SLURM调度系统的基础使用

目录

目录

概述

- Slurm(Simple Linux Utility for Resource Management， http://slurm.schedmd.com/)是开源的、具有容错性和高度可扩展的Linux计算集群资源管理和作业调度系统，适用于大型和小型 Linux 计算集群。
- Linux计算集群可利用Slurm对资源和作业进行管理，以避免相互干扰，提高运行效率。
- 所有需运行的作业，无论是用于程序调试还是业务计算，都可以通过交互式并行 srun 、批处理式 sbatch 或分配式 salloc 等命令提交，提交后可以利用相关命令查询修改作业状态等。

术语

- **Compute Node**：计算节点，**实际运行作业计算任务的节点。**
- **Login Node**：用户登录节点，用于用户登录的节点。通常用于用户**修改代码、编译程序和作业提交。**
- **socket**：CPU插槽，可以简单理解为CPU。
- **core**：CPU核，单颗CPU可以具有多颗CPU核。
- **job**：作业。
- **job step**：作业步，单个作业（job）可以有个多作业步。
- **tasks**：任务数，单个作业或作业步可有多个任务，一般一个任务需一个CPU核，可理解为所需的CPU核数。
- **Partition**：队列、分区。用于对**计算节点、作业并行规模、作业时长、用户**等进行分组管理，以合理分配资源。

目录

常用命令

命令	功能
sinfo	查看集群队列和节点状态信息
squeue	查看作业和作业步状态
sacct	报告作业/作业步的记账信息。
sstat	报告正在运行的作业/作业步的信息，包含状态采集。

常用命令 - sinfo

- **sinfo命令常用参数**

参数	解释
-a, --all	显示全部队列信息
-n, --nodes=<nodes>	显示节点状态信息
-N, --Node	以每行一个节点方式显示信息，即显示各节点信息。
-p, --partition=<partition>	显示队列信息
-t, --states=<states>	仅显示指定状态的节点状态信息
-r, --responding	仅显示响应的节点信息
-d, --dead	仅显示无响应或已宕机节点状态信息
-R, --list-reasons	显示不响应（down、drained、fail或failing状态）节点的原因
-s	显示摘要信息
-l, --long	显示详细信息

常用命令 - sinfo

■ sinfo 常用命令示例

✓ 查看所有节点信息

```
sinfo
```

```
[root@login01 ~]# sinfo
PARTITION     AVAIL  TIMELIMIT  NODES  STATE NODELIST
normal           up   infinite     17 drain* a3110n[16-19],a3401n16
normal           up   infinite      1  down* b3308r2n5
```

✓ 查看某个分区的节点信息

```
sinfo –p <分区名>
```

```
[root@login01 ~]# sinfo -p normal
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
normal       up   infinite     17 drain* a3110n[16-19],a3401n16
normal       up   infinite      1  down* b3308r2n5
```

常用命令 - sinfo

■ sinfo 常用命令示例

✓ 查看某些节点的状态信息

```
sinfo –n <节点名>
```

```
[root@login01 ~]# sinfo -n b3308r2n5
PARTITION     AVAIL  TIMELIMIT  NODES  STATE NODELIST
normal           up   infinite      1  down* b3308r2n5
```

✓ 按照指定状态查看节点信息

```
sinfo –t <节点状态>
```

```
[root@login01 ~]# sinfo -t drain
PARTITION     AVAIL  TIMELIMIT  NODES  STATE NODELIST
normal           up   infinite     17 drain* a3110n[16-19],a3401n16
```

常用命令 - sinfo

■ sinfo 命令输出结果示例

```
adev0: sinfo
PARTITION AVAIL  TIMELIMIT NODES  STATE NODELIST
debug*       up      30:00     2  down* adev[1-2]
debug*       up      30:00     3   idle adev[3-5]
batch        up      30:00     3  down* adev[6,13,15]
batch        up      30:00     3  alloc adev[7-8,14]
batch        up      30:00     4   idle adev[9-12]
```

■ 节点状态说明

节点状态	描述
idle	节点可用
down/drain	节点不可用
alloc/mix	节点被占用

■ sinfo 命令输出格式

PARTITION	AVAIL	TIMELIMIT	NODES	STATE	NODELIST
队列名（分区名）	队列状态（可用性）	默认时间	节点数量	节点状态	节点名列表

常用命令 – squeue

- **squeue 命令参数**

参数	解释
-A, --account=<account(s)>	查询指定账号的作业，默认全部账号下的作业
-j, --job=<job(s)>	已逗号分隔指定显示的jobid，默认显示全部
-n, --name=<job_name(s)>	逗号分隔指定的作业名称
-o, --format=<format>	指定显示的信息
-p, --partition=<partition(s)>	逗号分隔指定队列中的作业
-u, --user=<user_name(s)>	逗号分隔指定用户的作业
-s, --steps	显示作业步
-t, --states=<states>	显示特定状态的作业信息
-w, --nodelist=<hostlist>	显示特定节点的作业信息

常用命令 - squeue

■ squeue 常用命令示例

✓ 查看所有作业信息

```
squeue
```

```
[root@login01 ~]# squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4441105    bigmem    test1    user0  R    1:32:52     16 b2106r7n[1-8],b2107r1n[1-8]
           4441229    bigmem    test2    user0  R      49:16      8 b2107r7n[1,5-7],b2108r3n[5-8]
           4431474    normal   py_job    user1  R 1-09:10:51      1 b2107r4n2
           4397345    normal Pleistoc    user2  R 6-04:00:26      6 b2107r7n8,b2108r1n[1-5]
```

✓ 查看某个分区的作业信息

```
squeue –p <分区名>
```

```
[root@login01 ~]# squeue -p bigmem
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4441105    bigmem    test1    user0  R    1:32:52     16 b2106r7n[1-8],b2107r1n[1-8]
           4441229    bigmem    test2    user0  R      49:16      8 b2107r7n[1,5-7],b2108r3n[5-8]
```

常用命令 - squeue

■ squeue 常用命令示例

✓ 查看某些节点的作业信息

```
squeue –w <节点名>
```

```
[root@login01 ~]# squeue -w b2107r1n1
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4441105    bigmem    test1    user0  R    1:32:52     16 b2106r7n[1-8],b2107r1n[1-8]
```

✓ 按照指定状态查看作业信息

```
squeue –t <作业状态>
```

```
[root@login01 ~]# squeue -t R
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4441105    bigmem    test1    user0  R    1:32:52     16 b2106r7n[1-8],b2107r1n[1-8]
           4441229    bigmem    test2    user0  R      49:16      8 b2107r7n[1,5-7],b2108r3n[5-8]
           4431474    normal   py_job    user1  R 1-09:10:51      1 b2107r4n2
           4397345    normal Pleistoc    user2  R 6-04:00:26      6 b2107r7n8,b2108r1n[1-5]
```

常用命令 – squeue

- **squeue 命令查询排队和运行状态的作业**

```
JOBID PARTITION   NAME   USER ST   TIME NODES NODELIST(REASON)
65646     batch   chem   mike  R 24:19     2 adev[7-8]
65647     batch    bio   joan  R  0:09     1 adev14
65648     batch   math   phil PD  0:00     6 (Resources)
```

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
作业号	队列名（分区名）	作业名	用户名	**作业状态**	已运行时间	节点数	**分配给的节点名列表（原因）**

- **作业状态**

作业状态	描述
R	作业运行中
PD	作业排队中
CG	作业完成中

- **原因**

原因	描述
Resources	作业等待其所需的资源可用
Priority	作业所需的队列存在高等级作业或者预留
QOSJobLimit	作业的QOS达到其最大作业计数

常用命令 - sacct

- **sacct命令显示运行的或已完成作业或作业步的记账信息。**

参数	解释
--b	显示简要信息，主要包含：jobid、status和exitcode。
-l	显示详细信息
--n	不显示信息头（显示出的信息的第一行，表示个列含义）。
--N <node_list>	显示运行在特定节点的作业记账信息
-E, --endtime=<end_time>	查询在指定时间之前，任何状态的作业.如果通过-s参数指定状态则返回在此时间之前的指定状态的作业，有效格式为： **HH:MM[:SS] [AM|PM]** **MMDD[YY] or MM/DD[/YY] or MM.DD[.YY]** **MM/DD[/YY]-HH:MM[:SS]** **YYYY-MM-DD[THH:MM[:SS]]**
-S, --starttime=<starttime>	在指定时间后，任何状态的作业
--j <jobid>	显示在特定节点数量上运行的作业
-o, --format=<format>	指定显示字段以逗号分隔

常用命令 - sacct

■ sacct 常用命令示例

✓ 查看历史作业信息

```
sacct
```

```
[user3@login01 ~]$ sacct
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
1376006      mem_used_+      debug      user3        780    PENDING      0:0
1376007      mem_used_+      debug      user3        780    PENDING      0:0
1376005      mem_used_+      debug      user3        780  COMPLETED      0:0
1376005.bat+ mem_used_+                 user3        780  COMPLETED      0:0
1376005.ext+ mem_used_+                 user3        780  COMPLETED      0:0
```

✓ 查看运行在某个节点上的历史作业信息

```
sacct -N <节点名>
```

```
[user3@login01 ~]$ sacct -N b2100r3n2
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
1376005      mem_used_+      debug      user3        780  COMPLETED      0:0
1376005.bat+ mem_used_+                 user3        780  COMPLETED      0:0
1376005.ext+ mem_used_+                 user3        780  COMPLETED      0:0
```

常用命令 - sacct

■ sacct 常用命令示例

- ✓ 查看位于a3310n17节点且在2022-01-22 17:09:16 到2022-01-22 17:09:27这个时间段内的历史作业信息

```
sacct -N a3310n17 -S 2022-01-22T17:09:16 -E 2022-01-22T17:09:27
```

```
[user3@login01 ~]$ sacct -N a3310n17 -S 2022-01-22T17:09:16 -E 2022-01-22T17:09:27
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
585847              sleep     normal      user3        128  COMPLETED      0:0
585847.batch        batch                 user3         64  COMPLETED      0:0
585847.exte+       extern                 user3        128  COMPLETED      0:0
585847.0       pmi_proxy                 user3          2  COMPLETED      0:0
```

常用命令 – sacct

- **sacct 命令输出格式说明**

```
[user3@login01 ~]$ sacct
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
1376006      mem_used_+      debug      user3        780    PENDING      0:0
1376007      mem_used_+      debug      user3        780    PENDING      0:0
1376005      mem_used_+      debug      user3        780  COMPLETED      0:0
1376005.bat+ mem_used_+                 user3        780  COMPLETED      0:0
1376005.ext+ mem_used_+                 user3        780  COMPLETED      0:0
```

JobID	JobName	Partition	Account	AllocCPUS	State	ExitCode
作业号	作业名	队列名（分区名）	账户	分配CPU核数	状态	退出码

- **作业（步）状态**

原因	描述
PENDING	作业排队
COMPLETED	作业（步）正常结束
CANCELLED	作业（步）取消
FAILED	作业（步）失败

常用命令 - sstat

- **sstat 命令查看正在运行的作业**

参数	解释
-a, --allsteps	当没有指定步骤时，显示所有步骤。
-e, --helpformat	打印 –o可以指定的所有参数
-i, --pidformat	列出每个作业步骤运行的pid
-o, --format	格式化输出
-j, --jobs	指定作业jobid

常用命令 - sstat

■ sstat 常用命令示例

✓ 查看作业完整信息

```
sstat –j <作业ID>
```

```
[user4@login01 ~]$ sstat -j 4441314
       JobID  MaxVMSize  MaxVMSizeNode  MaxVMSizeTask  AveVMSize     MaxRSS MaxRSSNode MaxRSSTask     AveRSS MaxPages MaxPagesNode   MaxPagesTask   AvePages     MinCPU MinCPUNode MinCPUTask     AveCPU   NTasks AveCPUFreq ReqCPUF
reqMin ReqCPUFreqMax ReqCPUFreqGov ConsumedEnergy  MaxDiskRead MaxDiskReadNode MaxDiskReadTask  AveDiskRead MaxDiskWrite MaxDiskWriteNode MaxDiskWriteTask AveDiskWrite TRESUsageInAve TRESUsageInMax TRESUsageInMaxNode TRESUsageIn
MaxTask TRESUsageInMin TRESUsageInMinNode TRESUsageInMinTask TRESUsageInTot TRESUsageOutAve TRESUsageOutMax TRESUsageOutMaxNode TRESUsageOutMaxTask TRESUsageOutMin TRESUsageOutMinNode TRESUsageOutMinTask TRESUsageOutTot
------------ ---------- -------------- -------------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- -------- ---------- ---------- ---------- -------- ---------- -------
------ ------------ ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
------- -------------- ------------------ ------------------ -------------- -------------- -------------- ------------------ ------------------ -------------- ------------------ ------------------ ---------------
sstat: ROUTE: split_hostlist: hl=a3112n[11-18] tree_width 0
4441314.0     72757752K       a3112n12             1  71725376K  34493236K   a3112n16          5  34143685K     2469     a3112n11            0        542 2-02:10:55   a3112n12          1 2-02:20:15        8     12.75K       U
nknown       Unknown       Unknown              0  34770173912       a3112n12              1  10313902413  11472304781         a3112n11              0   1706689457 cpu=2-02:20:1+ cpu=2-02:26:4+ cpu=a3112n18,ener+ cpu=00:00:0
0,fs/d+ cpu=2-02:10:5+ cpu=a3112n12,ener+ cpu=00:00:00,fs/d+ cpu=16-18:42:+ energy=0,fs/di+ energy=0,fs/di+ energy=a3112n11,fs+           fs/disk=0 energy=0,fs/di+ energy=a3112n11,fs+           fs/disk=2 energy=0,fs/di+
```

✓ 查看输出格式化字段范围

```
sstat –e
```

```
[user4@login01 ~]$ sstat -e
AveCPU              AveCPUFreq          AveDiskRead         AveDiskWrite
AvePages            AveRSS              AveVMSize           ConsumedEnergy
ConsumedEnergyRaw   JobID               MaxDiskRead         MaxDiskReadNode
MaxDiskReadTask     MaxDiskWrite        MaxDiskWriteNode    MaxDiskWriteTask
MaxPages            MaxPagesNode        MaxPagesTask        MaxRSS
MaxRSSNode          MaxRSSTask          MaxVMSize           MaxVMSizeNode
MaxVMSizeTask       MinCPU              MinCPUNode          MinCPUTask
Nodelist            NTasks              Pids                ReqCPUFreq
ReqCPUFreqMin       ReqCPUFreqMax       ReqCPUFreqGov       TRESUsageInAve
TRESUsageInMax      TRESUsageInMaxNode  TRESUsageInMaxTask  TRESUsageInMin
```

常用命令 - sstat

■ sstat 常用命令示例

✓ 查看作业内存消耗等常见信息

sstat –j <作业ID>-o jobid%16,nodelist,ntasks,averss,maxrss,avevmsize,maxvmsize,pids

```
[user4@login01 ~]$ sstat -j 4441314 -o jobid%16,nodelist,ntasks,averss,maxrss,avevmsize,maxvmsize,pids
           JobID             Nodelist   NTasks     AveRSS     MaxRSS  AveVMSize  MaxVMSize                Pids
---------------- -------------------- -------- ---------- ---------- ---------- ---------- --------------------
sstat: ROUTE: split_hostlist: hl=a3112n[11-18] tree_width 0
       4441314.0        a3112n[11-18]        8 34145365.+  34493500K  71725376K  72757752K 10796,10800,10801,1+
```

```
Tasrs: 929 total,  33 runneng, 896 slsspeng,   0 stoppsd,   0 zombes
%Cpu(s): 49.3 us,  0.7 sy,  0.0 ne, 49.8 ed,  0.0 wa,  0.0 he,  0.1 se,  0.0
KeB Msm : 26168352+total, 20240814+urss, 45056840 ussd, 14218540 buuu/4a4hs
KeB Swap: 16364540 total, 16363772 urss,      768 ussd. 21053998+avael Msm

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
10805 user4     20   0 2297064   1.1g  34488 R 105.9  0.4 106:52.78 44sm.sxs
10808 user4     20   0 2291516   1.0g  32916 R 105.9  0.4 107:00.76 44sm.sxs
10815 user4     20   0 2185332 993140  32596 R 105.9  0.4 106:52.93 44sm.sxs
10816 user4     20   0 2174968 975304  34748 R 105.9  0.4 106:58.79 44sm.sxs
10826 user4     20   0 2181504 984864  31320 R 105.9  0.4 106:54.16 44sm.sxs
10800 user4     20   0 2275648   1.1g 125896 R 100.0  0.5 106:29.39 44sm.sxs
10801 user4     20   0 2268852   1.0g  33716 R 100.0  0.4 106:57.87 44sm.sxs
```

目录

作业提交

A

srun

交互式作业提交。运行并行作业，等获得作业分配的资源并运行，作业结束后返回命令行终端。

B

salloc

交互式资源申请。为需实时处理的作业分配资源，提交后等获得作业分配的资源后运行，作业结束后返回命令行终端。

C

sbatch

批处理作业。提交后无需等待立即返回命令行终端

作业提交 - 通用参数

■ **适用于srun、salloc和sbatch命令**

指定参数	参数含义
-J , --job-name	指定作业名称
-p, --partition	指定队列资源
-N, --nodes=<number>	指定节点数量
-n, --ntasks=<number>	指定作业任务数量
--ntasks-per-node=<number>	**指定每节点任务数，最大为节点物理核心数，默认1**
--cpus-per-task=<number>	**指定每任务CPU核心数对应多线程场景，需配合OMP_NUM_THREADS变量，默认1**
--mem=xx	预留申请内存容量，单位MB、GB
--gres=xxx:<number>	申请加速卡数，最大为节点加速卡数量
-w, --nodelist=<node name list>	指定申请的节点，格式可以为node1，node2 或者node[1-10]
-x, --exclude=<node name list>	排除指定的节点，格式可以为node1，node2 或者node[1-10]
-t, --time=DD-HH:MM	指定作业运行时间，例如7-08:00代表7天8小时
--exclusive	设置节点独占

作业提交 - CPU资源分配规则

■ CPU资源分配规则

✓ 单线程时，作业分配的总进程数等于分配的CPU核心数

ntasks-per-node * N<= N * 单节点CPU核心数
n <= N *单节点CPU核心数

```
[zlei@login01 ~]$ srun -p high -N 2 -n 128 --ntasks-per-node=64 hostname
srun: job 4478388 queued and waiting for resources
srun: job 4478388 has been allocated resources
srun: ROUTE: split_hostlist: hl=e2113r4n[3-4] tree_width 0
e2113r4n3
```

✓ 多线程时，作业分配的进程数不等于分配的CPU核心数，还要考虑线程的核心占用情况，且OMP_NUM_THREADS应等于cpus-per-task。

ntasks-per-node * cpus-per-task <= 单节点CPU核心数
n * cpus-per-task <= N *单节点CPU核心数

```
[zlei@login01 ~]$ srun -p high -N 2 -n 64 --ntasks-per-node=32 --cpus-per-task=2 hostname
srun: job 4478390 queued and waiting for resources
srun: job 4478390 has been allocated resources
srun: ROUTE: split_hostlist: hl=e2113r4n[3-4] tree_width 0
e2113r4n3
e2113r4n3
```

作业提交 - srun

■ srun 命令交互式作业提交示例

✓ 指定分区、节点数量和任务数提交作业

```
srun -p <分区> -N <节点数> -n <总任务数> <作业脚本或者命令>
```

```
[user4@login01 ~]$ srun -p normal -N 2 -n 4 hostname
a3107n04
a3107n04
a3107n04
a3107n05
```

✓ 指定分区、节点数量和任务数提交作业

```
srun -p <分区> -w <节点> -n <总任务数> <作业脚本或者命令>
```

```
[user4@login01 ~]$ srun -p normal -w a3107n04 -n 2 hostname
a3107n04
a3107n04
a3107n04
a3107n04
```

作业提交 - salloc

■ salloc 命令交互式资源申请

✓ 指定分区申请2个节点资源

```
salloc -p <分区> -N <节点数>
```

```
[user4@login01 ~]$ salloc -p normal -N 2
salloc: Pending job allocation 4441668
salloc: job 4441668 queued and waiting for resources
salloc: job 4441668 has been allocated resources
salloc: Granted job allocation 4441668
salloc: Waiting for resource configuration
salloc: Nodes a3107n[04-05] are ready for job
[user4@login01 ~]$ ssh a3107n04
[user4@a3107n04 ~]$ module purge
[user4@a3107n04 ~]$ module load mpi/intelmpi/2022.1.0
[user4@a3107n04 ~]$ mpirun -np 2 hostname
a3107n04
a3107n04
[user4@a3107n04 ~]$ srun -n 2 hostname
srun: ROUTE: split_hostlist: hl=a3107n[04-05] tree_width 0
a3107n05
a3107n04
[user4@a3107n04 ~]$ exit
logout
Connection to a3107n04 closed.
[user4@login01 ~]$ exit
exit
salloc: Relinquishing job allocation 4441668
salloc: Job allocation 4441668 has been revoked.
```

1、salloc请求资源

2、分配资源并生成jobid

3、登录分配节点执行任务

4、任务执行完成退出并回收资源

作业提交 - sbatch

■ sbatch 命令批处理作业提交示例

✓ 提交slurm作业脚本

```
sbatch <脚本路径>
```

```
[user4@login01 ~]$ sbatch test.slurm
Submitted batch job 4441736
```

作业提交后，默认将在当前目录生成输出日志，日志文件名格式为：
slurm-<作业ID>.out

```
[user4@login01 ~]$ cat slurm-4441736.out
a3404n19
```

作业提交 - sbatch脚本示例

■ sbatch 脚本示例

```
#!/bin/bash
#SBATCH -J test
#SBATCH -p high
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --ntasks-per-node=1

module purge
module load mpi/openmpi/gnu/4.0.4

cd $SLURM_SUBMIT_DIR
mpirun -np $SLURM_NTASKS hostname
```

指定作业名
指定分区
申请一个节点的资源
指定一个进程运行
指定每个节点进程数

加载作业运行所需的环境变量

切换到提交目录
执行任务运行语句

作业提交 - 环境变量

- slurm作业可以通过以下变量获取作业信息

Slurm环境变量	功能
$SLURM_JOB_ID	作业ID
$SLURM_JOB_NAME	作业名
$SLURM_JOB_PARTITION	队列的名称
$SLURM_NTASKS	任务（进程）总数
$SLURM_NTASKS_PER_NODE	每个节点请求的任务数
$SLURM_JOB_NUM_NODES	节点数
$SLURM_JOB_NODELIST	节点列表
$SLURM_LOCALID	作业进程的节点本地任务ID
$SLURM_ARRAY_TASK_ID	作业组中的任务ID
$SLURM_SUBMIT_DIR	工作目录
$SLURM_SUBMIT_HOST	提交作业的主机名

作业提交 - 环境变量

```
[user4@login03 ~]$ cat job3.slurm
#!/bin/bash
#SBATCH -J echo_env
#SBATCH -p normal
#SBATCH -N 2
#SBATCH -n 2
#SBATCH --ntasks-per-node=1

echo "Allocate Nodelist is ${SLURM_JOB_NODELIST}"

sleep 10

[user4@login03 ~]$ sbatch job3.slurm
Submitted batch job 4474470
[user4@login03 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4474470    normal echo_env     user4  R       0:01      2 a3107n[16-17]
[user4@login03 ~]$ cat slurm-4474470.out
Allocate Nodelist is a3107n[16-17]
[user4@login03 ~]$
```

作业提交 – PBS vs SLURM

功能	SLURM	PBS
任务名称	#SBATCH -J name	#PBS -N name
指定队列/分区	#SBATCH -p cpu	#PBS -q cpu
指定 QoS	#SBATCH --qos=debug	#PBS --qos=debug，需调度器支持
最长运行时间	#SBATCH -t 5:00	#PBS -l walltime=5:00
指定节点数量	#SBATCH -N 1	#PBS -l nodes=1
指定 CPU 核心	#SBATCH --cpus-per-task=4	#PBS -l ppn=4
指定加速卡	#SBATCH --gres=gpu:1	不支持
作业数组	#SBATCH -a 0-2	#PBS -t 0-2
输出文件	#SBATCH -o test.out	#PBS -o test.out
提交任务脚本	sbatch run.slurm	qsub run.pbs
查看任务状态	squeue	qstat
取消任务	scancel 1234	qdel 1234
交互式任务	salloc，手动切换	qsub -I，自动切换
指定特定节点	#SBATCH --nodelist=comput1	qsub -l nodes=comput1

作业提交 - scancel取消作业

- **scancel 命令用于取消挂起或正在运行的作业或作业步骤。**

指定参数	参数含义
-n, --name=<job_name>	指定作业名
-p, --partition=<partition>	指定分区名
-t, --state=<states>	指定作业状态
-u, --user=<user_name>	指定用户名
-w, --nodelist=<hostlist>	指定节点

作业提交 - scancel取消作业

- **scancel 命令用于取消挂起或正在运行的作业或作业步骤。**
  - ✓ 取消作业

```
scancel <作业ID>
```

```
[user4@login01 ~]$ squeue -j 4445358
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
           4445358    normal     test     user4 PD       0:00      1 (Nodes required
[user4@login01 ~]$ scancel 4445358
[user4@login01 ~]$ squeue -j 4445358
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

作业提交 - 作业重排队

- **处于完成或失败状态的job如何重新排队**

Slurm 支持重新安排处于完成或失败状态的作业。可以使用命令：

```
scontrol requeue <作业ID>
```

然后，该作业将被重新排队，回到 **PENDING** 状态

谢 谢