



冷冻电镜中心超算集群

用户操作手册(v3.3)

更新至 2026 年 04 月 01 日



目录

一、集群使用说明	1
二、登录.....	2
三、修改密码.....	3
四、上传/下载文件	3
五、加载软件模块	4
六、LSF 队列常用命令说明	6
七、提交作业到集群.....	6
八、集群已装软件说明	12
九、软件安装及存储扩容.....	13



一、集群使用说明

1. 集群环境

目前集群包括以下节点：

登录节点	172.18.5.205
计算节点	normal 队列 (c001-c100): 36 * Cores, 2 * V100 q4gpu 队列 (c101-c112): 32 * Cores, 4 * V100 qsr660 队列 (c113-c126): 56 * Cores
胖节点	smp 队列 (s001-s002): 192 * Cores

2. 使用说明

本集群为冷冻电镜实验配套支持系统，仅限南方科技大学校内用户处理在冷冻电镜中心实验所收集数据之用。用户须先注册冷冻电镜中心实验预约系统 (<https://cryoem.sustech.edu.cn/sys>) 方可申请开通集群账户。在账户创建后，须先绑定集群账户到预约系统—个人基本信息—超算用户名处，方可开始使用集群，否则会封禁；

使用须知：

- 禁止使用本集群于科研之外的其它用途；
- 禁止在登录节点运行计算任务或大内存程序，所有任务须提交到 LSF 队列运行；
- 禁止不投 job 在计算节点运行程序，比如故意遗留 session 进行进程控制；集群有监护进程进行查杀、记录，此行为将被公开警告，并视情况处罚；
- 禁止在登录节点进行高并发的海量网络与存储 I/O 操作；
- 禁止在登录节点进行大量实验数据或数据库文件的上传和下载，须自备移动存储设备找管理员帮助操作；数据库文件包括但不限于 AlphaFold、UniProt、EMPIAR、ZINC 等；
- 禁止在登录节点直接操作接近 8,000 量级数目小文件，须找管理员确定打包上传、分批、本地 SSD 处理方案。

违反上述规定将视情况被警告、临时或永久冻结集群使用权限。

3. 免责条款

用户开通账户、使用本集群，即默认认可以下免责条款：

- 本集群仅提供计算资源的有偿使用，但不保证硬件的完好、资源的随时可用性；
- 本集群不提供任何数据冗余备份，不对数据安全、因任何原因导致的数据损失负责。用户须自行对重要数据做及时转移、备份；
- 中心集群保留视管理需要处置一切用户账户、文件、数据的权力（包括但不限于无预警终止进程、踢出登录、删除数据、冻结或删除账户等操作）；
- 集群账户使用者在提交协助请求之前，应主动同课题组 PI 或其他利益相关方沟



通并得到授权；用户提交的任何请求，一概视为已经得到充分授权；中心采取任何协助操作时，无义务再征求任何他人意见，且对产生的后果不负任何责任；

- 中心对遗留进程进行积极监控查杀，但仍有概率新投 job 受短时影响；用户须对此可能情况知晓并理解，介意者请勿使用本集群。

4. 集群收费

中心集群的费用收取，来自【作业费+存储扩容费+账户维护费】三个部分，具体收费标准参考(<https://cryoem.sustech.edu.cn/service/charges.html>)。

特别说明：

(1) 账户维护费用为每课题组 2000 元/年，包括了账户维护管理、日常使用帮助和默认 10TB 的存储空间，原则上是在课题组开户时及每周年一次性收取，中心根据冷冻电镜主流方向—结构解析实验的费用特点，采取逐月收取方式。故，即使未提交作业且未申请扩容，只要集群账户使用状态有效/在使用集群空间，每月即会产生固定的账号维护费账单。如课题组不再使用超算集群/希望停止账号维护，须在清空集群所有账户数据后告知我们，管理员将对课题组账户进行删除操作；如课题组未尽告知义务，则默认其集群使用状态有效；

(2) 平台不对用户提交到队列的程序及计算内容是否正确运行负责，仅依据用户作业实际占用队列资源大小及时长收取费用。用户有责任及时关注所提交作业进度、确保程序/作业正确运行。如发生程序异常导致作业未及时终止，平台仍将按照队列资源实际占用收取费用。

注：集群账户开通、使用本集群，即默认课题组知晓并认可以上内容、对此类费用收取不持异议。

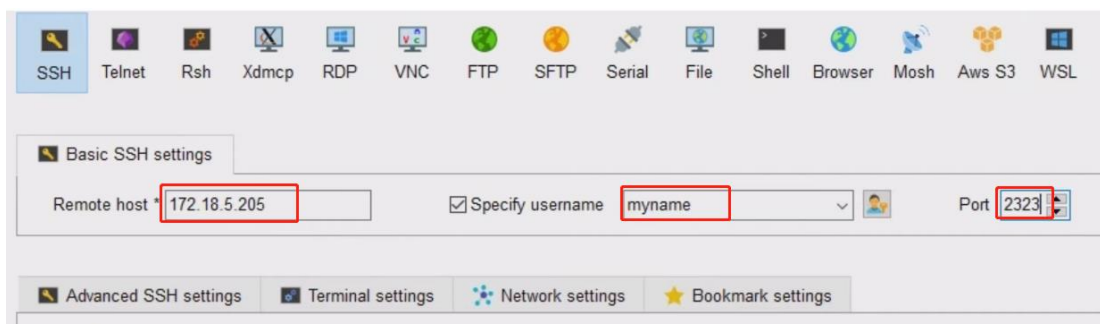
二、登录

集群登录节点 IP 地址为 172.18.5.205，端口为 2323。需在校园网环境下或与南科大 VPN 配合使用。

Linux 系统可使用如下命令登录：

```
$ ssh -X -p 2323 myname@172.18.5.205
```

Windows 系统可以使用 Mobaxterm/Xshell/Termius 等客户端工具登录。



注意：类似 Mobaxterm 等客户端软件一般有记住密码选项，务必保证密码输入正确，否则会因软件频繁尝试连接服务器而被视为暴力攻击，导致账户被系统封禁。如遇此情况，请与管理员联系。

MacOS X 用户安装 Xquartz 以启用 X11 转发功能。登录命令如下：

```
$ ssh -Y -p 2323 myname@172.18.5.205
```

命令中的 -X/-Y 参数用于启用 GUI 图形界面，如果需要打开 Relion 之类的图形应用，须加上该参数。

三、修改密码

新用户会被分配一个默认密码，第一次登录后需要修改自己的密码。

命令：passwd

注意，密码必须满足一定的复杂度要求：包括数字、字母和特殊字符的组合。我们在使用中发现很多用户的计算机中了挖矿病毒，故请务必定期每半年左右更改一次密码，以保证自己账户的安全。

四、上传/下载文件

1. 一般文件的转移

注意：以下方法仅适用于少量数据(<1GB)的上传和下载。大量实验数据的下载请自备移动硬盘并联系管理员。

Linux/MacOS 用户：

scp

示例. 将本地的 test.txt 文件上传到集群 myname 账户的 home 目录：

```
scp -P 2323 test.txt myname@172.18.5.205:~
```

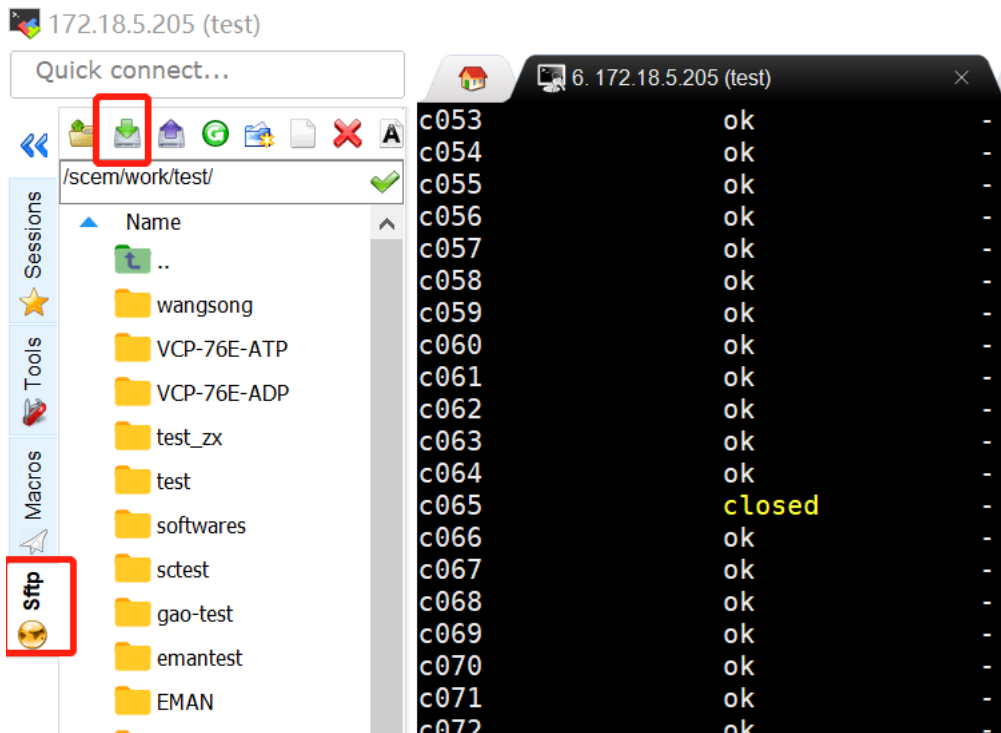
注意，参数 -P 为大写，ssh 命令中 -p 为小写。

Windows 用户：

sftp

mobaxterm 和 xshell 都自带 sftp 客户端，可以很方便的上传和下载。

Mobaxterm 示例：



2. 冷冻电镜实验数据的转移

在冷冻电镜中心所产生实验数据，已同步 mount 到中心控制间的预处理工作站中，由 EMShark 软件实时作 motion 处理/确认。EMShark 处理后的数据，可通过 cp.sh 或 cp.py 脚本 copy 到用户存储设备，或经过内网 IP mount 磁盘上传到集群进行下一步计算处理。具体 EMShark 使用方法及数据 copy，由电镜设备负责老师在 300 kV 电镜培训时统一讲授。转移数据时遇到任何问题，请务必向电镜设备对应负责老师寻求协助，切记不要盲目操作。

集群处理后的数据，用户须自带移动存储设备找中心管理老师协助 copy，禁止直接通过登录节点 172.18.5.205 远程下载。

五、加载软件模块

中心集群通过 module 或 conda 管理软件环境，大部分场景是二者结合使用。说明如下：

1. module 方式使用

以 relion 使用为例，执行以下命令添加 relion 的相关变量：

```
module load relion/3.0.6-gpu-cuda8.0-gcc455-mpi314
```

当我们需要切换到 eman2 时，执行：



```
module purge    #清空加载的模块列表
module load eman2/2.3
```

常用的 `module` 命令如下：

```
查看帮助信息：module help
查看可用软件模块：module avail
加载软件模块：module load MODULE
卸载软件模块：module unload MODULE
查看当前环境已加载的软件模块：module list
清除所有已加载的软件模块：module purge
```

更多 `module` 知识及相关命令，请查阅 <https://modules.readthedocs.io/en/latest/>

2. conda 方式使用

以 `miniconda3` 使用为例，执行以下命令添加 `miniconda3` 的相关变量：

```
module load miniconda3
```

常用的 `conda` 命令如下：

```
查看 conda 帮助信息：conda --help
查看已有的 conda 环境：conda env list
进入指定的 conda 环境：conda/source activate env_name
退出当前的 conda 环境：conda deactivate
创建 conda 环境：conda create -n env_name [python=<version>]
指定环境安装软件包：conda install pkg_name -n env_name
删除 conda 环境：conda env remove -n env_name
```

更多 `conda` 知识及相关命令，请查阅 <https://docs.conda.io/en/latest/>

3. module 与 conda 结合使用

大部分软件，皆用 `module` 进行管理和加载。愈来愈多依赖 Python 库的软件是通过 `conda` 运行，使用时先加载对应 `module` 模块，然后激活对应 `conda` 环境（`module load` 加载软件后，一般会有进一步提示如何激活）。示例如下：

```
module load pyem    #加载软件模块
source activate pyem #进入软件环境
```



六、LSF 队列常用命令说明

中心集群使用 LSF 队列管理系统进行作业调度。常用 LSF 命令：

- **bjobs**: 查看自己已经提交的任务
- **bjobs -l JOBID**: 查看任务的详细信息
- **bpeek JOBID**: 查看任务输出
- **bqueues**: 查看集群队列状态
- **bhosts**: 查看集群节点状态
- **bhist**: 查看历史作业记录
- **bkill**: 终止正在运行的作业

提交作业后，您可能需要检查其进度或取消它。通过调用 **bjobs -l <jobID>** 可以看到大多数作业的特征。

提交作业后，您可能需要查看作业的屏幕打印输出，通过调用 **bpeek -f <jobID>** 可以看到程序运行过程中的屏幕输出。

可以使用 **bkill** 命令（**bkill <jobID>**）取消挂起的作业（从队列中撤消）。**bkill** 命令也可用于终止正在运行的作业。默认行为是将作业发出 **SIGTERM**，等待 30 秒，如果作业中的进程继续运行，则发出 **SIGKILL** 命令。**bkill** 命令的 **-s** 选项（**bkill -s <signal> <jobID>**）允许用户向正在运行的作业发出任何信号。

LSF 作业的基本工作状态是：

- **PEND** - 作业在队列中，等待安排
- **PSUSP** - 作业已提交，但已处于暂停状态（不符合运行条件）
- **RUN** - 已授予作业分配。如果是批处理作业，则批处理脚本已运行
- **DONE** - 作业已成功完成

七、提交作业到集群

1. Relion 作业

1.1 提交 Relion 作业

提交到 LSF 队列的 Relion Running 参数和单机版不同，具体不同点说明如下：

“Number of MPI procs”: 将要申请的节点数*36，单个任务限制节点数≤4

“Submit to queue”: Yes

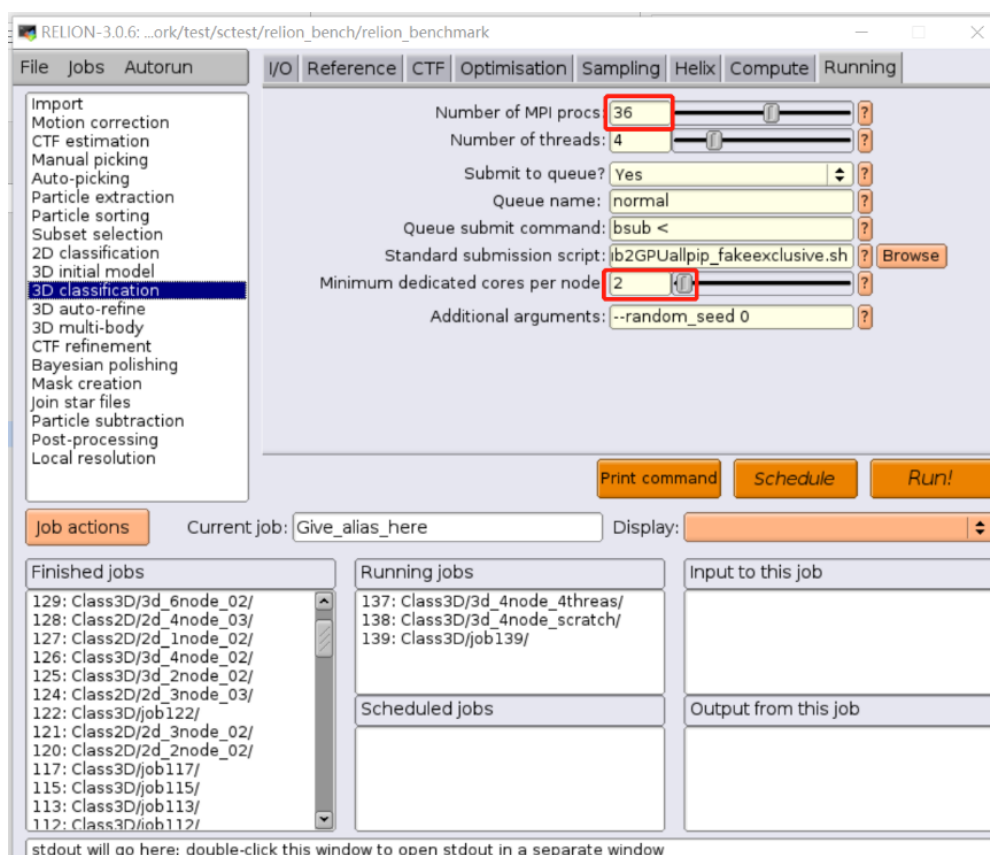
“Queue name”: normal

“Queus submit command”: bsub <

“Standard submission script”:

/share/apps/relion/3.0.6-gpu-cuda8-gcc485-ompi314/bin/bsub2GPU_fakeexclusive.sh

“Minimum dedicated cores per node”: 每个节点分配的 MPI 数量，一般等同于显卡数量



1.2 使用 scratch 目录

目前集群所有节点均配置了 960G 的 SSD 盘做 Relion 的 scratch 目录，经测试，可有效提高 Relion 计算速度。在使用 Relion 时在 Compute 选项中 Copy particles to scratch directory 中添加目录/gpu_temp 即可使用。



File Jobs Autorun I/O Reference CTF Optimisation Sampling Helix Compute Running

Import
Motion correction
CTF estimation
Manual picking
Auto-picking
Particle extraction
Particle sorting
Subset selection
2D classification
3D initial model
3D classification
3D auto-refine
3D multi-body
CTF refinement
Bayesian polishing
Mask creation
Join star files
Particle subtraction
Post-processing
Local resolution

Use parallel disc I/O? Yes
Number of pooled particles: 100
Skip padding? No
Pre-read all particles into RAM? No
Copy particles to scratch directory: /gpu_temp
Combine iterations through disc? No
Use GPU acceleration? Yes
Which GPUs to use:

Print command Schedule Continue!

Job actions Current job: 067: Class3D/2node_0429_ssd/ Display:

Finished jobs
084: Class3D/job084/
068: Class3D/2node_0429/
067: Class3D/2node_0429_ssd/
066: Class3D/2node_0428/
065: Class3D/2node_0428_ssd/
064: Class3D/2node_0424/
063: Class3D/2node_0424_ssd/
062: Class3D/2node_0423/
061: Class3D/2node_0423_ssd/
052: Class3D/1node_0423_ssd/
038: Class3D/4c_nopre_2node_2/
037: Class3D/4C_nopre_1node_2/

Running jobs
073: Class3D/job073/
074: Class3D/job074/
075: Class3D/job075/
076: Class3D/job076/
077: Class3D/job077/

Scheduled jobs

Input to this job

Output from this job

PS:
Read file <Class3D/job067/run.err> for stderr output of this job.

2. CryoSPARC 作业

2.1 CryoSPARC 服务管理

CryoSPARC 软件因 License 及端口资源占用要求,须以课题组为单位单独申请安装,课题组提供 License、安装的集群账户和安装邮箱,由管理员进行安装和端口分配。安装目录在用户 home 目录,如 /scem/work/qirx/cryosparc,相应登录地址为: <http://172.18.5.205:XXXXXX>,其中 XXXXXX 为分配的端口号。

服务启动和关闭方法:

```

cryosparcm stop          #关闭 cryosparc 服务
需要使用时执行:
cd ~/cryosparc/cryosparc_master/  #进入 master 目录
cryosparcm start         #启动 cryosparc 服务
cryosparcm cluster connect  #添加集群服务连接
cryosparcm status        #查看服务状态

```



2.2 常见错误及解决方法

注意：(1) 用户遇到问题请求协助之前，务必已先尝试如下修复措施，无效后再同返回错误代码与管理员联系；(2) 用户须定时重启 CryoSPARC 服务，以清理过大的 web 缓存、内存占用；(3) CryoSPARC 已知 bug 可能导致计算任务结束时队列作业仍未及时终止，用户须及时关注作业进度、必要时手动终止作业以避免产生额外作业费用。

2.2.1 服务启动问题

一般 CryoSPARC 服务无法启动或其它错误，皆可通过清理驻留进程和删除遗留文件来解决。若尝试后服务依旧启动失败，可联系管理员获得进一步帮助。

服务重启步骤：

- ① 确保课题组使用空间未超出磁盘配额：`mmlsquota -g <gid> --block-size auto`
- ② 手动停掉所有 cryosparc 队列任务：`bkill JOBID` (bjobs 可查看自己已提交的任
务)
- ③ 进入 master 目录：`cd ~/cryosparc/cryosparc_master/`
- ④ 关闭 cryosparc 服务：`cryosparcm stop`
- ⑤ 清理驻留进程：`ps -eo pid,user,cmd | grep qirx | grep 'cryosparc\|mongo' | awk '{print $1}' | xargs kill -9` (此处 qirx 更换为实际用户名)
- ⑥ 删除遗留文件：`ls -l /tmp/mongodb-*.sock /tmp/cryosparc-supervisor-*.sock | grep qirx | awk '{print $9}' | xargs rm` (此处 qirx 更换为实际用户名)
- ⑦ 启动 cryosparc 服务：`cryosparcm start`
- ⑧ 添加集群服务连接：`cryosparcm cluster connect`
- ⑨ 查看服务状态：`cryosparcm status`

2.2.2 SSD 缓存问题

在运行 CryoSPARC 计算过程中，如出现“SSD cache : requested file are locked for past 488s, checking again in 5s”等相关报错，可尝试以下方法解决。更多故障解决方法请参考：<https://guide.cryosparc.com/setup-configuration-and-management/troubleshooting>

修复数据库步骤：

- ① 手动停掉所有 cryosparc 队列任务：`bkill JOBID` (bjobs 可查看自己已提交的任
务)
- ② 重启 cryosparc 服务：`cryosparcm restart`
- ③ 进入交互式数据库提示符：`cryosparcm mongo`
- ④ 输入以下命令检查有多少条记录处于不一致状态：`db.cache_files.find({status:
{ $nin: ['hit', 'miss'] }).length()`
- ⑤ 如果结果不是 0，则输入以下命令修复：`db.cache_files.updateMany({status: { $nin:
['hit', 'miss'] }, { $set: {status: 'miss'}})`
- ⑥ 退出数据库提示符：`Ctrl + D`



3. 一般程序作业

一般计算任务提交到 LSF 队列，用户可以自行编辑作业脚本，将自己要运行的命令及必要的环境加载写入，正常 `bsub` 提交即可。集群提供模板文件 `lsf_template.sh`（模板文件保存在“`/share/apps/lsf_template.sh`”，用户可自行复制到自己 `home` 目录），用户可在此基础上修改，然后使用“`bsub < lsf_template.sh`”提交到队列。

具体见如下两个示例：

(1) cryoemsimpletools 工具计算示例：

```
#!/bin/bash
#BSUB -J MyJob          #任务名称
#BSUB -q normal         #队列名称
#BSUB -o %J.stdout      #输出文件
#BSUB -e %J.err         #错误输出文件
#BSUB -n 36             #申请 36 cores，即一个 normal 节点
#BSUB -R "span[ptile=36]" #按照每个节点 36 核心来分配 slot

module load cryoemsimpletools/2019 #加载 cryoem 的小工具包，包含 ctffind、
Gautomatch、Gctf、MotionCor2 等
Gautomatch xxx #运行 Gautomatch，或使用 MPI 并行：
mpirun -n 2 Gautomatch xxx #并行运行两个 Gautomatch 进程
```

(2) GisSPA 计算示例：

```
#!/bin/bash
#BSUB -J GisSPA          #任务名称
#BSUB -q q4gpu           #队列名称
#BSUB -o %J.stdout      #输出文件
#BSUB -e %J.err         #错误输出文件
#BSUB -n 32             #申请 32cores，即一个 q4gpu 节点
#BSUB -R "span[ptile=32]" #按照每个节点 32 核心来分配 slot

module load GisSPA_v2.0 #加载 GisSPA 相关环境
/share/apps/GisSPA_v2.0/build/main configx #运行 GisSPA
```



4. 交互作业

如需交互执行某些程序，如批量处理数据或以图形界面打开程序，则可以提交一个独占节点作业，ssh 到节点进行操作。

具体使用，可以参考下列步骤进行：

1) 提交下面的脚本，会申请一个节点 24 小时占用

```
[test@mn02 ~]$ bsub < /share/apps/sleep24hr.sh
Job <4523> is submitted to queue <normal>.
```

2) 使用 bjobs 查看申请的节点名称。如图中红框出，这次申请到的是 c007 节点。

```
[test@mn02 ~]$ bjobs
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
4522   test   RUN   normal  mn02       36*c086    *D/job144/ Jul 22 12:57
        36*c037
        36*c057
        36*c066
4523   test   RUN   normal  mn02       36*c007    Sleep24hr  Jul 22 13:46
```

3) ssh 登录节点，加载环境变量，即可像单机一样进行测试

```
[test@mn02 ~]$ ssh -X c007
Last login: Mon Jul 22 13:52:32 2019 from mn02
[test@c007 ~]$
[test@c007 ~]$ module purge
[test@c007 ~]$ module load cryoemsimpletools/2019
[test@c007 ~]$ Gautomatch --help
```

4) 在执行命令时，须用 **nohup ... &** 方式提交到后台运行，避免依赖登录窗口，或不合理占用登录节点 CPU 资源导致被管理进程杀掉。如：

```
nohup /share/apps/GisSPA_v2.0/build/main configx &
```

5) 测试完成后，退出节点，结束掉这个任务。

```
[test@c007 ~]$ exit
logout
Connection to c007 closed.
[test@mn02 ~]$
[test@mn02 ~]$ bjobs
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
4522   test   RUN   normal  mn02       36*c086    *D/job144/ Jul 22 12:57
        36*c037
        36*c057
        36*c066
4523   test   RUN   normal  mn02       36*c007    Sleep24hr  Jul 22 13:46
[test@mn02 ~]$ bkill 4523
Job <4523> is being terminated
```

注意：

(1) 此方式仅限于临时交互使用，用户用完须立刻 **bkill** 杀掉，防止遗留 job 导致额外扣费，或 job 提前结束而导致遗留计算进程被集群管理进程杀掉并加入惩戒名单。用户应尽量使用前述正常 job 提交方式运行任何计算；

(2) 不要在 sleep 文件中设置过大时间长度，以适合自己的计算任务为宜。

八、集群已装软件说明

1. 集群软件列表

集群已安装绝大多数常见 Cryo-EM 科研软件，以及越来越流行的 Cryo-ET 科研相关软件。执行 `module avai` 或 `module show xxx` 可查看可用软件或具体软件信息。

附软件安装列表：

```
[root@mn02 ~]# module avail

----- /share/base/modulefiles/compilers -----
gcc/12.4.0      gcc/8.3.0      go/1.11.1      intel/2017.8    intel/2018.4
gcc/8.2.0(default) gcc/9.1.0      intel/2015.6    intel/2018.3    intel/2019.1

----- /share/base/modulefiles/parallel -----
ipsxe/2018.3.222      mpi/intel/2018.4      mpi/openmpi/3.1.1rc1_gcc      mpi/openmpi/4.0.1_gcc485
itac/2018.3.022      mpi/intel/2019.1      mpi/openmpi/3.1.4_cuda10.1      mpi/openmpi/4.0.2_cuda10.1
knem/1.1.3.90lnx1      mpi/openmpi/1.8.5_gcc485      mpi/openmpi/3.1.4_cuda8      mpi/openmpi/4.1.2_cuda11.1
mpi/intel/2017.8      mpi/openmpi/3.1.0_cuda10.1_gdr      mpi/openmpi/3.1.4_cuda9.2      mpi/openmpi/4.1.6_gcc910
mpi/intel/2018.3      mpi/openmpi/3.1.0_gcc      mpi/openmpi/3.1.4_cuda9.2_gdr

----- /share/base/modulefiles/bio -----
alphafold      emclarity/1.5.0      relion/3.0.5-gpu-cuda9.2-gcc485-mpi401
alphafold3      emclarity/1.5.3      relion/3.0.5-gpu-cuda9.2-intel17.8
aretomo2      emready      relion/3.0.6-altcpu-intel2018.4
aretomo3      imod/4.10.46      relion/3.0.6-gpu-cuda10.1-gcc485-mpi314
chimera/1.13.1      imod/4.12.56      relion/3.0.6-gpu-cuda8.0-gcc455-mpi314
chimera/1.17.1      imod/4.9.12      relion/3.0.6-gpu-cuda9.2-gcc485-mpi314
cistem/1.0.0-beta      imod/5.0.1      relion/3.0.8-gpu-cuda10.1-gcc485-mpi402
cistem/2.0.0      matlab_runtime/2017b      relion/3.1.0-altcpu-intel2018.4
cryocare_10      matlab_runtime/2019a      relion/3.1.0-gpu-cuda10.1-gcc485-mpi402
cryocare_11      matlab_runtime/2020b      relion/3.1.2-gpu-cuda10.1-gcc485-mpi402
cryodrgn/3.4.2      matlab_runtime/2022b      relion/3.1.2_SIRM
cryoemssimpletools/2019      model-angelo      relion/3.1_beta
cryo-mem      peet/1.17.0      relion/4.0.0-gpu-cuda10.1-gcc485-mpi402
deepdwedge      protomo      relion/4.0_beta
dynamo      pym      relion/5.0_beta0-gpu-cuda11.1-gcc830-mpi412
eman2/2.21      relion/1.4-cpu-gcc4.8.5-mpi3.1.0      relion/5.0_beta4-gpu-cuda12.2-gcc910-mpi416
eman2/2.3      relion/2.1beta-cuda8.0-gcc4.8.5-mpi314      relion/5.0_stable-gpu-cuda12.2-gcc910-mpi416
eman2/2.9      relion/2.1-gpu-cuda8.0-gcc485-mpi314
eman2/2.99.47      relion/3.0.3-altcpu-intel17.8
embuild      relion/3.0.3-gpu-cuda9.2-intel17.8
emclarity/1.1.7      relion/3.0.5-altcpu-gcc485-mpi311
emclarity/1.3.0      relion/3.0.5-altcpu-intel17.8
emclarity/1.4.3      relion/3.0.5-gpu-cuda9.2-gcc485-mpi311
warp/2.0dev29

----- /share/base/modulefiles/libraries -----
boost/1.64.0/mpi      fftw/3.3.8-intel-double-avx512-mpi      isl/0.19-gcc-8.2.0      netcdf-c/4.6.2-gcc-4.8.5
boost/1.66.0/mpi      fftw/3.3.8-intel-single-avx512      metis/4.0.3      netcdf-cxx/4.3.0-gcc-4.8.5
boost/1.67.0/mpi      fftw/3.3.8-intel-single-avx512-mpi      metis/5.1.0/float32bit      netcdf-fortran/4.4.4-gcc-4.8.5
cudnn/cudnn-v7.6.1      gmp/6.1.2-gcc-4.8.5      metis/5.1.0/float64bit      petsc/3.10.3-intel-mpi-double
cudnn/cudnn-v8.0.2      hdf4/4.2.14-gcc-4.8.5      mkl/2018.3      petsc/3.10.3-intel-mpi-single
curl/7.63.0(default)      hdf5/1.10.4-gcc-4.8.5      mpc/1.1.0-gcc-4.8.5      pnetcdf/1.10.0/mpi
fftw/2.1.5      hdf5/1.10.4-intel-18.4      mpfr/4.0.1-gcc-4.8.5      spglib/1.9.7
fftw/3.3.8      hwloc/2.0.2      mpfr/4.0.1-gcc-8.2.0      szip/2.1.1
fftw/3.3.8-intel-double-avx512      isl/0.19-gcc-4.8.5      nccl/nccl_2.8.3-1-cuda10.1_x86_64      zlib/1.2.11(default)

----- /share/base/modulefiles/tools -----
autoconf/2.69      cuda/9.0      ior/3.1.0      python/2.7.18
autogen/5.18.12      cuda/9.2      iozone/3.482      python/3.7.0
autogen/5.18.4      darshan-runtime/mpi/2.3.1      IsoNet      python/anaconda2/5.2.0
binutils/2.31-gcc-4.8.5      darshan-runtime/mvapich2/2.3.1      itac/2018.3.022      python/anaconda3/5.2.0
cmake/3.12.2      darshan-util/mpi/2.3.1      java/10.0.2      python/intelpython2/2018.3.039
cmake/3.28.1      darshan-util/mvapich2/2.3.1      java/1.8.0_181      python/intelpython2/2019.0.047
collectl/4.1.3      deepEMhancer      jube/2.2.1      python/intelpython3/2018.3.039
collectl/4.3.0      dstat/0.7.3      libtool/2.4.6      python/intelpython3/2019.0.047
cryolo      gccam/1.0      m4/1.4.17      R/3.5.1
cryosieve      GtsSPA      mc/4.2.14      rest
CTFMeasure      GtsSPA_v2.0      miniconda2      ruby/2.5.0
cuda/10.0      git/2.18.0      miniconda3      scipion
cuda/10.1      gnuplot/6.5.4      mpicker      scipio/0.16.0
cuda/11.1      gnuparallel/20180922      mpitrace/0.11-b78a27b-mpi      scipio/0.16.1
cuda/11.8      health/clock/2019.0      nmon/15h      topaz
cuda/12.2      hp2p/1.0      pap/5.6.1-mpi      valgrind/3.14.0
cuda/8.0      htop/2.2.0      pdsh/2.33

----- /usr/share/Modules/modulefiles -----
dot      module-git      module-info      modules      null      use.own

[root@mn02 ~]#
```

2. cryoem 小工具

加载对应的 CUDA 环境即可，包含 `ctffind`、`Gautomatch`、`Gctf`、`MotionCor2` 等。路径位置：`/share/apps/relion/deps/`。

```
[root@mn02 ~]# ls /share/apps/relion/deps/
bin      Gctf_v1.18_b2_sm60_cu8.0      MotionCor2-1.0.0      MotionCor2_1.6.3_Cuda111_Feb18_2023      summovie_1.0.2
ctffind      Gctf_v1.18_b2_sm60_cu9.1      MotionCor2-1.2.3-Cuda100      MotionCor2-UserManual-09-13-2018.pdf      summovie_1.0.2.tar.gz
ctffind-4.1.8-linux64.tar.gz      Gctf_v1.18_b2_sm61_cu8.0      MotionCor2-1.2.3-Cuda80      MotionCor2-UserManual-10-22-2019.pdf      unblur_1.0.2
ctffind_plot_results.sh      Gctf_v1.18_b2_sm61_cu9.1      MotionCor2-1.2.3-Cuda91      MotionCor2_v1.3.1-Cuda101      unblur_1.0.2.tar.gz
Gautomatch_v0.56_sm30-75_cu10.1      Gctf_v1.18_b2_sm70_cu9.2      MotionCor2-1.2.3-Cuda92      MotionCor2_v1.3.1-Cuda102
Gautomatch_v0.56_sm60_cu8.0      GCTF_v1.18_sm30-75_cu10.1      MotionCor2-1.4.2_Cuda101      MotionCor2_v1.3.1-Cuda92
Gautomatch_v0.56_sm62_cu8.0      hub      MotionCor2-1.6.3_Cuda101_Feb18_2023      ResMap-1.1.4-linux64
```



九、软件安装及存储扩容

1. 软件安装

具有广泛公共价值的软件安装推荐，可发邮件至 chenhaijia@skybility.com（抄送 qirx@sustech.edu.cn）。我们鼓励用户提交具有公共使用价值的软件安装申请，以方便集中管理并节约资源。软件使用过程中如遇问题，或有任何意见及建议，也请发邮件件至上述邮箱。

注意：中心集群仅负责安装并维护广泛使用、有公共价值的软件；因不同用户所用软件五花八门，用户课题组个别特殊需求的软件，须自行安装在 **home** 目录下，中心无义务、不提供安装或维护服务。

2. 存储扩容

每课题组在开户时分配 10TB 的存储配额。默认存储空间不够时，可以发送邮件至前述邮箱申请扩容（注意价格是 12 元/(100GB*月)）。使用以下命令可以检查课题组磁盘配额使用情况：

```
mmquota -g <gid> --block-size auto
```

其中，<gid>为课题组 ID，可以用“id”命令获取。

鉴于冷冻电镜实验的高存储消耗特点以及用户数量的持续增长，中心所配存储空间十分有限。因此我们鼓励用户及时转移集群中已处理数据，并尽量自备小型存储如 NAS 磁盘阵列系统，以节约存储资源，同时也节约可观存储费用开销。