

1. 申请交互式 job

```
srun -p i64mltga800u -n 1 -c 64 --gres=gpu:8 --pty bash
```

如果想要申请 a40 服务器的话可以换成 a40 的分区。

可以从网页 ui 我的作业->仿真作业, 上查到这个 job 什么时候开始执行

| 作业号 | 作业名 | 集群名 | 状态 | 应用名 | 执行节点 | 执行目录 | 提交时间 | 执行时间 | 结束时间 |
|-------|------|-----------|----|-----|------------|----------------|--------------------|--------------------|------------|
| 28601 | bash | hkuhpc-ai | 等待 | - | - | /hpc2hdd/ho... | 2023-11-21 11:3... | - | - |
| 27388 | bash | hkuhpc-ai | 运行 | - | 64*gpu1-24 | /hpc2hdd/ho... | 2023-11-19 13:2... | 2023-11-20 18:5... | 2023-11-27 |

国庆前后 hpc 进行了调整, 需要加上--gres=gpu:8 才能申请上 gpu, 不加的话只能用 gpu 服务器上的 cpu, 浪费资源

2. Dependency

有时候为了科研稳定又不想浪费学校资源导致其他同学用不上, 可以对多个 job 加上 dependency, 比如想占用一台服务器 2 周, 由于每个 job 是 7 天, 所以可以连续申请 2 个 job 并且加上 dependency,

```
srun -p i64mltga800u -n 1 -c 64 --gres=gpu:8 --pty bash
```

```
srun -p i64mltga800u -n 1 -c 64 --gres=gpu:8 -d afterany:上一个 job 的 id --pty bash
```

hpc 最近限制了每个用户只能使用 8 卡 64cpu, 也限制了每个用户等待 job 的数量, 可以通过 `squeue -u 用户名` 查看每个 job 的状态

| | | | | | | |
|-------|-----------|------|--------|----|------------|----------------|
| 24804 | i64mltga8 | bash | hxu047 | PD | 0:00 | 1 (Dependency) |
| 24805 | i64mltga8 | bash | hxu047 | PD | 0:00 | 1 (Dependency) |
| 14939 | i64mltga8 | bash | hxu047 | R | 4-23:06:24 | 1 gpu1-4 |
| 17812 | i64mltga8 | bash | hxu047 | R | 5-02:01:53 | 1 gpu1-27 |

| | | | | | | |
|-------|-----------|------|--------|----|------|---------------------------|
| 28601 | i64mltga8 | bash | hxu047 | PD | 0:00 | 1 (QOSMaxCpuPerUserLimit) |
|-------|-----------|------|--------|----|------|---------------------------|

比如 (dependency) 代表这个 job 是依赖之前的 job, (qosmaxcpuperuserlimit) 代表这个 job 超出了 job 限制。

3. 其他

hpc ip 是 10.120.18.63:

```
scp testuser21@10.120.18.63:/hpc2hdd/home/testuser21/workspace/a.pth ./a.pth
```

```
scp -r xuhuihui@10.120.16.239:/home/xuhuihui/workspace/a ./a
```

之前的 scp 地址不能用了, 需要换成 10.120.18.63, 如果还不能用的话, 试一下 ping `hpc2login.hpc.hkust-gz.edu.cn`, 看一下解析到的具体 ip

4. Git 版本控制

在 hpc 上 git clone ssh 我们服务器上的项目, 我们服务器上这边代码修改并且 git commit 之后, hpc 这边执行 `git pull origin master/main` 就可以更新

大家合理使用, 避免造成我们内部也用不了的情况