



科学与工程计算中心

Center for Computational Science and Engineering

# 超级计算机 使用 手册

更新时间：2023 年 3月

<b>1. 集群基本情况</b>	<b>3</b>
1.1 系统概况	3
1.2 硬件概况	3
<b>2. 登陆集群</b>	<b>4</b>
2.1. VPN账号	4
2.2. 校外登录（校内不用vpn）	5
2.3. 集群账号	5
2.4. 登陆和ftp	6
2.5. 登陆界面	7
<b>3. 集群编译器及环境</b>	<b>7</b>
3.1. 查看模块	7
3.2. 加载模块	8
3.3. 卸载模块	8
3.4. Module使用手册	8
<b>4. 软件编译安装</b>	<b>8</b>
4.1. 已有软件手册	8
4.2. obspy软件安装	8
<b>5. 使用集群</b>	<b>9</b>
5.1. 集群目录	9
5.2. 集群回收站	9
5.3. 集群队列	9
5.4. 查询存储使用量	11
5.5. 作业范例	11
5.6. 作业调度系统使用	13
5.7. 计费与充值	17
5.8. 查询机时及费用	17
5.9. 问题反馈	18

# 1. 集群基本情况

## 1.1 系统概况

太乙包含815个双路刀片节点，2个大内存节点和4个GPU节点，GPFS 并行文件存储系统，Intel Omni-Path 高速(100 Gbps OPA)计算网络。太乙系统理论计算峰值超过2500万亿次，实测计算性能超过1687万亿次。在2018年11月的全球Top 500中排名第127。

启明2.0包含269个刀片节点，12个大内存节点和7个GPU节点，GPFS 并行文件存储系统，Infiniband高速(100 Gbps)计算网络。

## 1.2 硬件概况

### 1.2.1. 太乙硬件资源：

节点	数量	配置
计算节点：	815	2个Xeon Gold 6148 CPU(2.4GHz/20c), 192 GB内存
大内存节点：	2	8个Xeon Platinum 8160 CPU (2.1GHz/24c), 6 TB内存
GPU节点：	4	2个Xeon Gold 6148 CPU(2.4GHz/20c), 384 GB内存, 2张NVIDIA V100
存储：	5.5 PB	GPFS并行文件系统，实测读写带宽均超过40 GB/s

### 1.2.2. 启明2.0硬件资源

节点	节点数量	GPU数量	配置
登录节点-1	1	0	CPU: 2 * Intel Xeon Gold 4314 (2.4GHz,16核); 内存: 128GB;
登录节点-2	2	0	CPU: 2 * Intel Xeon E5-2690v3 (2.6GHz, 12核); 内存: 128 GB;
计算节点-38	36	0	CPU: 2 * Intel Xeon Gold 6338 (2.0GHz, 32核); 内存: 512 GB;
大内存节点-ot38	1	0	CPU: 2 * Intel Xeon Gold 6338 (2.0GHz, 32核); 内存: 512 GB + 2TB Optane;
大内存节点-2t50c	4	0	CPU: 2 * Intel Xeon Platinum 8350c (2.6GHz, 32核); 内存: 2TB;
计算节点-amd73x	1	0	CPU: 2 * AMD EPYC 7773X (2.2GHz, 64核); 内存: 512 GB;
计算节点-amd63	1	0	CPU: 2 * AMD EPYC 7763 (2.45GHz, 64核); 内存: 512 GB;
大内存节点-v3-6t	7	0	CPU: 8 * Intel Xeon E7-8880v3 (2.3GHz, 18核); 内存: 6 TB;
计算节点-v3-64	225	0	CPU: 2 * Intel Xeon E5-2690v3 (2.6GHz, 12核); 内存: 64 GB;
计算节点-v3-128	6	0	CPU: 2 * Intel Xeon E5-2690v3 (2.6GHz, 12核); 内存: 128 GB;
GPU节点-2a100-40	1	2	CPU: 2 * Intel Xeon Gold 6230 (2.1GHz, 20核); 内存: 192 GB; GPU: 2 * NVIDIA A100 (40GB显存);

GPU节点-4a100-40	1	4	CPU: 2 * Intel Xeon Gold 6258R (2.7GHz, 28核); 内存: 384 GB; GPU: 4* NVIDIA A100 (40GB显存, NV-Bridge);
GPU节点-2a100-80	1	2	CPU: 2 * Intel Xeon Gold 6230 (2.1GHz, 20核); 内存: 192 GB; GPU: 2 * NVIDIA A100 (80GB显存);
GPU节点-4a100-80	1	4	CPU: 2 * AMD EPYC 7763 (2.45GHz, 64核); 内存: 512 GB; GPU: 4 * NVIDIA A100 (80GB显存);
GPU节点-hgxa100	2	16	CPU: 2 * AMD EPYC 7763 (2.45GHz, 64核); 内存: 1TB; GPU: 8 * NVIDIA A100 (SXM 80GB显存);
GPU节点-2v100	1	2	CPU: 2 * Intel Xeon Gold 6230 (2.1GHz, 20核); 内存: 192 GB; GPU: 2 * NVIDIA V100(32GB显存);
存储	1PB		GPFS并行文件系统，实测读写带宽均超过36GB/s

## 2. 登陆集群

### 2.1. VPN账号

在学校外面登陆集群，首先需要VPN账号，登陆VPN账号后，再登陆集群。

- 2.1.1
- 申请VPN账号：发邮件至网络信息中心 [its@sustech.edu.cn](mailto:its@sustech.edu.cn);
- 2.1.2
- 图2.1描述VPN账号登陆过程：



1 <https://vpn.sustech.edu.cn>。

2 客户端中输入VPN 用户名和密码。

3 成功登录后，并点击启动按钮，若显示已成功网络拓展服务，则表明登录VPN 成功




(图：2.1)

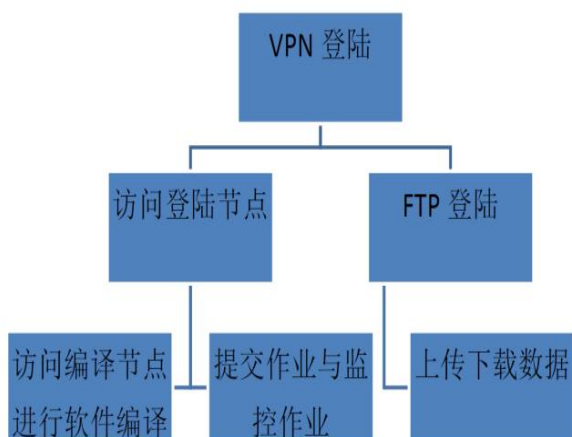
2.1.3      图：2.2红色框显示已连接和IP，表示vpn登陆成功



(图：2.2)

## 2.2. 校外登录（校内不用vpn）

图2.3显示校外登陆集群流程



(图:2.3)

## 2.3. 集群账号

### 2.3.1. 集群账号申请

申请集群账号发邮件至 (hpc@sustech.edu.cn);



(图：2.3.1-1)



(图：2.3.1-2)

- 注：1、图2.3.1-1为申请账号样例；
- 2、申请表在<http://hpc.sustech.edu.cn/>首页下载；
- 3、申请邮件需抄送给PI（课题组长）；
- 4、账号开通后，有回复邮件，图2.3.1-2为样例；

## 2.4. 登陆和ftp

太乙登陆和传输数据，采用的动态负载技术，使用端口号为22，IP：172.18.6.175

启明登陆和传输数据，使用端口号为18188，IP：172.18.6.10

## 2.5. 登陆界面



```

debug: 20 min less than 80c Open
ser: 168h less than 100c Open
short: 144h More Than 40c Open
medium: 72h More Than 200c Open
large: 36h More Than 600c Open
smp: 168h Unlimited Open
gpu: 168h 20c/gpu Open

User--guide: http://hpc.sustech.edu.cn/process/user_guide.html
login ip: 172.18.6.175

CCSE WeChat public account : gh_7cd85ddedaf7
Support Team QQ: 161195492
Support Team Email: hpc@sustech.edu.cn
Support Team Phone: 0755-88015831,17318031826, 15986647786

Attention:
1. Please do not run jobs that keep generating numerous small files for a long time before contacting the support team for a proper solution, because it will cause a high pressure to the storage system and slow down everyone.
2. Please do not run large memory jobs on the login node, because it will slow down everyone.
3. Please do not use MPICH in job, because it may cause a high pressure to the storage system and slow down everyone.
4. In order to ensure the security of data, please backup your data to your local disk regularly.

Warning: violation of the above rules may result in job termination and account blocking

BASIC COMMAND:
bsub < jobscript.bsub #submit a job, do not forget the '<'
bjobs #check the status of your jobs
bkill $JOBID1 $JOBID2 $JOBID3 #stop your job ID1 ID2 ID3 ...
bhosts hq_short #check the status of queue short, you can check debug ser short medium ...

We strongly recommend the users to use the Issue Tracking System, when you have any problems in using the supercomputers of CCSE.
1. Address: http://172.18.6.195:18090/.
2. User Guide: https://hpc.sustech.edu.cn/ref/HPMS_UserGuide.pdf.
3. Customer Service Phone Number: 0755-88015833.

```

- QUEUE' S RUNLIMIT NOTICE是队列时间限制和最小核数限制;
- Attention为上机注意事项: 1. 未经允许, 不能在集群上运行持续创建大量小文件的作业(文件数量级别在千万以上), 因为这样的作业会给共享存储带来巨大的IO压力从而影响正常的作业; 2. 在登陆节点上, 禁止运行消耗CPU或者内存资源的作业; 3. 禁止在作业中使用MPICH; 4. 为了数据安全, 请自行将重要数据备份到本地磁盘。
- 登陆界面的内容, 会根据实际情况进行调整。

## 3. 集群编译器及环境

由于用户可能需要使用不同的软件环境, 配置不同的环境变量, 因而在“太乙”和“启明2.0”上安装了“module软件”来进行管理, 用户方便环境变量的设置从而提高移植软件效率。因为“太乙”的cpu为intel Xeon Gold, 建议使用Intel编译器。

### 3.1. 查看模块

module avail: 查看可用的模块列表, 如图: 3-1

```

[root@login01 Li-bulk]# module avail

----- /share/base/modulefiles/compilers -----
gcc/8.2.0(default) intel/2015.6 intel/2018.3 intel/2019.1
go/1.11.1 intel/2017.8 intel/2018.4 intel/2019.4bk

----- /share/base/modulefiles/parallel -----
dapl/2.1.7 itac/2018.3.022 mpi/intel/2019.1 mpi/openmpi/3.1.2_gcc mpi/platform/9.1.4.r3
dapl/2.1.9 knem/1.1.3 mpi/intel/2019.4bk mpi/openmpi/3.1.2_intel mpiP/4.3.1/impi
dapl/2.1.9-fca mpi/intel/2017.8 mpi/mvapich2/2.3b mpi/platform/2.3.0
ga/5-4b/impi mpi/intel/2018.3 mpi/mvapich2/2.3_gcc mpi/platform/8.1.1
ipsxe/2018.3.222 mpi/intel/2018.4 mpi/openmpi/2.1.2_gcc mpi/platform/9.1.2

----- /share/base/modulefiles/bio -----
amber/16

----- /share/base/modulefiles/chem -----
vasp/5.4.1 vasp/5.4.4 vasp/5.4.4-vtst

----- /share/base/modulefiles/cae -----
ansys/192 CFX/192 fluent/192 matlab/2017b OpenFoam/3.0

----- /share/base/modulefiles/graphics -----
gnuplot/5.2.6 jasper/1.900.1/gnu jpeg/9a ncl/6.5.0-gnu485 png/1.6.17

```

(图: 3-1)

### 3.2. 加载模块

`module load [modulesfile]`: 加载使用的 modulefiles, 如图:3-2

```
[ccse-xiezy@login01 ~]$ module load intel/2018.4  
[ccse-xiezy@login01 ~]$ which icc  
/share/intel/2018u4/compilers_and_libraries_2018.5.274/linux/bin/intel64/icc  
[ccse-xiezy@login01 ~]$
```

(图: 3-2 )

### 3.3. 卸载模块

`module unload [modulesfile]`: 移除使用 module 加载的软件环境, 如图:3-3

```
[ccse-xiezy@login01 ~]$ module unload intel/2018.4  
[ccse-xiezy@login01 ~]$ which icc  
~/intel/compilers_and_libraries_2019.4.243/linux/bin/intel64/icc  
[ccse-xiezy@login01 ~]$
```

(图: 3-3 )

### 3.4. Module使用手册

为了方便用户, 计算中心编写了《“太乙”Module使用手册v1.0》, 手册下载链接:  
<https://hpc.sustech.edu.cn/business/softtest.html>

## 4. 软件编译安装

### 4.1. 已有软件手册

为了省去繁琐的安装调试, 方便用户直接使用软件; 计算中心会安装应用软件并形成手册。截止到目前为止, 已经完成的有:

《“太乙”cp2k使用手册v1.0》、《“太乙”FEniCS使用手册v1.0》、《“太乙”gpaw使用手册v1.0》、《“太乙”HiQ使用手册v1.0》、《“太乙”OPENMX使用手册v1.0》、《“太乙”QE使用手册v1.0》、《“太乙”wrf安装使用手册v1.0》。软件手册下载链接:  
<https://hpc.sustech.edu.cn/business/softtest.html>。

### 4.2. obspy软件安装

目前太乙以login01为例:

- 第1步, `ssh login01` ;
- 第2步, `module load python/anaconda3/5.2.0` ;
- 第3步, `conda config --add channels conda-forge` ;
- 第4步, `conda create -n mypy python=3.7` ;
- 第5步, `conda activate mypy` ;
- 第6步, `conda install obspy` ;



## 5. 使用集群

### 5.1. 集群目录

太乙三个目录可以使用(work, data, scratch), 限额是针对课题组而言, 数据量小(例如: 不到10G), 建议直接使用/work目录, 以下内容可以忽略。如果数据量大, 请仔细阅读以下内容:

- /work目录限额为1TB, 存放核心数据, 例如: 源代码, 安装部署软件;
- /data目录限额为8TB(可用容量4TB), 存放运算数据, 支持存储容量扩容, 扩容费用见收费政策;
- /scratch目录不限额, 用于中间数据量大的作业。例如: Gaussian运算的过程中, 产生10T的大文件, 而/work和/data空间限额没有10T, 作业运算不了。这时可以使用/scratch目录;
- 为了保证/scratch有充足的存储空间, 设置自动清理10天前目录的数据, /scratch每天会自动产生一个日期目录, 用户可以将数据写到这个目录, 但是数据最多保存10天, 例如/scratch/2020-01-01这个目录在2020年1月1日自动创建, 2020年1月11日会自动迁移目录/scratch/2020-01-01至回收站, 并且2020年1月11日前, 用户能收到数据删除的邮件提醒通知。如果有数据需要保存, 请在十天内及时把数据迁移到/data或者/work。如作业持续运行时间超过10天, 请不要使用/scratch目录, 使用/data或/work目录;
- /data目录采用双副本的存储方式(存储系统为了保证数据安全采用的一种存储方式), 例如: 1T的数据, 其实占有2T的空间; 扩容收费还是按照1T来收, 详细情况请阅收费政策;
- 基于存储空间容量的限制和对数据安全的考虑, 请用户及时把重要数据或敏感数据保存到自己的计算机中, 并及时清理自己的存储空间。

启明没有data和scratch, 只有work目录可以使用。

### 5.2. 集群回收站

太乙rm命令删除数据, 如果发生删除错误, 或者scratch目录数据被自动清理, 数据能否找回? 为了应对这个问题, 系统设置了回收站, 回收站的位置:

/scratch/recyclebin/。当天和前天目录, 分别回收当天和前天删除的数据, 回收站里面的数据保存2天, 2天后, 数据被彻底被清理。

因为回收站是放在目录为/scratch, 该目录没设限额, 所以不占用用户的存储空间。如果不想使用回收站, 想直接删除数据, 那么使用绝对路径命令: /bin/rm, 这种方式删除数据, 是不能找回。

由于整体存储空间的限制, 启明没有设置回收站。

### 5.3. 集群队列

“太乙”集群目前共开放以下七个队列:

队列	队列时长限制	应用场景	单作业核心数限制
large	36小时	大规模	600核以上（含600）
medium	72小时	中等规模	200核以上（含200）
short	144小时	小规模	40核以上（含40）
ser	168小时	串行	100核以下
smp	168小时	大内存	无限制
debug	20分钟	调试	80核以下
gpu	168小时	GPU	20核/gpu
Spec	无限制	整机找资源，申请使用	无限制

注：以上数据可能会根据实际情况，进行调整更改，请以登录节点登录界面信息为准。

“启明2.0”集群目前共开放以下队列：

队列名称	每节点CPU核数	每节点GPU数量	CPU和GPU配置	内存(GB)	节点名	最长时间(测试期)	最大作业数
38	64	-	Intel 6338/2.0GHz * 2	512	b04u[22-39][a-b]	48 小时	5 个 (测试期)
ot38	64	-	Intel 6338/2.0GHz * 2	512GB + 2TB Op tane	b04u14s	72 小时	
2t50c	64	-	Intel 8350C/2.6GHz * 2	2048	b05u[01-04]s	48 小时	
73x	128	-	AMD EPYC 7773X/2.2GHz * 2	512	b05u23g	48 小时	
63	128	-	AMD EPYC 7763/2.45GHz * 2	512	b05u29a	48 小时	
v3-6t	144	-	Intel E7-8880v3/2.3GHz * 8	6144	a0[1-2, 7-8]u21s、 a0[3, 4]u24s、a05u05s	72 小时	
v3-64	24	-	Intel Xeon E5-2690v3/2.6GHz * 2	64	a0[1, 2, 7]u[01-20][a-b]、 a0[3, 6]u[03-23][a-b]、 a08u[06-20][a-b]	72 小时	
v3-128	24	-	Intel Xeon E5-2690v3/2.6GHz * 2	128	a[01, 02, 07, 08]u30g、 a[03, 06]u33g	72 小时	
2a100-40	40	2	Intel 6230/2.1GHz * 2 A100 40GB * 2	192	b05u37g	72 小时	
4a100-40	56	4	Intel 6258R * 2 A100 40GB * 4, NV-Bridge	384	b05u31g	72 小时	
2a100-80	40	2	Intel 6230 * 2 A100 80GB * 2	192	b05u14g	72 小时	
4a100-80	128	4	AMD EPYC 7763 * 2 A100 80GB * 4	512	b04u15g	72 小时	
hgx	128	8	AMD EPYC 7763 * 2 A100 80GB * 8	1024	b05u17g、b05u08g	72 小时	
2v100	40	2	Intel 6230 * 2 V100 32GB * 2	192	b05u35g	72 小时	

spec		整机找资源，申请使用	无限制	无限制
------	--	------------	-----	-----

注：以上数据可能会根据实际情况，进行调整更改，请以登录节点登录界面信息为准。

## 5.4. 查询存储使用量

太乙提供work 1T和data 4T的免费空间，启明2.0提供work 1T的免费存储空间。使用mmlsquota,可以查询配额限制和使用情况，若是个人，使用参数-u。若是课题组，使用参数-g，如下图：

```
[para-test@login05 ~]$ sudo mmlsquota -g ese-zhengcm --block-size auto
```

Disk quotas for group ese-zhengcm (gid 1216):

Block Limits							File Limits					
Filesystem	type	blocks	quota	limit	in_doubt	grace	files	quota	limit	in_doubt	grace	Remarks
data	GRP	7.502T	8T	8.25T	1.251G	none	539123	0	0	218	none	
scratch	root	GRP	1.833T	100T	100.1T	13.62G	none	1115727	0	0	520	none
work	GRP	1.093T	1T	1.098T	1.776G	expired	2026266	0	0	601	none	

可以使用命令：sudo lsquota -g mae-xxl -f work data scratch，查看组内总存储使用

```
[mae-hulq@login01 ~]$ sudo lsquota -g mae-wanglp -f work data scratch
```

GroupName	UserName	work	data	scratch
mae-wanglp	mae-wanglp	477.9GB	16.9TB	843.8GB

sudo lsquota -g mae-xxl -f work data scratch -l，查看组内各成员存储使用情况，并有针对性地清理不必要的空间或者申请扩容，如下图：

```
[mae-hulq@login01 ~]$ sudo lsquota -g mae-wanglp -f work data scratch -l
```

GroupName	UserName	work	data	scratch
mae-wanglp	mae-anandhts	47.6GB	126.2GB	0.0KB
	mae-chent	40.9GB	276.6GB	10.6MB
	mae-chenxs	2.5GB	0.0KB	0.0KB
	mae-dongzq	3.5GB	1.1TB	0.0KB
	mae-fanzzhp	1.7GB	0.0KB	0.0KB
	mae-gaoh	49.6GB	0.0KB	0.0KB
	mae-guol	15.8GB	0.0KB	0.0KB
	mae-hutq	51.1GB	3.4TB	0.0KB
	mae-jix	8.6GB	23.4GB	0.0KB

## 5.5. 作业范例

5.5.1、太乙范例1-自编mpi代码，更多范例，请参考科学与工程计算中心官网：

<http://hpc.sustech.edu.cn/userguide.html>

```
#!/bin/bash
#BSUB -J test
#BSUB -q short
#BSUB -n 320
#BSUB -e %J.err
#BSUB -o %J.out
#BSUB -R "span[ptile=40]"
```



```
#BSUB -R "select[hname!='r03n43']"
#BSUB -R "select[hname!='r03n55']"
#BSUB -R "select[hname!='r03n64']"
module load fftw/2.1.5
module load intel/2018.4
module load mpi/intel/2018.4
cd $LS_SUBCWD
echo "processes will start at:"
date

mpirun -machinefile $LSB_DJOB_HOSTFILE -np 320 ./main > $LSB_JOBID.log 2>&1
echo "processes end at:"
date
```

- 编译 mpicc 必须与 mpirun 一致,这里是 2018.4
- 不推荐使用自己编译 mpi 软件和 fftw 库
- main 必须是可执行文件(chown a+x main)
- 经过测试,建议使用系统自带的 2018.4 版本 intel mpi

### 5.5.2、太乙范例2-单节点openMP

编译: gcc -fopenmp omp\_exam.c -o omp\_exam\_icc

```
#!/bin/bash
#BSUB -q short
#BSUB -n 40                                ##number of total cores
#BSUB -e %J.err
#BSUB -o %J.out
#BSUB -R "span[ptile=40]"
hostfile=`echo $LSB_DJOB_HOSTFILE`
NP=`cat $hostfile | wc -l`
cd $LS_SUBCWD
export OMP_NUM_THREADS=80                  ##number of total task
module load intel/2018.4

./omp_exam_icc > log 2>&1                  ##download:/share/user_guide/exam/omp_exam_icc
```

### 5.5.3、太乙范例3-跨节点openMP+mpi

编译: mpicc mpi0openmp.c -o mpi0openmp -fopenmp

```
#!/bin/sh
#BSUB -J 2nodes2cores                      ##job name
#BSUB -q debug                             ##queue name
#BSUB -n 4                                ##number of total cores
#BSUB -R "span[ptile=2]"                  ##2 cores per node
#BSUB -W 00:10                             ##walltime in hh:mm
#BSUB -e %J.err.log                       ##error log
#BSUB -o %J.output.log                   ##output log
module purge
module load intel/2018.4 mpi/intel/2018.4
```

```
cat $LSB_DJOB_HOSTFILE|sort|uniq >hostlist
NP=2
export OMP_NUM_THREADS=2
mpiexec.hydra -machinefile hostlist -np $NP mpiOpenmp ##download:/share/user_guide/exam/mpiOpenmp
```

- 注意：基于计算资源申请量最小化原则，此处我们申请 2 个节点，每个节点 2 个进程，export OMP\_NUM\_THREADS=2 启用了 2 个 openMP 线程。这里所取例子是头节点启用 2 个线程。

#### 5.5.4、启明2.0范例4-cpu

```
#!/bin/bash
#BSUB -q v3-64                                #作业运行队列
#BSUB -n 24                                    #作业运行核数
#BSUB -e %J.err                                #作业错误信息文件
#BSUB -o %J.out                                #作业输出文件
#BSUB -R "span[ptile=24]"                      #作业 24 核在一个节点运行

date
hostfile=`echo $LSB_DJOB_HOSTFILE`
NP=`cat $hostfile | wc -l`
module load compiler/2022.1.0 mpi/2021.6.0
mpirun -machinefile $hostfile -np $NP ./mpi_fortran > $LSB_JOBID.log 2>&1
date
```

#### 5.5.5、启明2.0范例5-GPU

```
#!/bin/bash
#BSUB -q 4a100-80                                ##队列名
#BSUB -n 4                                        ##申请的 CPU 总核数
#BSUB -e %J.err
#BSUB -o %J.out
#BSUB -R "span[ptile=4]"                        ##每个 host 上的 CPU 核数
#BSUB -gpu "num=4/host"
##"num=4/host"参数的含义是每个 host 申请 4 张 GPU 卡，那么此脚本总共申请了 1 个 host,每个 host 上 4 个 CPU 核，每个 host 上分配 4 张 GPU 卡；也可以使用"num=1/task"参数,其含义是每个 CPU 核申请 1 张 GPU 卡，那么此时脚本总共申请了 1 个 host，每个 host 上 4 个 CPU 核，每个 CPU 核分配 1 张 GPU 卡，总共 4 张 GPU 卡。

date
hostfile=`echo $LSB_DJOB_HOSTFILE`
NP=`cat $hostfile | wc -l`
module load cuda/11.8
#nvcc gputest.cu -o gputest
mpirun -machinefile $hostfile -np $NP./gputest > $LSB_JOBID.log 2>&1
date
```

## 5.6. 作业调度系统使用

作业调度平台是LSF 10.1。用户需要熟悉一些基本的Linux 命令行操作，特别是文件目录操作，并能熟练使用一种编辑器（vi 或emacs 等）。本文档介绍LSF常用命令，更



加详细的内容请参考LSF文档:

[http://hpc.sustech.edu.cn/ref/lsf\\_users\\_guide\\_v10.1.pdf](http://hpc.sustech.edu.cn/ref/lsf_users_guide_v10.1.pdf)

## 5.6.1. 查询节点信息

```
[ccse-xiezy@login01 ~]$ bhosts hg_short | grep ok
```

```
r02n03      ok      -      40      0      0      0      0      0
r05n05      ok      -      40      20     20      0      0      0
```

- 注意: 查询太乙 short 队列对应节点信息, 如果查询节点情况信息, 使用 `bhosts s001 s001`

```
[ccse-xiezy@b04u17l 2022.1.0]$ lsload -gpuload
```

HOST_NAME	gpu_id	gpu_model	gpu_mode	gpu_temp	gpu_ecc	gpu_ut	gpu_mut	gpu_mtotal	gpu_mused	gpu_pstate	gpu_status	gpu_error
b05u37g	0	NVIDIA100_	0.0	25C	0.0	0%	0%	40G	575M	0	ok	-
	1	NVIDIA100_	0.0	25C	0.0	0%	0%	40G	575M	0	ok	-
b05u23g	0	TeslaV100_P	0.0	26C	0.0	2%	0%	32G	257M	0	ok	-
b05u35g	0	TeslaV100_P	0.0	28C	0.0	0%	0%	32G	257M	0	ok	-
	1	TeslaV100_P	0.0	28C	0.0	1%	0%	32G	257M	0	ok	-
b05u14g	0	NVIDIA1008	0.0	30C	0.0	0%	0%	80G	819M	0	ok	-
	1	NVIDIA1008	0.0	32C	0.0	23%	0%	80G	819M	0	ok	-
b05u31g	0	NVIDIA100_	3.0	59C	0.0	99%	62%	40G	15.9G	0	ok	-
	1	NVIDIA100_	3.0	62C	0.0	99%	61%	40G	15.9G	0	ok	-
	2	NVIDIA100_	3.0	58C	0.0	99%	61%	40G	15.9G	0	ok	-
	3	NVIDIA100_	3.0	56C	0.0	99%	61%	40G	17.6G	0	ok	-
b04u15g	0	NVIDIA1008	3.0	46C	0.0	97%	43%	80G	15.1G	0	ok	-
	1	NVIDIA1008	3.0	45C	0.0	98%	43%	80G	15.2G	0	ok	-
	2	NVIDIA1008	3.0	45C	0.0	97%	44%	80G	15.2G	0	ok	-
	3	NVIDIA1008	3.0	46C	0.0	97%	44%	80G	16.3G	0	ok	-
b05u08g	0	NVIDIA100_	3.0	39C	0.0	99%	59%	80G	20.4G	0	ok	-
	1	NVIDIA100_	3.0	46C	0.0	99%	59%	80G	20.7G	0	ok	-
	2	NVIDIA100_	3.0	46C	0.0	100%	59%	80G	20.7G	0	ok	-
	3	NVIDIA100_	3.0	41C	0.0	99%	60%	80G	20.7G	0	ok	-
	4	NVIDIA100_	3.0	41C	0.0	99%	60%	80G	20.7G	0	ok	-
	5	NVIDIA100_	3.0	46C	0.0	100%	60%	80G	20.6G	0	ok	-
	6	NVIDIA100_	3.0	47C	0.0	99%	60%	80G	20.7G	0	ok	-
	7	NVIDIA100_	3.0	42C	0.0	100%	60%	80G	20.7G	0	ok	-
b05u17g	0	NVIDIA100_	3.0	38C	0.0	98%	52%	80G	21G	0	ok	-
	1	NVIDIA100_	3.0	43C	0.0	98%	51%	80G	21.3G	0	ok	-
	2	NVIDIA100_	3.0	43C	0.0	98%	52%	80G	21.3G	0	ok	-
	3	NVIDIA100_	3.0	41C	0.0	96%	52%	80G	21.3G	0	ok	-
	4	NVIDIA100_	3.0	38C	0.0	98%	52%	80G	21.3G	0	ok	-
	5	NVIDIA100_	3.0	44C	0.0	95%	52%	80G	21.3G	0	ok	-
	6	NVIDIA100_	3.0	44C	0.0	98%	52%	80G	21.3G	0	ok	-
	7	NVIDIA100_	3.0	40C	0.0	95%	52%	80G	21.3G	0	ok	-

- 如上截图为启明 2.0 查询 gpu 节点资源使用情况; 如要查询 gpu 节点上作业信息, 请参考下图

```
[ccse-xiezy@b04u17l ~]$ bhosts -gpu -w
```

HOST_NAME	GPU_ID	MODEL	MUSED	MRSV	NJOBS	RUN	SUSP	RSV
b04u15g	0	NVIDIA10080GBPCIe	76.7G	0M	1	1	0	0
	1	NVIDIA10080GBPCIe	76.7G	0M	1	1	0	0
	2	NVIDIA10080GBPCIe	819M	0M	0	0	0	0
	3	NVIDIA10080GBPCIe	76.7G	0M	1	1	0	0
b05u08g	0	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	1	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	2	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	3	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	4	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	5	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	6	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	7	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
b05u14g	0	NVIDIA10080GBPCIe	31G	0M	1	1	0	0
	1	NVIDIA10080GBPCIe	31G	0M	1	1	0	0
b05u17g	0	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	1	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	2	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	3	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	4	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	5	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	6	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
	7	NVIDIA100_SXM4_80GB	819M	0M	0	0	0	0
b05u23g	0	TeslaV100_PCIe_32GB	257M	0M	0	0	0	0
b05u31g	0	NVIDIA100_PCIe_40GB	6.2G	0M	1	1	0	0
	1	NVIDIA100_PCIe_40GB	17.4G	0M	1	1	0	0
	2	NVIDIA100_PCIe_40GB	6.2G	0M	1	1	0	0
	3	NVIDIA100_PCIe_40GB	575M	0M	0	0	0	0
b05u35g	0	TeslaV100_PCIe_32GB	257M	0M	0	0	0	0
	1	TeslaV100_PCIe_32GB	257M	0M	0	0	0	0
b05u37g	0	NVIDIA100_PCIe_40GB	6.7G	0M	1	1	0	0
	1	NVIDIA100_PCIe_40GB	6.7G	0M	1	1	0	0

### 5.6.2. 查询队列信息

bqueues -l 队列名      #例如: bqueues -l large

部分信息:

RUNLIMIT                      #作业时间限制

10080.0 min

TASKLIMIT                    #作业核数限制

600 1000 20000

MEMLIMIT                    #作业单节点内存限制

170 G

USERS: all                    #所有用户可使用该队列

HOSTS: hg\_large              #队列节点组

RERUNNABLE : yes            #作业在该队列可重排

### 5.6.3. 提交作业

bsub<test.lsf

注意: 需要有“<”符号

提交 LSF 任务, 成功后会给出此任务的 JOBID。bjobs查看自己的所有运行任务情况; 输入bjobs后, 会列出当前用户正在运行的所有作业, 最左边一列数字是每个作业的 JOBID, 一些其他命令使用的时候需要先调用 bjobs查看JOBID。

**bjobs -l** 查看所有运行任务的详细情况  
**bjobs -l JOBID** 查看 JOBID 这个任务的详细情况  
**bpeek -f JOBID** 跟踪查看某任务屏幕输出  
**bkill JOBID** 终止某任务运行  
**bkill JOBID1 JOBID2 JOBID3** 终止多个任务运行  
**bqueues** 查看所有任务队列的状态  
**bstop JOBID** 临时挂起某个计算作业, 为其它计算腾出资源  
**bresume JOBID** 恢复由 bstop 挂起的作业  
**lshosts** 查看节点的信息  
**bhosts** 查看节点的作业使用信息  
**lsload** 查看节点的即时负载信息

作业结束后, 会在计算目录下生成output. JOBID, 此文件为程序输出内容。

### 5.6.4. 任务监控

使用bjobs可以监控到以下参数:

RUN-作业已经分派到节点机上, 正在运行;

DONE-作业已经正常结束;

EXIT-作业已经结束, 但可能异常退出或者被终止;

UNKWN-作业状态不确定;

作业提交完后的5分钟, 建议查看作业是否正常(A和B查看节点负载,C查看输出文件):

A 在login01上: jobToLoad 150614, 能查到这个作业所有节点负载:

```

40*107/H33
login01 ~]$ jobToLoad 150614
HOST_NAME      status  rl5s  rlm  rl5m  ut    pg  ls   it   tmp  swp  mem
r07n47         ok     40.1  41.2  40.6  100%  0.0  0  46944 213G  3.3G 157G
r05n03         ok     40.1  40.7  40.6  100%  0.0  0  46944 212G  3.4G 157G
r07n27         ok     40.1  40.2  40.6  100%  0.0  0  28048 212G   3G 157G
r05n07         ok     40.2  40.2  40.6  100%  0.0  0  46944 213G  3.2G 157G
r07n23         ok     40.2  40.5  40.5  100%  0.0  0  46944 212G  2.9G 157G
r07n21         ok     40.3  41.1  40.7  100%  0.0  0  46944 212G  3.1G 157G
r07n51         ok     40.3  40.4  40.6  100%  0.0  0  46944 212G  3.5G 157G

```

B 查看当前用户所有作业负载: jobToLoad -a

```

login01 ~]$ jobToLoad -a
HOST_NAME      status  rl5s  rlm  rl5m  ut    pg  ls   it   tmp  swp  mem
r09n04         ok     40.0  40.7  40.6  100%  0.0  0  46944 212G  2.8G 157G
r09n44         ok     40.0  40.2  40.5  100%  0.0  0  46912 212G  3.4G 157G
r08n31         ok     40.1  41.2  40.7  100%  0.0  0  46944 212G  2.7G 157G
r11n11         ok     40.1  40.7  40.6  100%  0.0  0  41344 212G  3.3G 157G
r08n37         ok     40.2  40.7  40.6  100%  0.0  0  46944 212G  2.3G 157G
r11n54         ok     40.2  40.5  40.6  100%  0.0  0  46912 212G   3G 157G
r09n50         ok     40.3  40.9  40.9  100%  0.0  0  46912 212G  2.6G 157G
r08n58         ok     40.3  40.5  40.6  100%  0.0  0  15656 212G  3.5G 157G
r11n17         ok     40.4  40.7  40.5  100%  0.0  0  46912 212G  2.9G 157G
r09n01         ok     40.4  40.3  40.6  100%  0.0  0  46944 212G  3.1G 157G
r08n56         ok     41.0  40.7  40.8  100%  0.0  0  46944 211G  3.5G 157G
r08n60         ok     42.3  40.2  40.6  100%  0.0  0  46944 212G   3G 157G
HOST_NAME      status  rl5s  rlm  rl5m  ut    pg  ls   it   tmp  swp  mem
r09n12         ok     40.0  40.0  40.2  100%  0.0  0  46912 212G  2.9G 157G
r09n55         ok     40.0  40.3  40.3  100%  0.0  0  46912 212G   3G 157G
r10n53         ok     40.0  40.1  40.3  100%  0.0  0  46912 212G   3G 157G
r11n06         ok     40.0  40.4  40.3  100%  0.0  0  46080 212G  3.3G 157G
r09n17         ok     40.3  40.2  40.2  100%  0.0  0  46944 213G  3.2G 157G
r07n12         ok     40.5  40.5  40.2  100%  0.0  0  46944 212G  3.1G 157G

```

C 查看输出文件:

```

#~~~~~
#lattice_vector -0.0000000000000000 5.4630289765190732 5.4630289765190732
#lattice_vector 5.4630289765190732 -0.0000000000000000 5.4630289765190732
#lattice_vector 5.4630289765190732 5.4630289765190732 -0.0000000000000000
# constrain_relaxation full x y z ...
#
#atom_frac 0.1250000000000000 0.6250000000000000 0.6250000000000000 Ag
#atom_frac 0.3773919634347005 0.3773919634347005 0.8678251096958962 S
# constrain_relaxation full x y z ...
#~~~~~
lattice_vector 18.00000000 0.00000000 0.00000000
lattice_vector 0.00000000 18.00000000 0.00000000
lattice_vector 0.00000000 0.00000000 18.00000000

atom_frac 0.91666667 0.08333333 0.91666667 Fe
initial_moment -4
atom_frac 0.41666667 0.41666667 0.58333333 Fe
initial_moment -4
atom_frac 0.08333333 0.91666667 0.08333333 Fe
initial_moment -4
atom_frac 0.91666667 0.25000000 0.75000000 Fe
initial_moment -4
atom_frac 0.91666667 0.25000000 0.08333333 Fe

```

### 5.6.5. 任务实时输出

bpeek 输出所有内容; bpeek -f JOBID 通过重定向的方式将标准输出和标准错误重定向到指定的文件; 如果任务有异常退出, 可以查看计算目录文件下的 output. JOBID来查看具体的原因。



## 5.7. 计费与充值

目前实施充值收费政策。相关内容链接：

<http://hpc.sustech.edu.cn/newsdetail/recharge.html>

### 机时费内转充值流程

● 内转充值方式：提交新版呈批申请。

● OA 呈批模板如下：

标题：XXX 课题组申请本次给计算中心机时充值 XXX 元（20XX 年第 X 次）

内容：

#### 一、经费来源

项目负责人：XXX；

项目名称：XXX；

项目编号：XXX；

项目类型：XXX（例如：国家自然科学基金面上项目等）；

经费号：XXX（注意：建议提前与财务老师确认该经费是否可用于机时费充值）；

充值金额：XXX 元；

#### 二、计算需求

XXX。

● OA 审核流程：

1、经费负责人。

2、PI。

3、系主任审批。

4、分发给科学与工程计算中心谢作扬老师审核。

5、分发给财务段入源老师，确认经费可用，完成经费划扣。

6、分发给科学与工程计算中心李云老师，完成充值。

**注：因后一位审批人需要参考前一位审批人的意见，分发步骤请按顺序进行。请**

**务必最后分发给李云老师充值。**

如对以上流程有疑问，请联系科学与工程计算中心田玲老师，联系方式：

0755-88015833, [tianl@mail.sustech.edu.cn](mailto:tianl@mail.sustech.edu.cn)。

## 5.8. 查询机时及费用

科学与工程计算中心自研开发了《南科大高性能集群用户管理系统》，可使用该系统进行查询：<http://172.18.6.195:18090/>

该系统使用说明书：  
[http://hpc.sustech.edu.cn/ref/HPMS\\_UserGuide.pdf](http://hpc.sustech.edu.cn/ref/HPMS_UserGuide.pdf)

可从中心主页上进行登陆系统：



可从中心主页上下载系统说明书：

资料	更多
"太乙"Parsl使用介绍：基于Python编写面向数据的并行工作流	2022-07-29
[问题解答]: FAQ	2021-08-02
[下载]: 结算单结算流程和常见问题	2022-03-22
[下载]: 机时费充值流程	2023-2-21
[下载]: 账号开通流程	2021-08-12
[下载]: 南科大高性能集群用户管理系统使用手册	2022-10-11
[下载]: LSF用户手册	2019-01-01
[下载]: PBS用户手册	2019-01-01

5.9. 问题反馈

用户对集群使用的相关问题，可以使用工单系统：<http://172.18.6.195:18090/>，或者发邮件至[hpc@sustech.edu.cn](mailto:hpc@sustech.edu.cn)，或者在QQ群里进行提问（QQ群号码：161195492）。另外经过对问题的总结和提炼，我们整理出来FAQ，建议用户先从FAQ里面寻找答案。FAQ链接：<http://hpc.sustech.edu.cn/userguide.html>

资料	更多
"太乙"Parsl使用介绍：基于Python编写面向数据的并行工作流	2022-07-29
[问题解答]: FAQ	2021-08-02
[下载]: 结算单结算流程和常见问题	2022-03-22
[下载]: 机时费充值流程	2023-2-21
[下载]: 账号开通流程	2021-08-12
[下载]: 南科大高性能集群用户管理系统使用手册	2022-10-11
[下载]: LSF用户手册	2019-01-01
[下载]: PBS用户手册	2019-01-01