

生物电镜平台电镜集群用户操作手册

一， 电镜集群用户计算资源配置

1， 管理/登录节点

- ip: 10.15.56.103

- 登录方法:

打开 terminal 终端软件，windows 电脑推荐使用 MobaXterm，mac 或 linux 电脑使用自带 terminal

- 通过云桌面登录: ssh 账号@10.15.56.103

- 通过校园网登录: ssh 账号@10.15.56.103 -p 10086

- 注意事项:

管理节点上

- 允许:

- 浏览/移动/拷贝/编辑 文件

- 编辑作业脚本

- 通过 Slurm 提交作业

- 检查作业状态

- 不允许:

- 运行作业

- 多进程下载大量数据

2， 计算节点

- CPU 节点: 80 个，

- 每节点两个 **cpu**，每节点含 **28** 个核
- 每 **20** 个节点分为一个队列，分别为 **cpu-1**，**cpu-2**，**cpu-3**，**cpu-4**
- **GPU 节点：8* 4 卡 P100 节点 & 6* 4 卡 P40 节点。**
 - **P100** 卡的前 **6** 个节点为 **GPU-A** 队列，后两个节点分别为 **cryosparc-1** 与 **cryosparc-2** 队列。
 - **P40** 节点为 **GPU-B** 队列。
- **FAT 节点：一个，**
 - **64 核，2TB 内存**
 - 队列名 **FAT**

二， 修改账户密码的方法：

为了安全考虑，用户在集群上的账户在首次登陆之后应该及时修改密码，修改密码的方式：

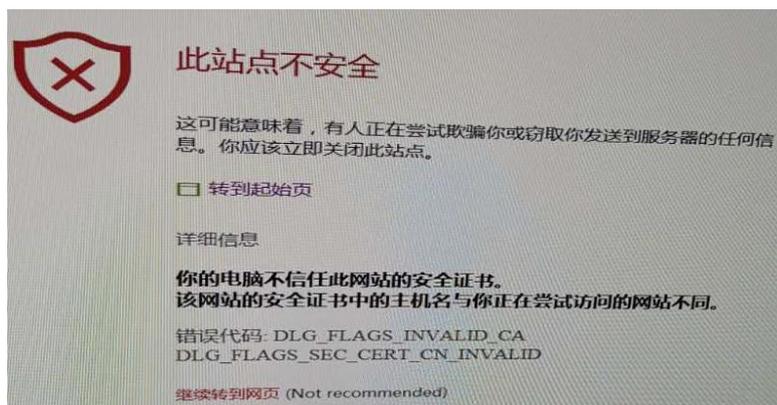
■ 云桌面平台方式：

登录电镜平台云桌面，账户是默认的，密码是 **Abc123!**

进入云桌面系统后点击浏览器，并输入管理节点 ip: **10.15.56.103**



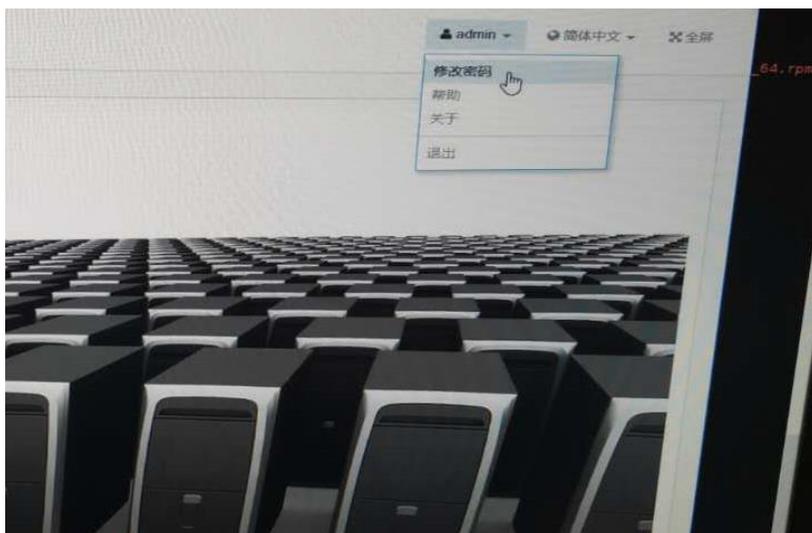
点击 详细信息：



点击 继续转到网页，出现以下 CHES 系统登录界面：



输入集群的用户账户以及密码，进入 chess 管理网页，在右上角账户下面选择修改密码，进行密码修改操作：



三， 用户提交作业方式

提交作业分为三种方式：CHESS 平台提交、终端文本提交、应用软件 UI 界面提交。首先无论以何种方式提交作业，应用软件必须先安装在集群上。

下面针对三种方式进行详细描述：

■ CHESS 平台提交作业

1.1 使用云桌面打开浏览器，输入 <https://10.15.56.103> 进入登录页面，输入账户名及密码

1.2 新建作业，点击“作业调度” — “作业” — “新建作业”

1.3 查看作业

1.4 删除作业

■ 终端文本提交

2.1 登录终端，使用 `mobaxterm` 选择 `ssh` 然后使用个人的账户登录集

群管理节点

2.2 编写提交脚本，脚本名称可以自定义，这里使用 1.slurm

脚本内容为：

```
#!/bin/bash
#SBATCH -J test    ###指定作业名称
#SBATCH -p GPU-A  ### 指定队列名称
#SBATCH -t hh:mm:ss  ### 指定本次作业运行的最大时长
#SBATCH --gres=gpu:tesla:x  ###指定使用 gpu 的个数，x 为数字
#SBATCH -n x    ###指定使用的核数
#SBATCH -N x    ###指定使用的节点数

#Command
Mpirun -n x --machinefile ./host.txt relion
```

2.3 提交作业

```
sbatch 1.slurm
```

输出信息为：

```
[reliantest@pre_mgmt01 ~]$ sbatch 1.slurm
Submitted batch job 134 → JobID
```

2.4 查看作业，根据上面的输出查找对应的 jobID

```
squeue
```

2.5 查看作业详情

```
scontrol show job <job_id>
```

2.6 删除指定作业

```
scancel <job_id>
```

■ 应用软件 GUI 界面提交

需要根据软件特性做相应配置，具体见后面几种常用软件提交作业用法。

四， relion 在集群中的使用方法

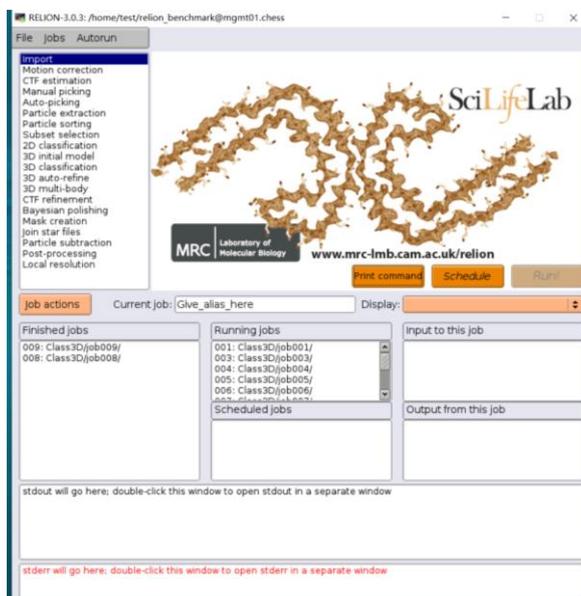
目前集群中安装的 relion 版本有：relion2.0, relion3.03, relion3.1-beta。

1， 登录集群，调用 relion 相应模块

```
cistem          cuda/8.0          cuda/9.2          fftw-3.3.8      intel2018
[test@mgmt01 ~]$ module list
No Modulefiles Currently Loaded.
[test@mgmt01 ~]$ module load relion/3.03_release
[test@mgmt01 ~]$
```

2， 进入到数据所在的文件夹，调用 relion。注意：一定要 cd 到要计算的数据所在文件夹后再调用 relion！

```
[test@mgmt01 ~]$ ls
ncin.sh  relion_benchmark  slurm.txt
[test@mgmt01 ~]$ cd relion_benchmark/
[test@mgmt01 relion_benchmark]$ relion
```



3， 提交作业方法：

- a) 实例一：以其中一个 **3D classification** 任务为例，以下参数设置方法是使用 GPU-A 队列中的两个节点，并且申请所有的八张 GPU 卡。

RELION-3.0.3: /home/test/relion_benchmark@mgmt01.chess

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:

Combine iterations through disc? No

Use GPU acceleration? Yes

Which GPUs to use: 0:0:1:1:2:2:3:3

Print command Schedule Continue!

Job actions Current job: 009: Class3D/job009/ Display:

Finished jobs	Running jobs	Input to this job
009: Class3D/job009/ 008: Class3D/job008/	001: Class3D/job001/ 003: Class3D/job003/ 004: Class3D/job004/ 005: Class3D/job005/ 006: Class3D/job006/ 007: Class3D/job007/	
	Scheduled jobs	Output from this job

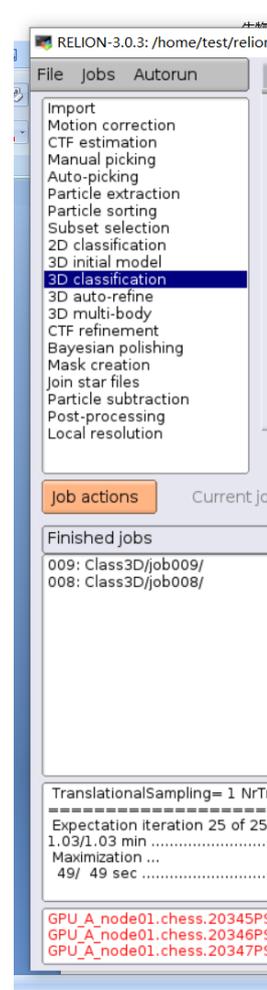
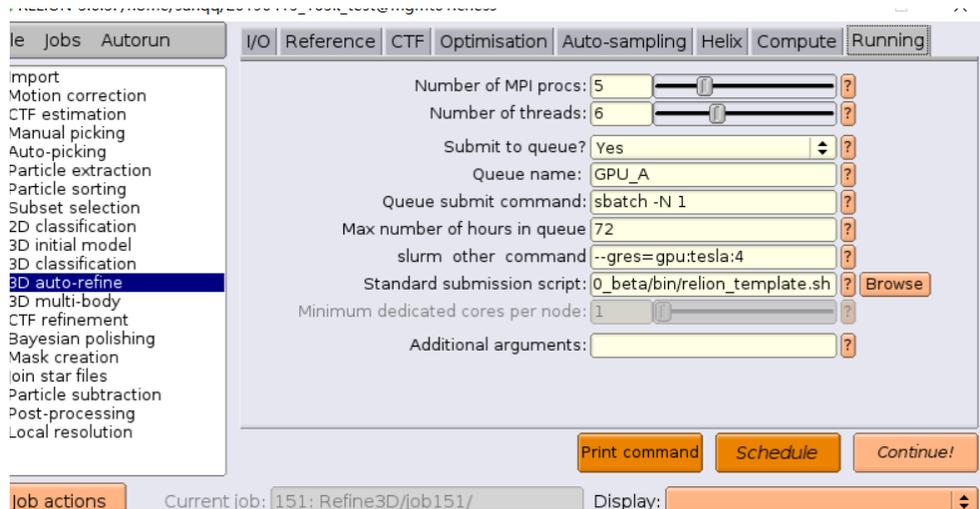
TranslationalSampling= 1 NrTranslations= 84

```

=====
Expectation iteration 25 of 25
1.03/1.03 min .....~~(,_">
Maximization ...
49/ 49 sec .....~~(,_">

```

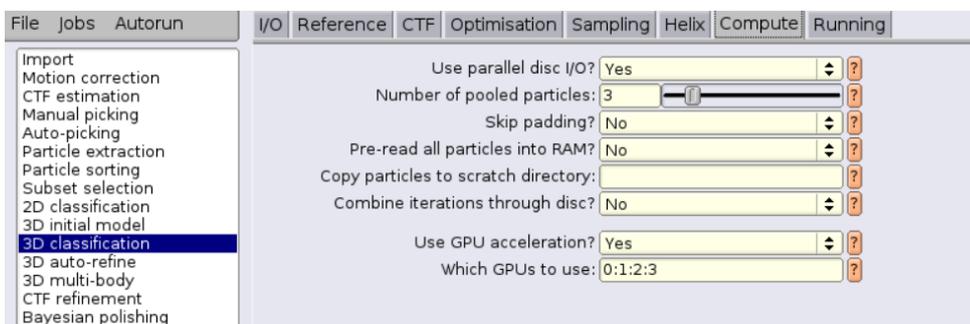
GPU_A_node01.chess.20345PSM2 no hfi units are available (err=23)
GPU_A_node01.chess.20346PSM2 no hfi units are available (err=23)
GPU_A_node01.chess.20347PSM2 no hfi units are available (err=23)



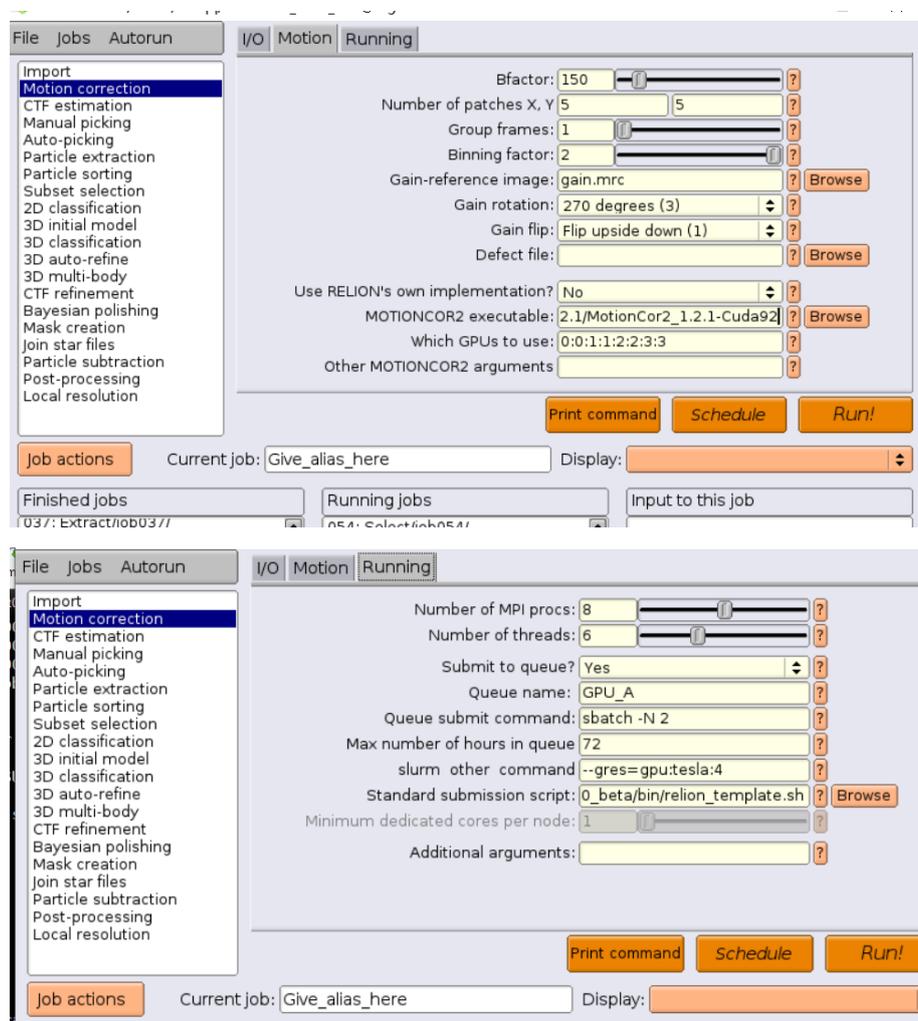
关键参数：

Submit to queue? 一定要选 Yes，如此该任务才会提交到计算节点上运行，否则会在管理节点运行。

如果申请一个 GPU 节点，相应参数改为：



b) 实例 2: 以 motion correction 任务为例, 以下任务申请两个 GPU 节点, 使用 8 张卡



五, cisTEM 在集群中的使用方法

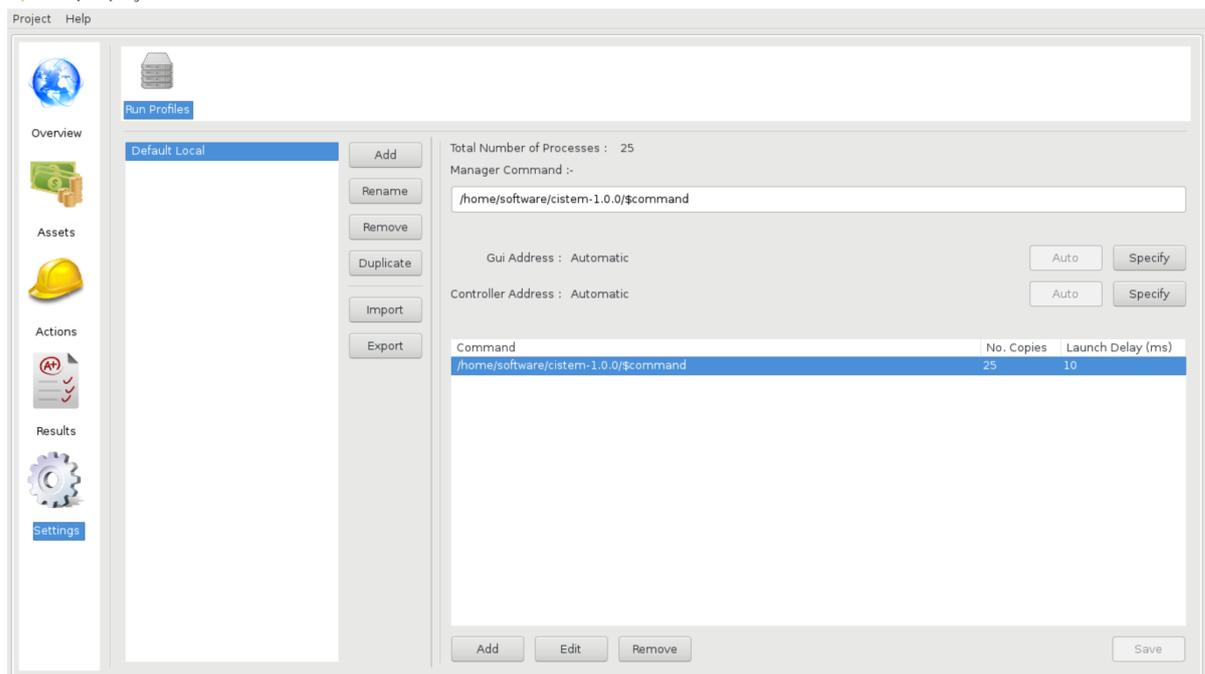
■ 登陆 管理节点 10.15.56.103, 并运行:

```
module load cistem
```

```
cisTEM
```

```
[2020-01-15 16:40:04] ~
[lenovo.LAPTOP-GDLAE9T3] > ssh sunqq@10.15.56.103 -p 10086
Last login: Wed Jan 15 16:14:04 2020 from 10.20.70.161
[sunqq@mgmt01 ~]$ module load cistem
[sunqq@mgmt01 ~]$ cisTEM
```

之后打开已有 project 或者新建 project，点击 左栏的 settings:

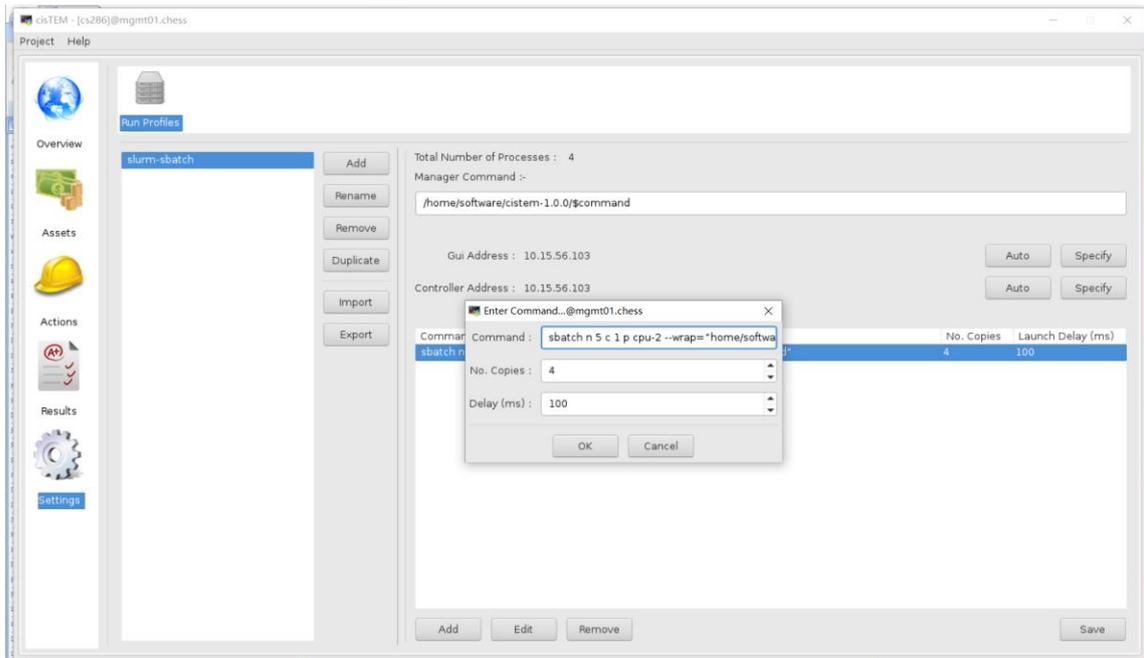


■ 在打开的 Gui 窗口里设置 settings:

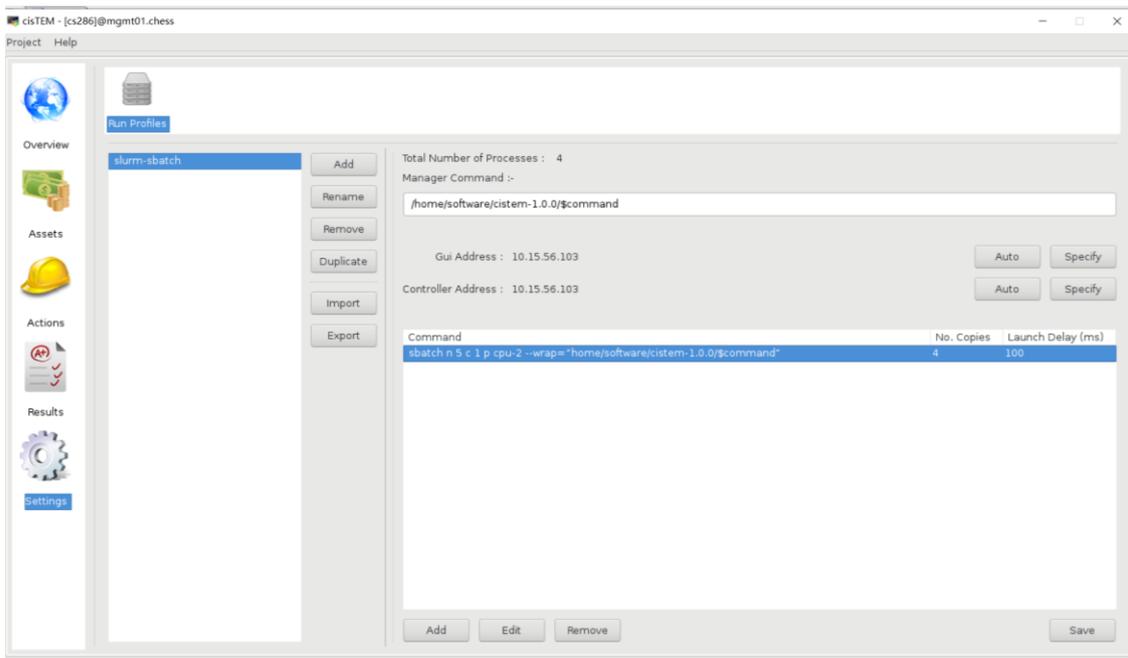
1, 在左边窗口处点击 rename, 将当前 profile 名字改为 slurm-sbatch (也可选用其他名字)

2, 点击右边窗口的 Specify 按钮, 分别将 GUI Address 与 Controller Address 改为登陆节点 IP: 10.15.56.103

3, 双击 Command 下面的蓝条, 可以进入编辑命令状态



编辑完命令，点击 OK，GUI 窗口内容此时如下：



4，此时点击右下角 save 保存。切记一定要保存，尤其首次使用 cisTEM 时，否则就会在管理节点上跑任务。

后续每次调整该提交命令中的参数之后都需点击 save。

- 提交作业命令示例 1:

`sbatch -n 5 -c 1 -p cpu-2 --wrap="home/software/cistem-1.0.0/$command"`

No copies 为 4

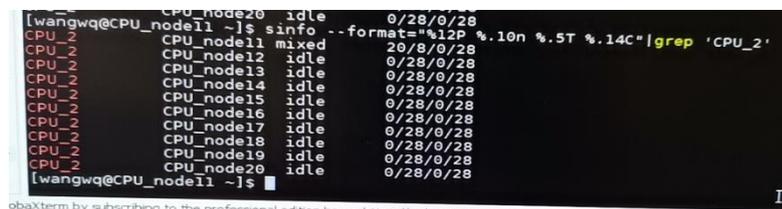
命令含义：5 个任务，每个任务使用 1 个核，在 cpu-2 队列上执行。

并复制 4 个提交命令，也即产生 4 个作业，有 4 个不同的 jobid。

使用如下命令可以查看队列 cpu-2 上的使用核的情况

```
sinfo --format="%12P %.10n %.5T %.14C" |grep 'cpu-2'
```

显示：



```
[wangwq@CPU_node11 ~]$ sinfo --format="%12P %.10n %.5T %.14C" |grep 'cpu-2'
CPU_2 CPU_node11 mixed 20/8/0/28
CPU_2 CPU_node12 idle 0/28/0/28
CPU_2 CPU_node13 idle 0/28/0/28
CPU_2 CPU_node14 idle 0/28/0/28
CPU_2 CPU_node15 idle 0/28/0/28
CPU_2 CPU_node16 idle 0/28/0/28
CPU_2 CPU_node17 idle 0/28/0/28
CPU_2 CPU_node18 idle 0/28/0/28
CPU_2 CPU_node19 idle 0/28/0/28
CPU_2 CPU_node20 idle 0/28/0/28
[wangwq@CPU_node11 ~]$
```

● 提交作业命令 多节点示例 2:

将提交作业的命令改为

```
sbatch -n 5 -c 2 -p cpu-2 --wrap="home/software/cistem-1.0.0/$command"
```

No copies 为 4

命令含义：5 个任务，每个任务使用 2 个核，在队列 cpu-2 上执行。

产生四个 jobid

每个作业使用 10 个核。4 个作业使用 40 个核，因为每个节点 28 个核，所以提交任务后 squeue 显示分配 2 节点。

同样使用命令查看 cpu-2 上的使用核的情况

```
sinfo --format="%12P %.10n %.5T %.14C" |grep 'cpu-2'
```

显示：

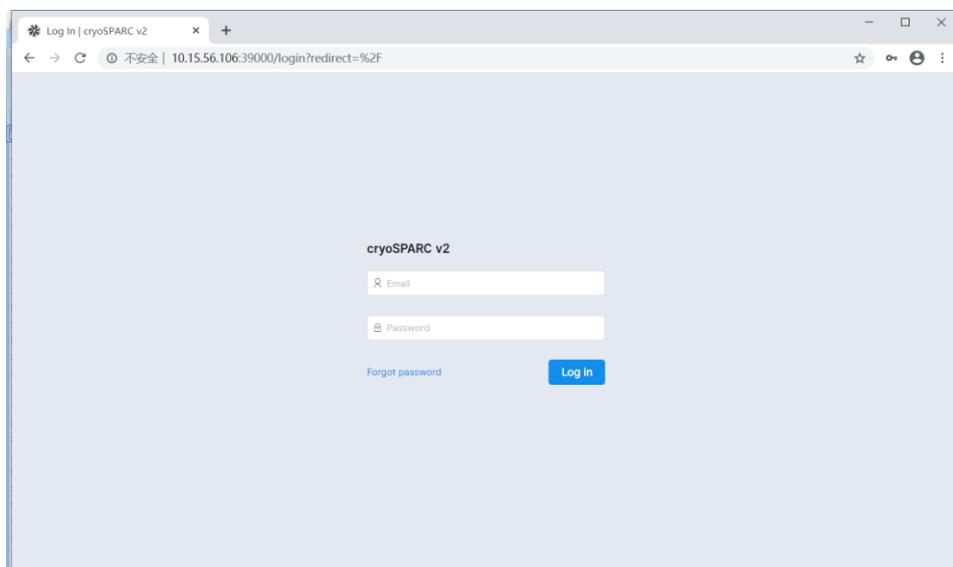
```
[wangwq@CPU_node11 ~]$ sinfo --format="%12P %.10n %.5T %.14C"|grep 'CPU_2'
CPU_2 CPU_node11 mixed 20/8/0/28
CPU_2 CPU_node12 mixed 20/8/0/28
CPU_2 CPU_node13 idle 0/28/0/28
CPU_2 CPU_node14 idle 0/28/0/28
CPU_2 CPU_node15 idle 0/28/0/28
CPU_2 CPU_node16 idle 0/28/0/28
CPU_2 CPU_node17 idle 0/28/0/28
CPU_2 CPU_node18 idle 0/28/0/28
CPU_2 CPU_node19 idle 0/28/0/28
CPU_2 CPU_node20 idle 0/28/0/28
[wangwq@CPU_node11 ~]$
```

六， cryoSPARC 在集群中的使用方法

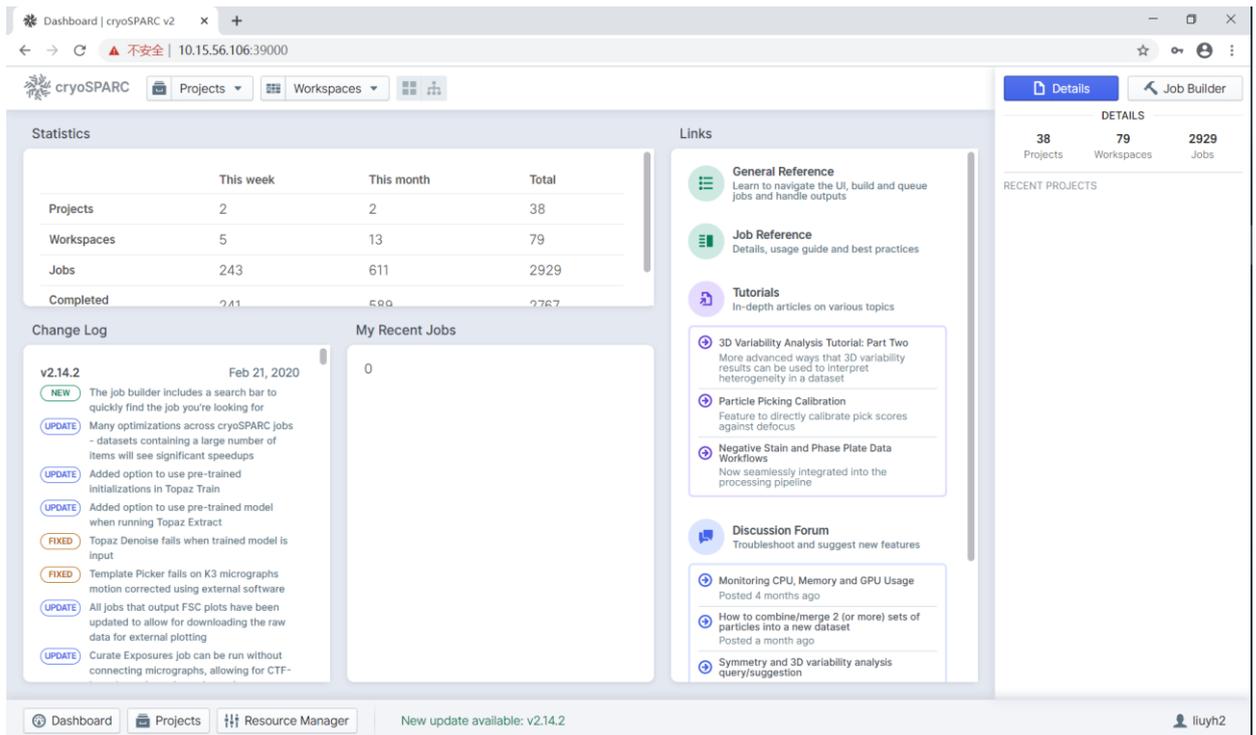
目前集群中有专门的两个节点 `gpu-a-node07.chess` 与 `gpu-a-node08.chess` 分别作为队列 `cryosparc-1` 与 `cryosparc-2` 运行用户的 `cryoSPARC` 作业。

使用 `cryoSPARC` 前，用户需要先向集群管理员申请账号，并向管理员说明处理数据所在位置，由管理员修改文件夹权限，增加 `cryoSPARC` 管理员读写权限。

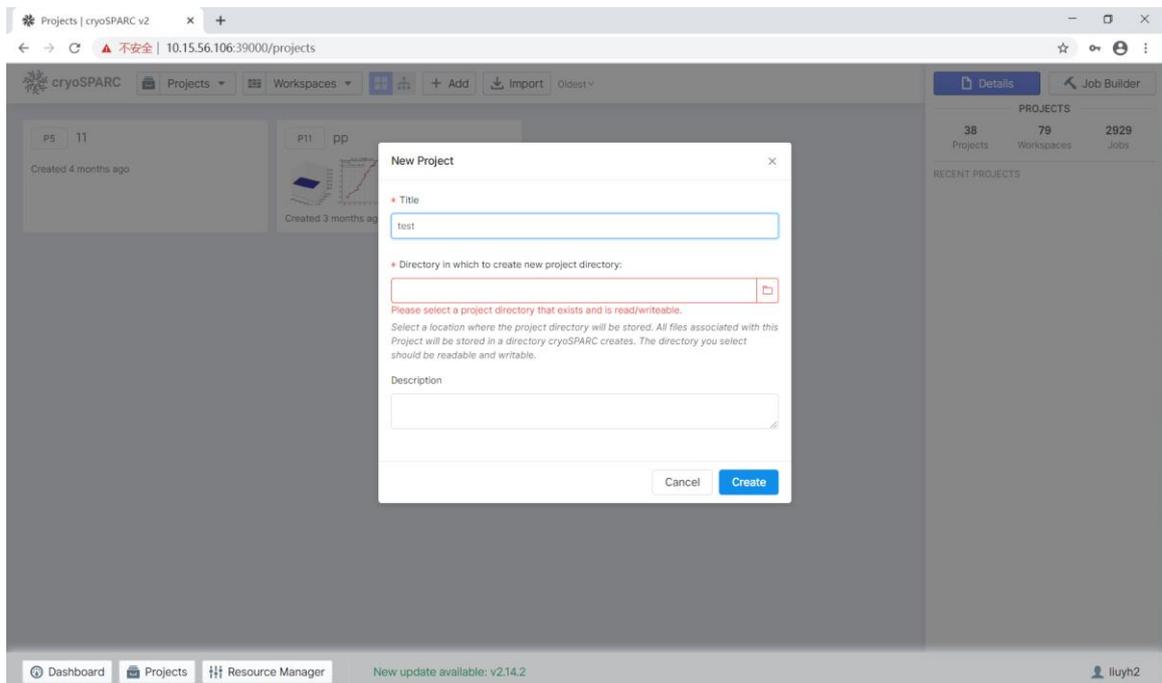
1， 使用浏览器（推荐 google chrome）登录 10.15.56.106: 39000



使用申请到的账号密码登录之后，显示如下：

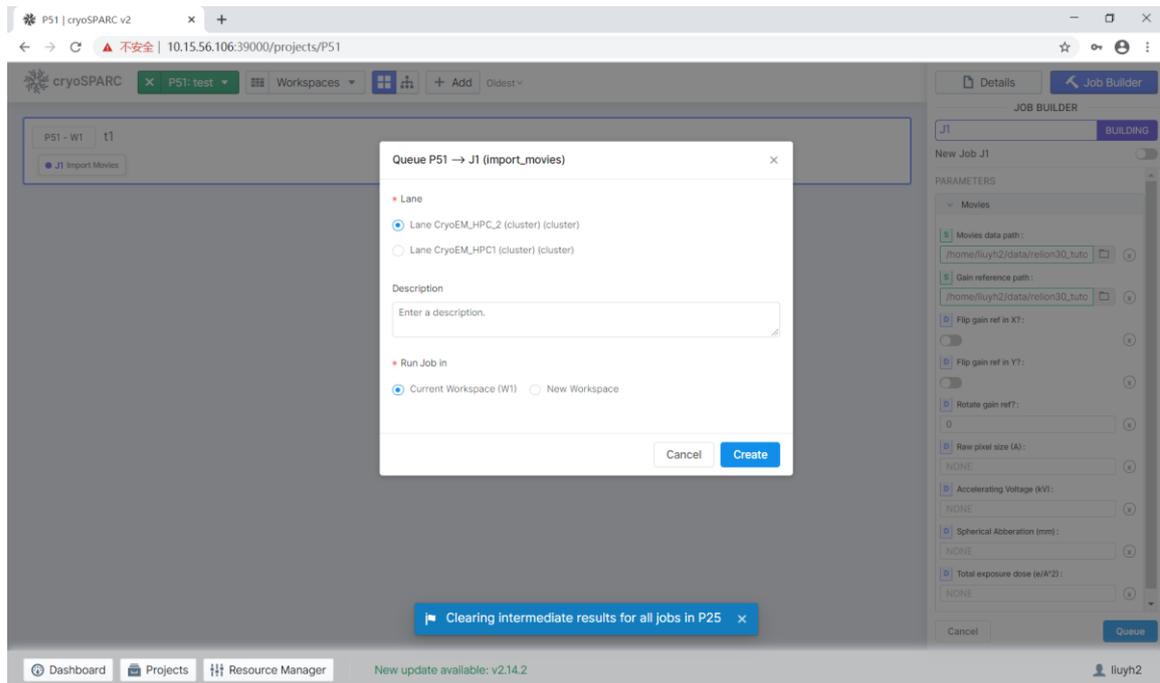


2, 点击 Projects-Add



新建 project。

3 提交作业



目前集群有两个单节点队列作为 **cryoSPARC** 专门队列,提交作业时点击右下角的 **queue** 按钮,出现选择队列的对话框,用户可以根据队列的忙闲程度选择,之后点击 **Create** 按钮。