：fluent.jou 文件需要单独编写，如有问题请联系集群管理员。

lsdyna 试列

```
#!/bin/bash
#SBATCH-J test
#SBATCH-p cpu
#SBATCH-n 64
#SBATCH--error=%J.err
```

```
#SBATCH--output=%J.out
unset SLURM_GTIDS
ulimit-s
module load intel/2017u5
PATH=$PATH:/data/software/ansys/17.2/v172/ansys/bin
lsdyna-np 64 memory=200000000-mpi intelmpi i=111-1-1-3-v172.k-lsdynampp
```

附录 3 modulefile（使用场景）

二期平台使用 Environment Modules 以模块的形式对环境变量进行管理。在高性能计算集群系统中，安装有多种软件及其不同版本，它们需要设置不同的环境变量，Environment Module 可以将这些环境变量做成模块文件（modulefile）。模块可以被加载(load)、卸载(unload)、切换(switch)等，这些操作会改变相应的环境变量设置，让用户方便的在不同环境之间切换。相比将环境变量设置写入 /etc/profile、~/.bashrc 或 ~/.bash_profile，Environment Module 操作只影响当前用户的当前登录环境；相比直接 source 文件，Environment Module 的操作可以撤销（卸载）。普通用户还可以自己编写 module，具有很好的定制性。用户不仅可以在命令行中 Environment Modules，也能在作业提交脚本中使用，对编译和计算环境都能够很好的控制。

首先介绍 module 常见的一些指令，接着介绍几个 module 的几个使用案例，最后介绍如何编写 modulefile 来管理自己的软件环境。

首先介绍 module 常见的一些指令：

```
module help#显示帮助信息
module avail#显示已经安装的软件环境
module load#导入相应的软件环境
module unload#删除相应的软件环境
module list#列出已经导入的软件环境
module purge#清除所有已经导入的软件环境
module switch[mod1]mod2#删除 mod1 并导入 mod2
```

接着介绍几个 module 的使用例子：

查看集群现有软件活库：

```
module avail
```

查看集群可用的 intel 版本：

```
module avail intel
```

导入 intel/2017u5 软件环境：

```
module load intel/2017u5
```

清除所有通过 module 导入的软件环境：

```
module purge
```

最后介绍如何编写 modulefile 来管理自己的软件环境。首先，创建目录用来存放自己的 modulefile：

```
mkdir${HOME}/mymodulefiles#创建目录用于放自己的 module file
```

```
echo"export
MODULEPATH=${HOME}/mymodulefiles:\$MODULEPATH">>~/.bashrc
source~/.bashrc#或者退出重新登录即可
#编写自己的第一个 module file
cd${HOME}/mymodulefiles
vim myfirstmodulefile
```

然后在创建好的目录下编写 modulefile, 假设/share/home/test/soft/gcc/7.2.0 安装了 gcc 编译器, 则可以这么编写 modulefile:

```
##Module1.0
##
##
module-whatis"my first modulefile"
set topdir"/share/home/test/soft/gcc/7.2.0"
prepend-path PATH"${topdir}/bin"
prepend-path LIBRARY_PATH"${topdir}/lib"
prepend-path LD_LIBRARY_PATH"${topdir}/lib"
prepend-path LIBRARY_PATH"${topdir}/lib64"
prepend-path LD_LIBRARY_PATH"${topdir}/lib64"
prepend-path CPATH"${topdir}/include"
prepend-path CMAKE_PREFIX_PATH"${topdir}"
setenv CC"${topdir}/bin/gcc"
setenv CXX"${topdir}/bin/g++"
setenv FC"${topdir}/bin/gfortran"
setenv F77"${topdir}/bin/gfortran"
setenv F90"${topdir}/bin/gfortran"
```

编写好后执行 module avail 即可查看到刚刚写好的 modulefile 了。下为编写 modulefile 文件常见的语法:

```
set#设置 modulefile 内部的变量
setenv#设置环境变量
prepend-path#效果类似于 export PATH=xxx:$PATH
append-path#效果类似 export PATH=$PATH:xxx
```