□□□□

Slurm

□□□□

□□□□□□□ `sinfo`

```
$ sinfo
PARTITION   AVAIL  TIMELIMIT  NODES  STATE NODELIST
e5v3ib         up   infinite     19  alloc f01n[01-04],f02n[04-06,13-14],f05n[05-14]
e5v3k40ib      up   infinite      1   idle x001
e7v4ib         up   infinite      1  alloc r011
6140ib         up   infinite      9  alloc b01n[01-05,08],b02n[03,05-06]
5218*          up   infinite      1  alloc o003
7702ib         up   infinite      2    mix a[001,009]
722080tiib     up   infinite      3   idle g[001-003]
6330ib         up   infinite      3  inval n[025,027-028]
6330ib         up   infinite      3   mix- n[020-022]
6330ib         up   infinite      9  alloc n[016-019,024,026,029-031]
```

- `PARTITION` □□□□□□□□□□□□□□ `*` □□□□□□
- `AVAIL` □□□□□□ `up` □□□□□ `down` □□□□□□ `inact` □□□□□□
- `TIMELIMIT` □□□□□□□□□□□□□□□□
- `NODES` □□□□□□□□□
- `STATE` □□□□□□□
- `NODELIST` □□□□□□□

□□□□□□□ `sinfo -p <partition>`

□□□□□□□□□ `scontrol show partition` □ `scontrol show partition=<name>`

```
$ scontrol show partition e5v3ib
PartitionName=e5v3ib
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=NO QoS=N/A
   DefaultTime=NONE DisableRootJobs=YES ExclusiveUser=NO GraceTime=0
   Hidden=NO MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=0
   MaxCPUsPerNode=UNLIMITED Nodes=f01n[01-04],f02n[04-06,13-14],f05n[05-14]
   PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO
   OverSubscribe=NO OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=456 TotalNodes=19
   SelectTypeParameters=NONE
   DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
```

□□□□□□

- `PartitionName` □□□□□
- `AllowGroups` □□□□□□□□ ALL □□□□□
- `AllowAccounts` □□□□□□
- `AllowQos` □□□□ QoS
- `AllocNodes` □□□□□□□□
- `Default` □□□□□□□□
- `MaxNodes` □□□□□□□ UNLIMITED □□□□□
- `MaxTime` □□□□□□□
- `Nodes` □□□□□□□□
- `State` □□□□□□ UP □□□□
- `TotalCPUs` □□ CPU □□
- `TotalNodes` □□□□□
- `DefMemPerNode` □□□□□□□□
- `MaxMemPerNode` □□□□□□□□

- `OverSubscribe`: 是否允许超额订阅

查看分区汇总: `sinfo -s`

`sinfo -s` 以分区为单位汇总节点状态，输出中的 `NODES(A/I/O/T)` 表示:

- `A`: Allocated, 包含完全分配的 `alloc` 节点以及部分分配的 `mix` 节点。
- `I`: Idle, 空闲节点数
- `O`: Other, 其他状态节点数，指既非完全空闲（ `idle` ），也非完全或部分分配（ `alloc` / `mix` ）的节点，常见状态包括 `drain` 、 `down` 、 `inval` 、 `resv` 、 `comp` 、 `unk` 等。
- `T`: Total, 该分区节点总数

```
$ sinfo -s
PARTITION   AVAIL  TIMELIMIT   NODES(A/I/O/T) NODELIST
6140ib         up   infinite        13/0/3/16 b01n[01-08],b02n[01-08]
5218*          up   infinite          1/0/0/1 o003
6330ib         up   infinite        13/2/1/16 n[016-031]
83a100ib       up   infinite          1/0/0/1 m001
```

查看节点

```
sinfo -N -l                # 查看节点详细信息（含配置和状态）
scontrol show node         # 查看所有节点详细配置
scontrol show node <name> # 查看指定节点详细信息
scontrol -o show node      # 单行显示节点信息
scontrol show node | grep CfgTRES | tr ',' '\n' | awk -F '[:=]' '/gres\/gpu:/ { print $2 }' |
sort -u  # 查看节点GPU配置
sinfo -t idle              # 查看空闲节点
sinfo -t alloc             # 查看已分配节点
sinfo -t mix               # 查看混合状态节点
sinfo -t drain             # 查看已下线节点
scontrol show nodes | grep CfgTRES   # 查看节点可追踪资源配置（含GPU）
```

节点状态说明:

alloc / ALLOCATED：节点已分配给作业，所有资源已用满
idle：节点空闲，可接收新作业
mix / MIXED：节点部分资源已分配，仍有空闲资源可接受新作业
mix- ：节点部分资源已分配，同时被回填调度器规划给更高优先级作业
inval / INVALID：节点注册无效，通常因实际资源与配置不一致（如 Low RealMemory） down ：节点宕机，不可用
drain / DRAINED：节点已排空，现有作业继续运行但不接受新作业
drng / DRAINING：节点正在排空
comp / COMPLETING：作业已完成，节点正在执行收尾工作
fail / FAIL：节点失效
maint / MAINT：节点处于维护保留状态
resv / RESERVED：节点被预留
unk / UNKNOWN：节点状态未知

节点状态后缀：

* ：节点当前无响应
~ ：节点已关闭电源
#：节点正在启动或配置
%：节点正在关闭电源
$：节点处于维护保留
@：节点待重启
^ ：已发出节点重启指令
- ：节点被回填调度器规划用于更高优先级作业

LSF

查看队列

查看所有队列: `bqueues`

```
$ bqueues
QUEUE_NAME      PRIO STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN  SUSP
x7542!           50  Open:Active       -    -    -    -    24     0    24     0
e5645!           50  Open:Active       -    -    -    -     0     0     0     0
e52643tgb!       50  Open:Active       -    -    -    -     8     0     8     0
……
6226rib          30  Open:Active       -    -    -    -     0     0     0     0
5218             30  Open:Active       -    -    -    -     0     0     0     0
6230r            30  Open:Active       -    -    -    -    32     0    32     0
```

- `QUEUE_NAME`: 队列名称
- `PRIO`: 队列优先级，越大优先级越高
- `STATUS`: 队列状态。 Open/Closed 表示是否可以提交，即用户是否可以提交作业到该队列;

  Active/Inact 表示否可以派发，即该队列的作业是否会被分发到计算节点运行。 Open:Active

  表示可提交可派发， Open:Inact 表示可提交但是不派发。
- `NJOBS`: 排队、运行和挂起的作业所占总 CPU核数
- `PEND`: 排队中的作业所需总 CPU核数
- `RUN`: 运行中的作业所占总 CPU核数
- `SUSP`: 挂起的作业所占总 CPU核数

查看队列详细信息: `bqueues -l`

```
$ bqueues -l e5v3ib

QUEUE: e5v3ib
  -- CPU: 2*E5-2680v3, RAM: 256GB/128GB, NET: 56Gb FDR InfiniBand

……

SCHEDULING POLICIES:  FAIRSHARE  EXCLUSIVE
```

```
FAIRSHARE_QUEUES:  e5v3ib e5v3ib! e7v4ib x5650ib 6140ib 62v100ib 722080tiib 72rtxib 7702ib ……
DISPATCH_ORDER:  QUEUE
USER_SHARES:  [root=, 999999] ……

SHARE_INFO_FOR: e5v3ib/
 USER/GROUP   SHARES  PRIORITY  STARTED  RESERVED  CPU_TIME  RUN_TIME   ADJUST  GPU_RUN_TIME
root=       999999  202255.328      0        0      8456.5     1542       0.000             0
……

USERS: all ~test/
HOSTS:  f01+10 f02+10 f03+10 f04+10 f05s+10 f05l/
RES_REQ:  span[ptile=24]
Maximum slot reservation time: 43200 seconds
```

查看节点

```
lshosts查看节点配置和资源

lshosts -gpu查看节点GPU配置和拓扑结构

lsload查看节点当前负载信息

lsload -gpu查看节点GPU整体负载

lsload -gpuload查看节点每个GPU负载

bhosts查看节点状态
```

自动关机

集群会对动力环境进行监控，遇市电中断或温度过高，将会自动终止所有作业，按照安全顺序进行关机操作。

关机时会在 `/fs00/reports/bjobs/` 目录下会自动保存一份作业列表备查。如文件

/fs00/reports/bjobs/bjobs.20130728070457 表明 2013年 07月 28日 07点 04分 57秒时刻所有作业的状态（ `bjobs -uall -w`

的输出），同时也说明这个时间点开始自动关机。

如果温度未触及高点，但已明显增高，为了防止温度继续增高，集群会停止派发新作业，并且关闭空闲节点。

进程监督

用户只能登录到登录节点，且可从登录节点登录到有自己正在运行作业的计算节点。

所有节点的用户进程都会被检查，如果发现如下情况，将会杀掉该节点此用户所有进程，并记录在

/fs00/reports/process 中

计算节点有进程未通过作业调度系统提交运行的

计算节点用户进程占用的 CPU资源明显高于在作业调度系统中请求的 CPU资源

登录节点用户占用大于 2个CPU核心的

Revision #13
Created 2021-05-09 22:37:22 CST by Yao Ge
Updated 2026-10-09 19:50:31 CST by Jiang Meng