



北京航空航天大学
BEIHANG UNIVERSITY

高性能计算平台使用介绍

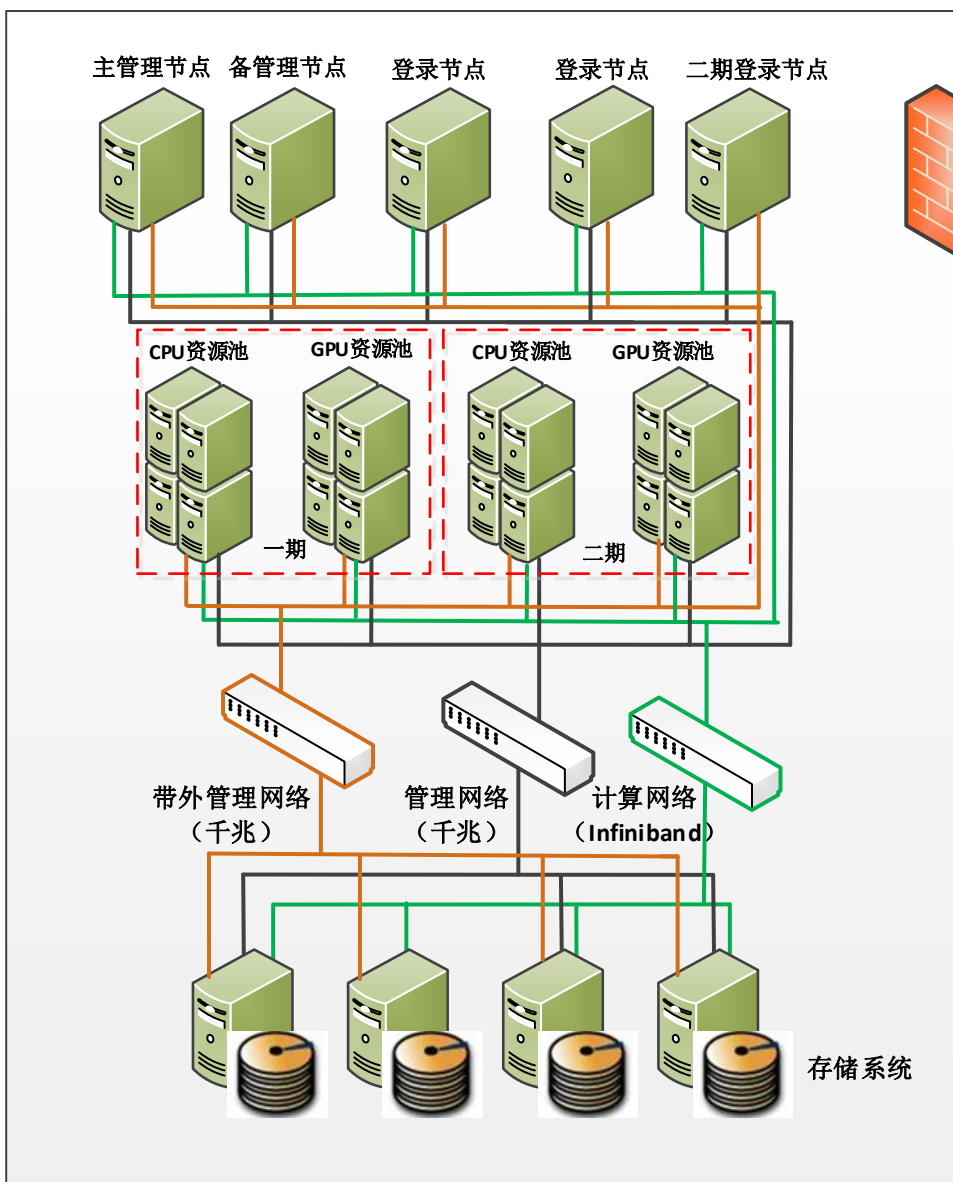
孙佩源

2021/10/14



目录

1	HPC平台结构
2	集群环境
3	作业管理
4	应用示例



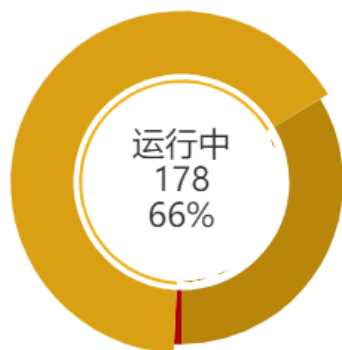
校级高性能计算中心

- 功能节点：3个登录节点
- CPU：260个计算节点
2*Intel 6240 2.6GHz (**9360核**)
- GPU：10台计算节点，每台 8*Nvidia Tesla V100 (**80张卡**)
- 计算网络：100G Infiniband 高速计算网络
- 集群共享存储：1.8PB
- 现有平台计算峰值为**1250万亿次**
- 3期建设：CPU超过万核，GPU上百张卡，计算峰值将达到**1800万亿次**

集群状态

节点

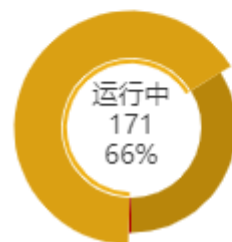
总数	271
可用数	91
运行中	178
不可用	2



CPU节点

节点

总数	260
可用数	88
运行中	171
不可用	1



核心

总数	9360
可用数	4346
运行中	
不可用	



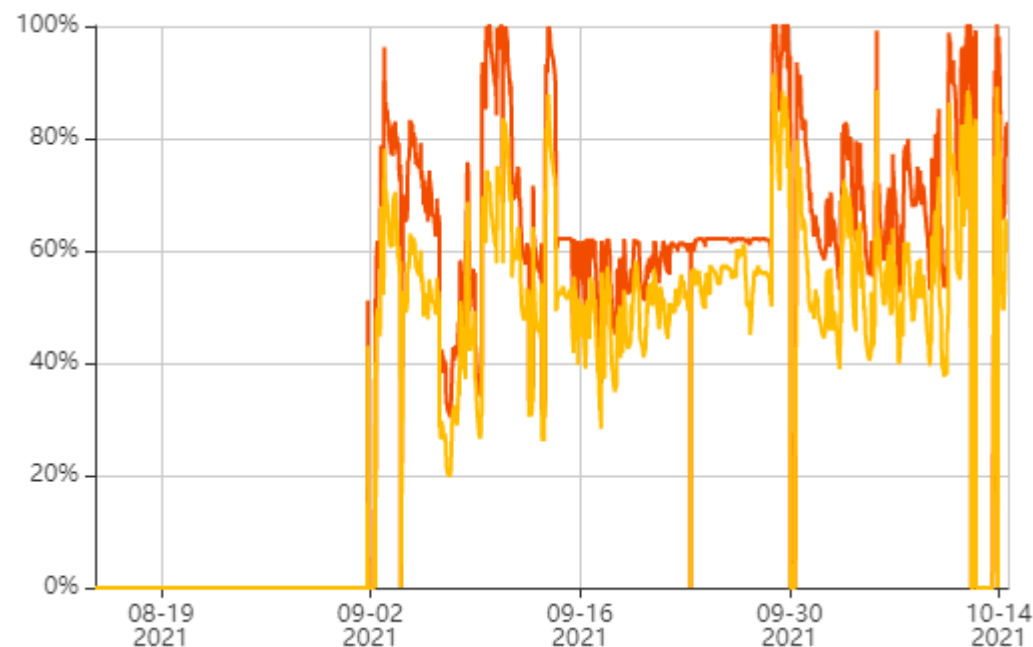
GPU节点

节点

总数	10
可用数	2

核心

总数	360
可用数	149



应用层

大数据/人工智能应用

- 数据分析以交互式或RestAPI为主
- HPC以批处理、MPI为主

高性能计算应用

机器学习
数学库

图形图像
库

SQL/Hive
/Pig

JAVA, Scala, Python, IDL

数学库

工作流

Fortran, C, C++, IDL

MapReduce/Spark

MPI/Openmp/Cuda

RDB
PostgresQL

CloudDB/Hbase
Cassandra

调试/优化器

HDFS

ZooKeeper

GPFS/Lustre

Slurm/LSF/Torque

- 数据分析注重高生产率，以数据为主
- HPC注重高的性能，以计算为主
- 数据分析注重数据库、数据操作的流程
- HPC注重计算过程，多任务并发完成一个作业，需要大量的调试和性能优化

系统软件层

虚拟机/容器

物理机

Linux操作系统

Linux操作系统

- 数据分析建立在虚拟机、容器上，需要考虑隔离和安全性
- HPC建立在物理机上，需要考虑系统软件的支持

操作系统层

硬件层

服务器

网络

存储

服务器

网络

存储

- 数据分析以以太网为主，以服务器集群建立分布式存储
- HPC以延时更低的IB为主，大多建立在高性能的SAN/NAS存储上

- 操作系统:采用 CentOS 7.6 版本操作系统;
- 作业调度软件:采用 Slurm 19.04 作业调度系统;
- 并行文件系统:采用 DDN 并行文件系统;
- 编译环境:GNU、GCC、CUDA;
- 并行环境:MPICH、MVAPICH、OPENMPI、Intel MPI;
- 应用软件:matlab , Gromacs、Lammps、R、 VASP(需要授权)、OpenFoam、Ansys(试用)、 Fluent(试用)、CFX(试用)、Matlab、Moose 等
- 其他工具软件:Anaconda、fftw、CAFFE、HDF5、QT、Tensorflow。

1) 计算资源收费标准

序号	费用名称	单位	单价 (元)	QOS	分区名	最大资源限制	最大运行时长	服务质量
1	CPU 计算费	核/时	0.05	cpu-low	cpu-low	无	7天	低
2		核/时	0.07	cpu-normal	cpu-normal	无	7天	中
3		核/时	0.1	cpu-high	cpu-high	无	7天	高
4		核/时	0.05	cpu-quota	cpu-quota	无	7天	学科科研优先
5	GPU 计算费	卡/时	2.50	gpu-low	gpu-low	2卡、8核	7天	低
6		卡/时	3.75	gpu-normal	gpu-normal	4卡、16核	7天	中
7		卡/时	5.00	gpu-high	gpu-high	8卡、36核	7天	高
8		卡/时	2.50	gpu-quota	gpu-quota	8卡、36核	7天	学科科研优先

计费单位:

CPU: 核小时

GPU: 卡小时

***-low** , 优先级最低的作业队列,计算费用最便宜

***-normal** , 优先级比*-low 高,计算费用同样略高于*-low

***-high** , 优先级和计算费用比前两个队列高

***-quota** , 该队列为学科科研优先队列,具有最高优先级,只有**经相关部门认定后购买的**机时才能享受该队列



目录

1

HPC平台结构

2

集群环境

3

作业管理

4

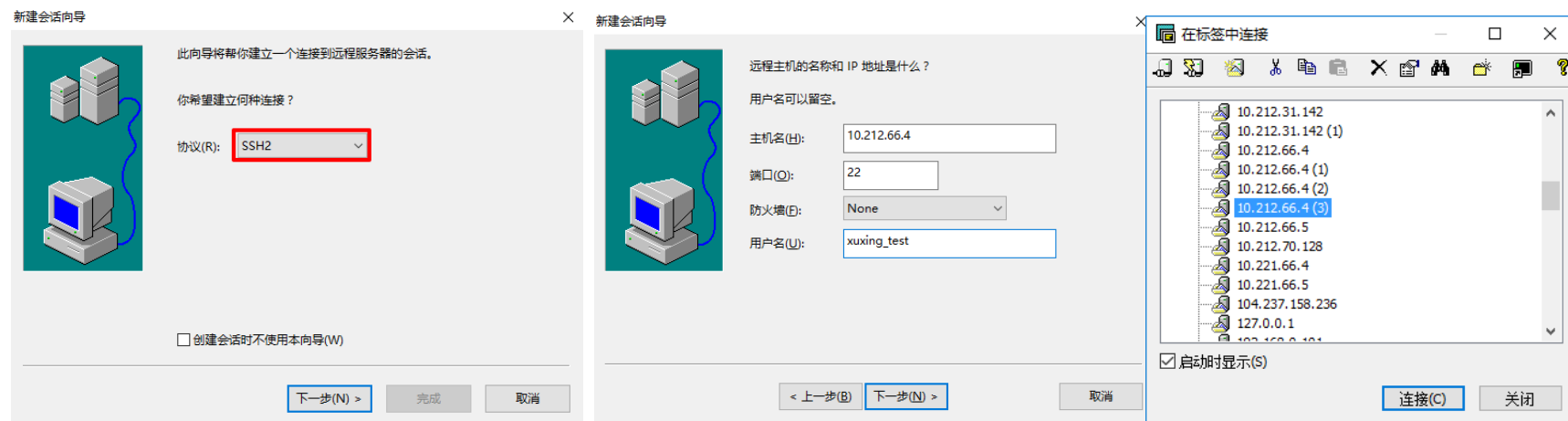
应用示例

- 1) 管理节点和登录节点设置为 root 不能登录;
- 2) 短时间内 3 次密码错误登录,那么登录时所使用 IP 将被自动封锁 5 分钟,需要等待 5 分钟再认证。
 - **注意：**密码不能使用123456或admin123等简单数字字母组成，应使用较为复杂的数字、大小字母、字符等组合形式，以避免账号被破解导致整个高算平台被学校防火墙封禁，以及黑客盗用。
- 3) 登录节点不能运行负载大的任务,设置定时任务,对负载大的任务进行定时清理。
- 4) 本系统用户默认的存储配额为 2TB,存储空间超出 2TB 以后读写文件将提示错误。

- Linux / Mac用户可以直接使用终端登录
- Windows用户建议采用 XShell 或 Putty 等软件登录
- 登录节点的 IP 地址
为:10.212.66.4/10.212.66.5/10.212.70.128
- 使用远程终端登录集群：

XShell / Putty / SecureCRT

- (1) 新建会话，选择SSH协议
- (2) 填入登陆节点IP地址和端口号
- (3) 点击“完成”，登陆系统

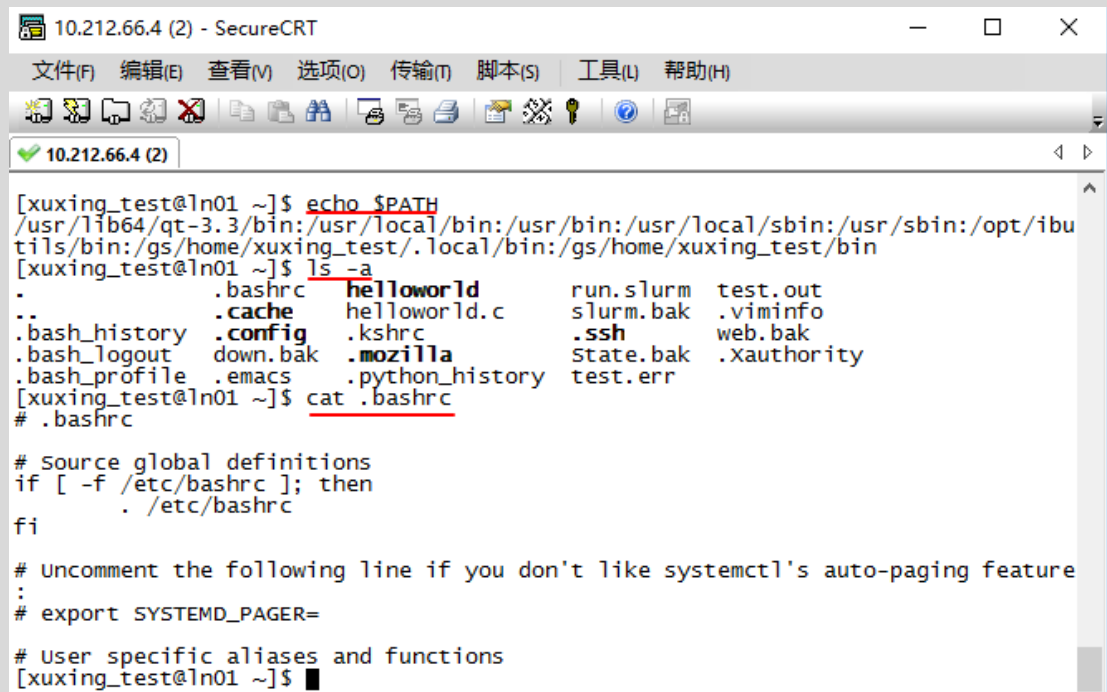


-

- 程序源码准备完毕后,您需要为程序搭建其所需的运行环境
 - python, MPI Library, Compiler, etc (with diverse versions)

如何获知程序运行环境?

`$PATH`, `$LD_LIBRARY_LOAD` etc in `.bashrc` file



```
[xuxing_test@ln01 ~]$ echo $PATH
/usr/lib64/qt-3.3/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/opt/ib
utils/bin:/gs/home/xuxing_test/.local/bin:/gs/home/xuxing_test/bin
[xuxing_test@ln01 ~]$ ls -a
.          .bashrc  helloworld  run.slurm  test.out
..         .cache   helloworld.c slurm.bak  .viminfo
.bash_history .config  .kshrc      .ssh       web.bak
.bash_logout down.bak  .mozilla    State.bak  .Xauthority
.bash_profile .emacs   .python_history test.err
[xuxing_test@ln01 ~]$ cat .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

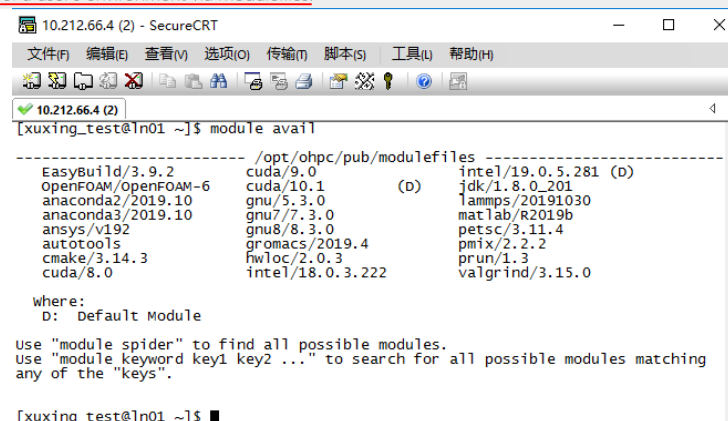
# Uncomment the following line if you don't like systemctl's auto-paging feature
:
# export SYSTEMD_PAGER=

# User specific aliases and functions
[xuxing_test@ln01 ~]$
```



ENVIRONMENT
MODULES

Welcome to the Environment Modules open source project. [The Environment Modules package provides for the dynamic modification of a user's environment via modulefiles.](#)



```
[xuxing_test@ln01 ~]$ module avail
----- /opt/ohpc/pub/modulefiles -----
EasyBuild/3.9.2          cuda/9.0          intel/19.0.5.281 (D)
OpenFOAM/OpenFOAM-6     cuda/10.1         (D)      jdk/1.8.0_201
anaconda2/2019.10       gnu/5.3.0         lammps/20191030
anaconda3/2019.10       gnu7/7.3.0        matlab/R2019b
ansys/v192              gnu8/8.3.0        petosc/3.11.4
autotools               gromacs/2019.4    pmix/2.2.2
cmake/3.14.3            hwloc/2.0.3       prun/1.3
cuda/8.0                intel/18.0.3.222  valgrind/3.15.0

where:
D: Default Module

use "module spider" to find all possible modules.
use "module keyword key1 key2 ..." to search for all possible modules matching
any of the "keys".

[xuxing_test@ln01 ~]$
```

- 查看系统可用的模块(module)

```
[xuxing_test@ln01 ~]$ module avail
```

```
----- /opt/ohpc/pub/modulefiles -----  
EasyBuild/3.9.2      cuda/9.0      intel/19.0.5.281 (D)  
OpenFOAM/OpenFOAM-6  cuda/10.1    (D) jdk/1.8.0_201  
anaconda2/2019.10    gnu/5.3.0    lammmps/20191030  
anaconda3/2019.10 (L) gnu7/7.3.0    matlab/R2019b  
ansys/v192           gnu8/8.3.0    petsc/3.11.4  
autotools            gromacs/2019.4 pmix/2.2.2  
cmake/3.14.3         hwloc/2.0.3    prun/1.3  
cuda/8.0             intel/18.0.3.222 valgrind/3.15.0
```

- 加载所需模块

```
[xuxing_test@ln01 ~]$ module load anaconda3
```

```
[xuxing_test@ln01 ~]$ python
```

```
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
```

```
[GCC 7.3.0] :: Anaconda, Inc. on linux
```

```
Type "help", "copyright", "credits" or "license" for more  
information.
```

```
>>>
```

- 卸载模块(module)

```
[xuxing_test@ln01 ~]$ module list
```

```
Currently Loaded Modules:
```

```
1) anaconda3/2019.10
```

```
[xuxing_test@ln01 ~]$ module unload anaconda3/2019.10
```

```
[xuxing_test@ln01 ~]$ python
```

```
Python 2.7.5 (default, Nov 16 2020, 22:23:17)
```

```
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux2
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>>
```



目录

1

HPC平台结构

2

集群环境

3

作业管理

4

应用示例

- 本平台配置**Slurm**作业调度系统，SLURM（Simple Linux Utility for Resource Management）是一种可用于大型计算节点集群的高度可伸缩和容错的集群管理器和作业调度系统，被世界范围内的超级计算机和计算集群广泛采用。
- 目前在国内主要超级计算机上均采用Slurm作为作业调度系统，包括“天河二号”、“神威太湖之光”、E级原型机等；
- 常用用户命令
 - ***sinfo***:查看系统分区状态信息
 - ***squeue***:查看系统作业状态信息
 - ***scontrol***:查看详细作业/队列/节点信息

- 查看系统分区状态信息

分区节点宕机

```
[xuxing_test@ln01 ~]$ sinfo
```

```
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
```

```
normal drain infinite 1 down* compute-081
```

分区内节点无法调度

```
normal drain infinite 98 drain compute-[161-194,196-257,259-260]
```

```
normal drain infinite 85 mix compute-[001-002,005,008,010-012,017-018,020,022-032,034-035,038-044,048-053,060-063,065-066,068,072,075-079,082-083,088,090-092,094-095,097-099,101,103-105,108,110,112-114,117,120-130,134-135,143,145]
```

已分配

```
normal drain infinite 61 alloc compute-[003-004,006-007,009,013-016,021,033,036-037,045-047,054-059,064,067,069-071,073-074,084,089,093,096,100,102,106-107,109,111,115-116,118-119,131-133,137-139,146-149,151-152,154-158,160]
```

空闲，可调度新作业

```
normal drain infinite 13 idle compute-[019,080,085-087,136,140-142,144,150,153,159]
```

```
gpu drain infinite 1 drain* gpu-1
```

节点在运行作业，但有部分空闲核心

```
gpu drain infinite 5 drain gpu-[01-05]
```

```
gpu drain infinite 3 mix gpu-[02-04]
```

```
gpu drain infinite 1 alloc gpu-01
```

```
cpu-low up infinite 1 down* compute-081
```

```
cpu-low up infinite 100 drain compute-[161-260]
```

```
.....
```

查看系统作业状态信息

作业当前运行状态

```
[xuxing_test@ln01 ~]$ squeue
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	ODELIST(REASON)
281993	gpu-quota	bash	zhengcl	PD	0:00	8	(Nodes required for
281994	gpu-quota	bash	wangziha	PD	0:00	8	(Nodes required for
310656	cpu-quota	ring1000	pc183732	PD	0:00	182038	(PartitionConfig)
298759	gpu-norma	train	wangrp	PD	0:00	2	(PartitionNodeLimit)
299733	gpu-norma	train_co	wangrp	PD	0:00	2	(PartitionNodeLimit)
299867	gpu-norma	train_v1	wangrp	PD	0:00	2	(PartitionNodeLimit)
310116	gpu-quota	myFirstG	chenhao	PD	0:00	1	(QOSMaxGRESPerUser)
317785	gpu-quota	test_hpc	djhuae	PD	0:00	29127	(PartitionNodeLimit)
332837	gpu-quota	main	pc183732	PD	0:00	18204	(PartitionNodeLimit)
1201918	gpu-high	vos	statchao	R	2-20:48:57	1	gpu-01
1201921	cpu-low	in.oinde	dengyao7	R	6-08:40:42	3	compute-[045-047]
1205053	gpu-quota	2_4.txt	huiwang	R	2:15:20	1	gpu-04
1201933	cpu-high	in.tic1f	dengyao7	R	6-08:40:42	3	compute-[054-056]
1201934	cpu-high	in.o1f	dengyao7	R	6-08:40:42	3	compute-[057-059]
1204910	gpu-quota	UGC5	sy193920	R	7:29:19	1	gpu-03
1204909	gpu-quota	UGC3	sy193920	R	7:29:22	1	gpu-03
1204907	gpu-quota	UGC6	sy193920	R	7:31:07	1	gpu-02
1204908	gpu-quota	UGC7	sy193920	R	7:31:07	1	gpu-03
1204905	gpu-quota	UGC4	sy193920	R	7:31:13	1	gpu-02
1204903	gpu-low	UGC2	sy193920	R	7:31:16	1	gpu-02
1204902	gpu-low	UGC1	sy193920	R	7:31:19	1	gpu-02
1204901	gpu-quota	UGC0	sy193920	R	7:31:22	1	gpu-02
1204891	gpu-norma	bash	zhoukang	R	7:46:29	1	gpu-03
1204002	cpu-norma	Vs-Mo-O	sunjiami	R	21:08:53	1	compute-116
1204003	cpu-norma	Vs-Mo-O	sunjiami	R	21:08:53	1	compute-138
1203992	gpu-quota	UGC2	sy193920	R	22:15:12	1	gpu-02
1203991	gpu-quota	UGC1	sy193920	R	22:15:13	1	gpu-02
1203961	gpu-low	POINTNET	zhubing	R	1-00:50:24	1	gpu-04
1202812	cpu-quota	nohup	by180910	R	4-09:59:39	1	compute-001
1205073	gpu-norma	SOL-TC	royzh	R	27:32	1	gpu-04
1205059	gpu-quota	gputest4	wangxuey	R	1:32:44	1	gpu-03
1205056	gpu-quota	gputest3	wangxuey	R	2:07:06	1	gpu-02
1204942	gpu-quota	2_3.txt	huiwang	R	4:53:34	1	gpu-04
1204941	gpu-quota	2_2.txt	huiwang	R	4:53:38	1	gpu-04
1204940	gpu-quota	2_1.txt	huiwang	R	4:53:42	1	gpu-04

- 查看系统作业状态信息

查看user的所有作业信息

```
[xuxing_test@ln01 ~]$ squeue | grep jiangyin
```

1203538	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-076
1203539	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-076
1203540	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-076
1203542	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-078
1203543	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-094
1203544	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-094
1203546	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-094
1203547	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-095
1203548	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-095
1203550	cpu-norma	test	jiangyin	R	2-10:58:35	1	compute-095

自定义格式输出作业信息

```
[xuxing_test@ln01 ~]$ squeue -o '%P %j %u %T %M %D %R %a %C %q %V %Y' | head -n 10
```

PARTITION	NAME	USER	STATE	TIME	NODES	ODELIST(REASON)	ACCOUNT	CPUS	QOS	SUBMIT_TIME	SCHEDNODES
gpu-quota	bash	zhengcl	PENDING	0:00	8	(Nodes required for job are DOWN, DRAINED or reserved for jobs in higher prio					
gpu-quota	bash	wangzihan15	PENDING	0:00	8	(Nodes required for job are DOWN, DRAINED or reserved for jobs in higher					
cpu-quota	ring10000	pc18373205	PENDING	0:00	1820388	(PartitionConfig)	liyc	10000	normal	2021-05-06T17:41:40	(null)
gpu-normal	train	wangrp	PENDING	0:00	2	(PartitionNodeLimit)	songxiao	16	normal	2021-04-15T19:27:27	(null)
gpu-normal	train_copy	wangrp	PENDING	0:00	2	(PartitionNodeLimit)	songxiao	16	normal	2021-04-18T13:23:29	(null)
gpu-normal	train_v18	wangrp	PENDING	0:00	2	(PartitionNodeLimit)	songxiao	16	normal	2021-04-18T16:40:56	(null)
gpu-quota	myFirstGPUJob	chenhao	PENDING	0:00	1	(QOSMaxGRESPerUser)	shizw	6	normal	2021-05-05T00:08:57	(null)
gpu-quota	test_hpc_2	djhuae	PENDING	0:00	29127	(PartitionNodeLimit)	liyc	16	normal	2021-05-21T21:37:20	(null)
qpu-quota	main	pc18373205	PENDING	0:00	18204	(PartitionNodeLimit)	liyc	10	normal	2021-06-23T14:46:38	(null)

- 查看详细作业/队列/节点信息

查看cpu-normal详细信息

```
[xuxing_test@ln01 ~]$ scontrol show partition=cpu-normal
PartitionName=cpu-normal
AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
AllocNodes=ALL Default=NO QoS=cpu-normal
DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=0 LLN=NO MaxCPUsPerNode=UNLIMITED
Nodes=compute-[001-260]
PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO OverSubscribe=NO
OverTimeLimit=NONE PreemptMode=OFF
State=UP TotalCPUs=9360 TotalNodes=260 SelectTypeParameters=NONE
JobDefaults=(null)
DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
```

查看cpu-normal节点名

查看特定节点详细信息

```
[xuxing_test@ln01 ~]$ scontrol show node=compute-076
NodeName=compute-076 Arch=x86_64 CoresPerSocket=18
CPULoad=32 CPULoad=32.29
AvailableFeatures=(null)
ActiveFeatures=(null)
Gres=(null)
NodeAddr=compute-076 NodeHostName=compute-076 Version=19.05.4
OS=Linux 3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018
RealMemory=352000 AllocMem=0 FreeMem=122876 Sockets=2 Boards=1
State=MIXED ThreadsPerCore=1 TmpDisk=0 weight=1 owner=N/A MCS_label=N/A
Partitions=normal,cpu-low,cpu-normal,cpu-high,cpu-quota
BootTime=2020-11-13T03:30:03 slurmdStartTime=2021-07-06T20:46:40
CfgrTRES=cpu=36,mem=352000M,billing=36
AllocTRES=cpu=32
CapWatts=n/a
CurrentWatts=0 AveWatts=0
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
```

- 查看详细作业/队列/节点信息

查看特定作业详细信息

```
[xuxing_test@ln01 ~]$ scontrol show job=1203526
JobId=1203526 JobName=test
  UserId=jiangying(1263) GroupId=jiangying(1263) MCS_label=N/A
  Priority=498 Nice=0 Account=jiangying QOS=normal
  JobState=RUNNING Reason=None Dependency=(null)
  Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
  RunTime=2-11:22:48 TimeLimit=5-00:00:00 TimeMin=N/A
  SubmitTime=2021-09-19T09:53:27 EligibleTime=2021-09-19T09:53:27
  AccrueTime=2021-09-19T09:53:27
  StartTime=2021-09-19T09:53:28 EndTime=2021-09-24T09:53:28 Deadline=N/A
  SuspendTime=None SecsPreSuspend=0 LastSchedEval=2021-09-19T09:53:28
  Partition=cpu-normal AllocNode:Sid=ln01:4682
  ReqNodeList=(null) ExcNodeList=(null)
  NodeList=compute-052
  BatchHost=compute-052
  NumNodes=1 NumCPUs=8 NumTasks=8 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
  TRES=cpu=8,node=1,billing=8
  Socks/Node=* NtasksPerN:B:S:C=8:0:*:* Corespec=*
  MinCPUsNode=8 MinMemoryNode=0 MinTmpDiskNode=0
  Features=(null) DelayBoot=00:00:00
  OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
  Command=/gs/home/jiangying/chenyg/ABdiblock_particle/DeltaFE_kapa/dpc_0.5D0_entropy/0.1383/RAB/0.5_min_0.0_7/0.375/script.slurm
  WorkDir=/gs/home/jiangying/chenyg/ABdiblock_particle/DeltaFE_kapa/dpc_0.5D0_entropy/0.1383/RAB/0.5_min_0.0_7/0.375
  StdErr=/gs/home/jiangying/chenyg/ABdiblock_particle/DeltaFE_kapa/dpc_0.5D0_entropy/0.1383/RAB/0.5_min_0.0_7/0.375/test.err
  StdIn=/dev/null
  StdOut=/gs/home/jiangying/chenyg/ABdiblock_particle/DeltaFE_kapa/dpc_0.5D0_entropy/0.1383/RAB/0.5_min_0.0_7/0.375/test.out
  Power=
```

- 作业提交（三种模式）

- *sbatch* : 提交批处理作业

对于批处理作业（提交后立即返回该命令行终端，用户可进行其它操作）使用sbatch命令提交作业脚本，作业被调度运行后，在所分配的首个节点上执行作业脚本。在作业脚本中也可使用srun命令加载作业任务。提交时采用的命令行终端终止，也不影响作业运行。**推荐用户主要使用该模式提交作业**

- *srun* : 提交交互式作业

资源分配与任务加载两步均通过srun命令进行：当在登录shell中执行srun命令时，srun首先向系统提交作业请求并等待资源分配，然后在所分配的节点上加载作业任务。采用该模式，用户在该终端需等待任务结束才能继续其它操作，在作业结束前，如果提交时的命令行终端断开，则任务终止。**一般用于短时间小作业测试。**

- *salloc* : 提交节点资源获取作业

分配作业模式类似于交互式作业模式和批处理作业模式的融合。用户需指定所需要的资源条件，向资源管理器提出作业的资源分配请求。提交后，作业处于排队，当用户请求资源被满足时，将在用户提交作业的节点上执行用户所指定的命令，指定的命令执行结束后，运行结束，用户申请的资源被释放。在作业结束前，如果提交时的命令行终端断开，则任务终止。**典型用途是分配资源并启动一个shell，然后在这个shell中利用srun运行并行作业。**

- srun交互式提交作业

1. 申请一个节点并且连接到登录节点shell

指定登陆节点名

```
[xuxing_test@ln02 ~]$ srun -p cpu-low -w compute-005 --pty bash
[xuxing_test@compute-005 ~]$ hostname
compute-005
```

2. 配合tmux实现会话保存

①创建新会话并attach

```
[xuxing_test@ln01 ~]$ tmux new -s compute-005-tmux
```

②在tmux会话里登陆计算节点

```
[xuxing_test@ln01 ~]$ srun -w compute-005 --pty bash
srun: job 1205104 queued and waiting for resources
srun: job 1205104 has been allocated resources
[xuxing_test@compute-005 ~]$
```

tmux会话名称

```
[compute-0 0:xuxing_test@ln01:~*]
```

③ Ctrl-b d脱离会话

```
[detached]
[xuxing_test@ln01 ~]$
```

④ 重新attach会话

```
[xuxing_test@ln01 ~]$ tmux ls
compute-005-tmux: 1 windows (created Tue Sep 21 22:14:35 2021) [237x67]
[xuxing_test@ln01 ~]$ tmux attach -t compute-005-tmux
```


- sbatch批量提交作业

1. 编写Slurm作业脚本

```
[xuxing_test@ln01 ~]$ cat run.slurm
#!/bin/bash

#SBATCH -J sunpy-test
#SBATCH -p cpu-low
#SBATCH -N 1
#SBATCH -n 1
#SBATCH -t 5:00
#SBATCH -o test.out
#SBATCH -e test.err

module load anaconda3

./helloworld

sleep 200
```

2. 提交作业脚本并查看作业信息

```
[xuxing_test@ln01 ~]$ sbatch run.slurm
Submitted batch job 1205121
[xuxing_test@ln01 ~]$ scontrol show job=1205121
JobId=1205121 JobName=sunpy-test
  UserId=xuxing_test(3363) GroupId=xuxing_test(3364) MCS_label=N/A
  Priority=499 Nice=0 Account=xuxing_test QOS=normal
  JobState=RUNNING Reason=None Dependency=(null)
  Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
  RunTime=00:00:09 TimeLimit=00:05:00 TimeMin=N/A
  SubmitTime=2021-09-21T22:44:37 EligibleTime=2021-09-21T22:44:37
  AccrueTime=2021-09-21T22:44:37
  StartTime=2021-09-21T22:44:37 EndTime=2021-09-21T22:49:37 Deadline=N/A
  SuspendTime=None SecsPreSuspend=0 LastSchedEval=2021-09-21T22:44:37
  Partition=cpu-low AllocNode:Sid=ln01:31566
  ReqNodeList=(null) ExcNodeList=(null)
  NodeList=compute-031
  BatchHost=compute-031
  NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
  TRES=cpu=1,node=1,billing=1
  Socks/Node=* NtasksPerN:B:S:C=0:0:*:* Corespec=*
  MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
  Features=(null) DelayBoot=00:00:00
  OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
  Command=/gs/home/xuxing_test/run.slurm
  WorkDir=/gs/home/xuxing_test
  StdErr=/gs/home/xuxing_test/test.err
  StdIn=/dev/null
  StdOut=/gs/home/xuxing_test/test.out
  Power=
```


- sacct查看历史作业信息

1. 缺省命令列出个人用户所有历史作业信息

```
[xuxing_test@ln02 ~]$ sacct
```

JobID	JobName	Partition	Account	AllocCPUS	State	ExitCode
1205295	bash	cpu-quota	xuxing_te+	1	CANCELLED+	0:0
1205301	bash	cpu-quota	xuxing_te+	1	CANCELLED+	0:0
1205304	bash	cpu-quota	xuxing_te+	1	CANCELLED+	0:0
1205305	bash	cpu-quota	xuxing_te+	1	CANCELLED+	0:0
1205310	bash	cpu-quota	xuxing_te+	1	CANCELLED+	0:0

2. 根据作业号查看历史信息

```
[xuxing_test@ln02 ~]$ sacct --job=1205230
```

JobID	JobName	Partition	Account	AllocCPUS	State	ExitCode
1205230	hetero	cpu-quota	peizhet	162	FAILED	9:0
1205230.bat+	batch		peizhet	18	FAILED	9:0
1205230.0	hostname		peizhet	162	COMPLETED	0:0
1205230.1	pmi_proxy		peizhet	9	COMPLETED	0:0

3. 根据作业号查看退出原因

```
[xuxing_test@ln02 ~]$ sjobexitmod -l 1205230
```

JobID	Account	NNodes	NodeList	State	ExitCode	DerivedExitCode	Comment
1205230	peizhet	9	compute-[061,0+	FAILED	9:0	0:0	



目录

1

HPC平台结构

2

集群环境

3

作业管理

4

应用示例

- 下载开源代码
 - git clone https://github.com/mperlet/matrix_multiplication.git
- 加载环境并编译可执行程序
 - module load intel/18.0.3.222
 - module load openmpi3/3.1.4
 - make mpi
 - mpicc \$(TUNE) \$(CFLAGS) -o bin/mpi \$(LIBS) src/mpi.c
 - generating test matrix data
 - ./random_float_matrix.py 1000 1000 > demo-mpi-mat[1|2].dat

- 编写Slurm作业脚本

```
[xuxing_test@ln01' matrix_multiplication]$ cat demo.slurm
#!/bin/bash
```

```
#SBATCH -J mpi-demo
#SBATCH -p cpu-low
#SBATCH -N 2
#SBATCH --ntasks-per-node=4
#SBATCH -t 7-00:00
#SBATCH -o mpi-demo.out
#SBATCH -e mpi-demo.err
```

```
echo Time is `date`
echo Directory is $PWD
echo This job runs on the following nodes:
echo $SLURM_JOB_NODELIST
echo This job has allocated $SLURM_JOB_CPUS_PER_NODE cpu cores.
```

```
srun hostname | sort > machinefile.${SLURM_JOB_ID}
NP=$(cat machinefile.${SLURM_JOB_ID} | wc -l)
```

```
module load intel/19.0.5.281
module load openmpi3/3.1.4
```

```
mpirun -np ${NP} -machinefile ./machinefile.${SLURM_JOB_ID} ./bin/mpi demo-mpi-mat1.dat demo-mpi-mat2.dat
```

- 提交作业

Q1: 无法登陆超算平台？

A: 尝试切换登陆节点，如果均无法登陆请联系超算老师帮忙解封

Q2: 程序运行崩溃，没有错误信息？

A: 尝试使用gdb a.out core调试错误信息

Q3: 如何自行安装软件？

A: Linux下使用./configure --prefix=/gs/home/your_home_dir; make; make install

Q4: 提交作业显示failed？

A: `squeue`查看作业失败原因

Q5: 程序在超算平台没有得到加速？

A: 程序耗时主要包括计算和I/O，首先对程序做性能测量，提升并行化是取得良好加速比的关键

Q6: 如何对程序做并行优化？

A: 可参考并行计算相关学习资料，后续平台也会根据大家反馈开展相关培训

欢迎大家使用！ 谢谢！

- 学生自己查看机时（没有导师密码）
- 师兄毕业后账号是否仍有效
- 手册中conda装不上
- 作业无故退出（后提交任务会把之前任务挤掉？）
- 计算软件安装
 - 源码获取
 - makefile文件的解读，安装的路径，环境等
 - 安装和编译过程中最常见的一些错误
 - 比如cp2k