中科曙光 Sugon

SLURM调度系统-命令使用

HPC产品事业部

2022-05-23

携手成就梦想

目录

作业查询常用命令

squeue	查看作业和作业步的状态。
scontrol	查看或更新各种对象（如集群、分区、作业、作业步、预约等）的状态。
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。
sprio	（启用multifactor插件时）查询排队作业的优先级，以及各优先级因子的具体取值。

作业查询常用命令-squeue

■ 作业查询选项squeue:查询排队和运行状态的作业

squeue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

```
[root@gv78 log]# squeue -j 365054 -O JobID,Account,AllocNodes,Cluster
JOBID                 ACCOUNT               ALLOC_NODES           CLUSTER
365054                qq1                   gv78                  cancon
[root@gv78 log]#
```

常用基本命令-squeue

默认命令下squeue显示的字段

```
[root@gv78 ~]# squeue |head
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            226015 kshdexclu mcs_test    qq100 PD       0:00      1 (Priority)
            226014 kshdexclu mcs_test    qq100 PD       0:00      1 (Priority)
            226013 kshdexclu mcs_test    qq100 PD       0:00      1 (Resources)
```

squeue主要输出项及其含义

JOBID	PARTITION	NAME	USER	ST	TIME	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	分配给的节点名列表（原因）

作业查询常用命令scontrol

scontrol可以用来查看状态和配置命令

scontrol show <COMMAND>

COMMAND	解释
job	显示作业信息
node	显示节点信息
partition	显示队列信息
config	显示配置信息

作业查询常用命令scontrol

scontrol show job [-d <**jobid**>]　　#如果不加作业jobid将显示近期所有的作业

```
[root@gv78 log]# scontrol show job 365040
JobId=365040 JobName=mcs_test.slurm
   UserId=qq100(1300) GroupId=qq100(1300) MCS_label=qq100
   Priority=1000 Nice=0 Account=qq100 QOS=normal
   JobState=RUNNING Reason=None Dependency=(null)
   Requeue=0 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:12:30 TimeLimit=10-00:00:00 TimeMin=N/A
   SubmitTime=2022-04-27T13:38:42 EligibleTime=2022-04-27T13:38:42
   AccrueTime=2022-04-27T13:38:42
   StartTime=2022-04-27T13:38:42 EndTime=2022-05-07T13:38:42 Deadline=N/A
   SuspendTime=None SecsPreSuspend=0 LastSchedEval=2022-04-27T13:38:42
   Partition=kshdexclu07 AllocNode:Sid=gv78:11170
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=gv78
   BatchHost=gv78
   NumNodes=1 NumCPUs=1 NumTasks=1 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=1,mem=2850M,node=1,billing=1
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryCPU=2850M MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/home/qq100/mcs_test.slurm
   WorkDir=/home/qq100
   StdErr=/home/qq100/slurm-365040.out
   StdIn=/dev/null
   StdOut=/home/qq100/slurm-365040.out
   Power=
   NtasksPerTRES:0

[root@gv78 log]#
```

作业查询常用命令scontrol

scontrol show job显示各项含义

参数	含义
JobId	作业号
JobName	显示作业名称
UserId	**用户名**
GroupId	**用户组**
MCS_label	标签
Priority	优先级，越大越优先，如果为0则表示被管理员挂起，不允许运行
Nice	Nice值，越小越优先，20到19
Account	**记账用户名**
QOS	**作业的服务质量**
JobState	**作业状态**
Reason	排队原因
Dependency	**依赖关系**
Requeue	**节点失效时，是否重排队，0为否，1为是**
Restarts	失败时，是否重运行，0为否，1为是。

作业查询常用命令scontrol

续

参数	含义
BatchFlag	是否为批处理作业，0为否，1为是。
Reboot	节点空闲时是否重启节点，0为否，1为是。
ExitCode	作业退出代码
RunTime	已运行时间
TimeLimit	**作业允许的剩余运行时间**
TimeMin	作业运行允许最小时间
SubmitTime	**提交时间**
StartTime	**开始运行时间**
EndTime	**预计结束时间**
Deadline	截止时间
PreemptTime	先占时间
SuspendTime	挂起时间
Partition	**队列名**
AllocNode:Sid	分配的节点:系统ID号

作业查询常用命令scontrol

续

参数	含义
ReqNodeList	要求的节点列表
ExcNodeList	排除的节点列表
NodeList	**实际运行节点列表**
NumNodes	**节点数**
NumCPUs	**CPU核数**
NumTasks	**任务数**
CPUs/Task	**CPU核数/任务数**
ReqB:S:C:T:	所需的主板数:每主板CPU颗数:每颗CPU核数:每颗CPU核的线程数,
TRES	显示分配给作业的可被追踪的资源
Socks/Node	每节点CPU颗数
NtasksPerN:B:S:C	每主板数:每主板CPU颗数:每颗CPU的核数:每颗CPU核的线程数启动的作业数
CoreSpec	各节点系统预留的CPU核数，如未包含，则显示*
MinCPUsNode:	每节点最小CPU核数
MinMemoryCPU	每节点最小内存大小，0表示未限制。

作业查询常用命令scontrol续

参数	含义
MinTmpDiskNode	每节点最小临时存盘硬盘大小，0表示未限制。
Features	特性
Gres	通用资源
Reservation	预留资源
OverSubscribe	是否允许与其它作业共享资源，OK允许，NO不允许。
Contiguous	是否要求分配连续节点，OK是，NO否。
Licenses	**申请的license类型**
Network	网络标识
Command	**作业脚本位置**
WorkDir	**工作目录**
StdErr	**标准出错输出文件**
StdIn	**标准输入文件**
StdOut	**标准输出文件**

作业查询常用命令sacct

sacct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：作业号jobid、状态status和退出码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
-i nodename	显示在特定节点数量上运行的作业
-o, --format	指定显示字段以逗号分隔

作业查询常用命令sacct

sacct查询位于gv78上且时间范围为2022-04-27-14:20到2022-04-27-14:27分的作业

```
sacct -N gv78 -S 2022-04-27-14:20:30 -E 2022-04-27-14:27:30
```

```
[qq100@gv78 ~]$ sacct -N gv78 -S 2022-04-27-14:20:30 -E 2022-04-27-14:27:30
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
1006                sleep kshdexclu+        qq1          1    RUNNING      0:0
1006.extern        extern                   qq1          1    RUNNING      0:0
1006.0              sleep                   qq1          1    RUNNING      0:0
365040        mcs_test.+ kshdexclu+      qq100          1    RUNNING      0:0
365040.batch        batch                  qq100          1    RUNNING      0:0
365040.exte+       extern                  qq100          1    RUNNING      0:0
365040.0            sleep                  qq100          1    RUNNING      0:0
365041        mcs_test.+ kshdexclu+      qq100          1    RUNNING      0:0
365041.batch        batch                  qq100          1    RUNNING      0:0
365041.exte+       extern                  qq100          1    RUNNING      0:0
365041.0            sleep                  qq100          1    RUNNING      0:0
[qq100@gv78 ~]$
```

作业查询常用命令sstat

sstat：查看正在运行的作业

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

实时作业状态查询sstat

中科曙光 Sugon

（1）查询完整信息 sstat -j <jobid>

```
(py39) [tangxiao@login01 ~]$
(py39) [tangxiao@login01 ~]$ sstat 1370040
       JobID  MaxVMSize  MaxVMSizeNode  MaxVMSizeTask  AveVMSize     MaxRSS MaxRSSNode MaxRSSTask     AveRSS MaxPages MaxPagesNode   MaxPagesTask   AvePages     MinCPU MinCPUNode MinCPUTask     AveCPU   NTasks AveCPUFreq ReqCPUFreqMin R
eqCPUFreqMax ReqCPUFreqGov ConsumedEnergy  MaxDiskRead MaxDiskReadNode MaxDiskReadTask  AveDiskRead MaxDiskWrite MaxDiskWriteNode MaxDiskWriteTask AveDiskWrite TRESUsageInAve TRESUsageInMax TRESUsageInMaxNode TRESUsageInMaxTask TRESUsag
eInMin TRESUsageInMinNode TRESUsageInMinTask TRESUsageInTot TRESUsageOutAve TRESUsageOutMax TRESUsageOutMaxNode TRESUsageOutMaxTask TRESUsageOutMin TRESUsageOutMinNode TRESUsageOutMinTask TRESUsageOutTot
------------ ---------- -------------- -------------- ---------- ---------- ---------- ---------- ---------- -------- ------------ -------------- ---------- ---------- ---------- ---------- ---------- -------- ---------- ------------- -
------------ ------------- -------------- ------------ --------------- --------------- ------------ ------------ ---------------- ---------------- ------------ -------------- -------------- ------------------ ------------------ --------
------ ------------------ ------------------ -------------- --------------- --------------- ------------------- ------------------- --------------- ------------------- ------------------- ---------------

sstat: ROUTE: split_hostlist: hl=a3105n16 tree_width 0
1370040.bat+    767036K       a3105n16              0    298856K    135760K   a3105n16          0    130456K     4760     a3105n16              0       4760  46:26.000   a3105n16          0  46:26.000        1      533K             0
           0             0              0 285817918482        a3105n16              0 285817918482 264243803716        a3105n16               0 264243803716 cpu=00:46:26,+ cpu=00:46:26,+ cpu=a3105n16,ener+ cpu=00:00:00,fs/d+ cpu=00:4
6:26,+ cpu=a3105n16,ener+ cpu=00:00:00,fs/d+ cpu=00:46:26,+ energy=0,fs/di+ energy=0,fs/di+ energy=a3105n16,fs+         fs/disk=0 energy=0,fs/di+ energy=a3105n16,fs+          fs/disk=0 energy=0,fs/di+
(py39) [tangxiao@login01 ~]$
(py39) [tangxiao@login01 ~]$
```

（2）查询字段范围 sstat -e

```
(py39) [tangxiao@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
TRESUsageInMinNode     TRESUsageInMinTask     TRESUsageInTot         TRESUsageOutAve
TRESUsageOutMax        TRESUsageOutMaxNode    TRESUsageOutMaxTask    TRESUsageOutMin
TRESUsageOutMinNode    TRESUsageOutMinTask    TRESUsageOutTot
(py39) [tangxiao@login01 ~]$
```

作业查询常用命令sstat

（3）查询内存消耗等常见信息

```
(py39) [tangxiao@login01 ~]$
(py39) [tangxiao@login01 ~]$ sstat -j 1370040 -o jobid%16,nodelist,ntasks,averss,maxrss,avevmsize,maxvmsize,pids
           JobID             Nodelist   NTasks     AveRSS     MaxRSS  AveVMSize  MaxVMSize                Pids
---------------- -------------------- -------- ---------- ---------- ---------- ---------- --------------------
sstat: ROUTE: split_hostlist: hl=a3105n16 tree_width 0
   1370040.batch             a3105n16        1    130456K    135760K    298856K    767036K          32156,32167
(py39) [tangxiao@login01 ~]$
```

（4）sstat -j 365041 -a -o JobID,NTasks,Nodelist

```
(py39) [tangxiao@a3105n16 ~]$ top -u tangxiao
top - 08:57:09 up 1 day, 17:11,  1 user,  load average: 0.30, 0.36, 0.38
Tasks: 856 total,   1 running, 855 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.4 sy,  0.0 ni, 99.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 26167425+total, 24578782+free, 11262168 used,  4624264 buff/cache
KiB Swap: 16364540 total, 16364540 free,        0 used. 24700468+avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
32167 tangxiao  20   0  185536 128916    912 S   8.9  0.0  47:09.22 revis_term_c5.e
97319 tangxiao  20   0  185296   4088   2276 R   0.4  0.0   0:00.07 top
94105 tangxiao  20   0  115524   3928   1640 S   0.0  0.0   0:00.18 bash
94103 tangxiao  20   0  168600   3572   1036 S   0.0  0.0   0:00.00 sshd
32156 tangxiao  20   0  113320   1524   1216 S   0.0  0.0   0:00.02 slurm_script
```

队列操作-队列查询

scontrol查询分区配置

scontrol show partition <分区名>

```
[root@gv78 log]# scontrol show partition
PartitionName=kshcexclu16
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=NO QoS=normal
   DefaultTime=10-00:00:00 DisableRootJobs=YES ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=0 LLN=NO MaxCPUsPerNode=UNLIMITED
   Nodes=gv78
   PriorityJobFactor=1 PriorityTier=6000 RootOnly=NO ReqResv=NO OverSubscribe=NO
   OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=8 TotalNodes=1 SelectTypeParameters=NONE
   JobDefaults=(null)
   DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
```

队列操作-队列查询

scontrol show partition显示各项含义

参数	解释
PartitionName	**队列名**
AllowGroups	**允许的用户组**
AllowAccounts	**允许的账户**
AllowQos	**允许的QoS**
AllocNodes	**队列中包含的节点**
Default	**是否为默认队列**
QoS	服务质量
DefaultTime	默认作业可以运行的最长时间
DisableRootJobs	是否禁止root用户提交作业
ExclusiveUser	排除的用户
GraceTime	抢占时间，单位秒
Hidden	是否为隐藏队列
MaxNodes	**单个作业允许申请的最大节点数**
MaxTime	**最大运行时间**

队列操作-队列查询

scontrol show partition显示各项含义

参数	解释
MinNodes	最小节点数
LLN	是否按照最小负载节点调度
MaxCPUsPerNode	每个节点的最大CPU颗数
Nodes	节点名
PriorityJobFactor	作业因子优先级
PriorityTier	调度优先级
RootOnly	是否只允许Root
ReqResv	要求预留的资源
OverSubscribe	**是否允许作业节点间共享**
PreemptMode	是否为抢占模式
State	**分区状态**
TotalCPUs	**总CPU核数**
TotalNodes	**总节点数**
SelectTypeParameters	资源选择类型参数（启用的话会覆盖slurm.conf中的参数）
DefMemPerNode	**每个节点默认分配的内存大小，单位MB**
MaxMemPerNode	每个节点最大内存大小，单位MB

队列操作-队列参数修改

scontrol可以用来查看状态和配置命令

scontrol updat partition <分区中参数>

常用参数	解释
AllowGroups	允许的用户组
AllowAccounts	允许的账户
PriorityJobFactor	作业优先级计算项
AllowQos	允许的QOS
PriorityTier	队列间抢占优先级
OverSubscribe	是否允许节点资源在作业间共享
State	分区状态

队列操作

例：修改kshdexclu07 分区的OverSubscribe为YES

```
scontrol updat partition=kshdexclu07 OverSubscribe=yes
```

```
PartitionName=kshdexclu07
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=NO QoS=normal
   DefaultTime=10-00:00:00 DisableRootJobs=YES ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=0 LLN=NO MaxCPUsPerNode=UNLIMITED
   Nodes=gv78
   PriorityJobFactor=2 PriorityTier=300 RootOnly=YES ReqResv=NO OverSubscribe=YES:4
   OverTimeLimit=24 PreemptMode=OFF
   State=UP TotalCPUs=8 TotalNodes=1 SelectTypeParameters=NONE
   JobDefaults=(null)
   DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED

[root@gv78 log]#
```

使用scontrol write config可以将修改的地方保存到一个新的文件中，根据此文件对conf文件修改替换

```
[root@gv78 log]# scontrol write config
Slurm config saved to /opt/gridview/slurm/etc/slurm.conf.2022-04-27T15:36:56
```

资源查询

在普通用户下，查看分区、节点、以及license的可用性

功能	命令
sinfo	（普通用户下）查看当前用户可用分区
scontrol show node	查看节点资源的可用性
scontrol show license	查看可用license

资源查询

或者查询账户下的资源限制，修改账户部分资源的命令如下：

修改账户资源

```
sacctmgr -i modify account 账户名 set GrpSubmit=1000
sacctmgr -i modify account 账户名 set GrpJobs=1000
sacctmgr -i modify account 账户名 set GrpCPUS=2
sacctmgr -i modify account 账户名 set MaxNodes=2
sacctmgr -i modify account 账户名 set GrpTRES=cpu=2,mem=200,gres/gpu=2
GrpSubmit（最大提交作业数）
GrpJobs（最大运行作业数）
GrpTRES（最大CPU核数、最大DCU卡数、最大GPU卡数、最大节点数）
```

设置完资源限制后可以使用使用sacctmgr show account <账户名> withassoc来查询

```
[qq1@gv78 ~]$ sacctmgr show account qq1 withassoc
   Account                Descr                  Org    Cluster   Par Name       User     Share   Priority GrpJobs GrpNodes  GrpCPUs  GrpMem GrpSubmit     GrpWall  GrpCPUMins MaxJobs MaxNodes  MaxCPUs MaxSubmit     MaxWall
 MaxCPUMins                   QOS   Def QOS
---------- -------------------- -------------------- ---------- ---------- ---------- --------- ---------- ------- -------- -------- ------- --------- ----------- ----------- ------- -------- -------- --------- -----------
----------- -------------------- ---------
       qq1                  qq1                  qq1     cancon       root                 1             1000               2     200      1000                                                2
                           normal
       qq1                  qq1                  qq1     cancon               qq1_1        1                                                                                              2
                           normal
       qq1                  qq1                  qq1     cancon                 qq1        1                                                                                              2
                           normal
[qq1@gv78 ~]$
```

目录

作业提交

sbatch	srun	salloc
批处理作业提交	交互式作业提交	交互式资源申请

作业提交

常用的作业提交参数，适用于上述两种作业提交方式。

参数	参数解释
-J 或者 --job-name	指定作业名称
-p或者 --partition	指定队列资源
-N或者 –nodes=<number>	指定节点数量
-n或者 --ntasks = <number>	指定处理器数量
-o或者--output=<filename pattern>	指定stdout的输出文件。如果指定的文件已经存在它将被覆盖。
-e或者 --error=<filename pattern>	指定stderr的输出文件。如果指定的文件已经存在，它将被覆盖。

作业提交-SBATCH

一些常见的资源需求参数（在脚本文件中使用 #SBATCH -XX XXX的方式写入脚本）

指定参数	参数含义
--mem=10G	指定每个节点上使用的物理内存
--reservation	使用创建的资源预留
--begin	指定作业开始时间
-D，--chdir:	指定脚本/命令的工作目录
-c, --cpu-per-task=NCPUs	指定每个进程（task）使用核数，不指定默认为1
-n，--ntask=NTASKs	指定总进程数；不使用cpus-per-task，可理解为进程数即为核数
--ntask-per-node=N	指定每个节点进程数/核数，使用-n参数后变为每个节点最多运行的进程数
-t, --time=dd-hh:mm:ss	作业最大运行时间
-w, --nodelist=node[1,2]	指定优先使用节点，不可与避免节点冲突
-x，--exclude=node[3,5-6]	指定避免使用节点，不可与优先节点冲突
--mem-per-cpu=2048MB	指定计算cpu最大占用内存大小

作业提交-SBATCH

简化的MPI/OpenMP作业脚本，示例如右图所示：

示例中，calc是一个单纯的OpenMP程序。

提交作业，查看作业信息。（使用export OMP_NUM_THREADS=1 可以设置每个进程只开启一个线程）

```
#!/bin/bash
#SBATCH -J openmp_mpi   #任务名
#SBATCH -p kshdexclu07   #提交到 kshdexclu07 分区
#SBATCH -N 1                #申请 1 个节点
#SBATCH -n 2                #申请2个进程（task）
#SBATCH --cpus-per-task=2#每个进程分2个CPU
#SBATCH -o   %j.loop
#SBATCH -e   %j.log
#SBATCH --comment=WRF

module load compiler/devtoolset/7.3.1
module load mpi/openmpi/4.0.2/gcc-7.3.1

mpirun ./calc  1000000
```

作业提交-SBATCH

简化MPI程序示例：

示例中，基于srun启动mpi程序。

```
#!/bin/bash
#SBATCH -o %j
#SBATCH -J MPI
#SBATCH -t 00:10:00
#SBATCH -p hpc
#SBATCH --mem-per-cpu=3G
#SBATCH --tasks-per-node=32
#SBATCH -N 2

module load compiler/devtoolset/7.3.1
module load compiler/rocm/2.9
module load mpi/hpcx/2.4.1/gcc-7.3.1

#  4x32, about   60 sec
 export LOOPMAX=1000000

CORELOOP=$(expr $LOOPMAX / 128)
echo "CORELOOP=$CORELOOP"
export LOOPMAX=$(expr $SLURM_NTASKS \* $CORELOOP )
echo "LOOPMAX=$LOOPMAX"

export MPITYPE=pmix_v3
echo "use srun, loop=$LOOPMAX" && time srun --mpi=$MPITYPE ./open_fire_v5 $LOOPMAX
```

作业提交-SRUN

使用srun --mpi=list查看当前系统所支持的pmi

```
[qyh@gv78 ~]$ srun --mpi=list
srun: MPI types are...
srun: cray_shasta
srun: none
srun: pmi2
srun: pmix
srun: pmix_v3
```

注意：openmpi可以使用pmix_v3,pmix，pmi2；intelmpi只能使用pmi2,因为intel没有集成pmix

```
srun --mpi=pmix_v3 <执行程序>
```

目录

主配置文件slurm.conf

配置类型	文件名称	主要内容	其它说明
主配置文件	slurm.conf	包含主要的调度配置参数，包括调度策略、运行配置、日志配置、记账采集、权限控制、容错配置、认证方式、作业前后处理等等。	必选
记账存储服务配置文件	slurmdbd.conf	包含slurmdbd使用的配置参数，包括认证方式、权限控制、运行配置、日志配置和数据库访问配置。	必选
节点配置文件	slurm_node.conf	包含所有计算节点的配置参数，如节点名、CPU核数、内存、默认状态等。	可选，可合并到slurm.conf
分区配置文件	slurm_partition.conf	包含所有分区的配置参数，如分区名、节点列表、优先级、合法账号、默认时间、默认内存、默认状态等。	可选，可合并到slurm.conf

主配置文件slurm.conf

ClusterName=cluster_gvm03 #集群名称
ControlMachine=gvm03 #主用节点
BackupController=gvm04 #备用节点
AuthType=auth/munge #内部认证
CryptoType=crypto/munge #加密方式
MaxJobCount=200000 #最大作业运行数20万
JobSubmitPlugins=lua #提交参数过滤
ProctrackType=proctrack/linuxproc #进程跟踪插件
ReturnToService=1 #禁用自动恢复
SlurmctldPort=6817 # 主控服务端口
SlurmdPort=6818 #计算代理端口
SlurmUser=root #slurmctld运行用户
SlurmdSpoolDir=/opt/gridview/slurm17/slurmd_spool/ # 计算代理缓存
StateSaveLocation=/opt/gridview/slurm17/spool # slurmctld本地文件缓存

TaskPlugin=task/affinity # cpu亲和性插件
MinJobAge=300 #完成作业保留时间
SlurmctldTimeout=30 #主备切换时间
SlurmdTimeout=300 #计算代理响应时间
FastSchedule=1 #快速调度作业
SchedulerType=sched/backfill #启用回填
SchedulerPort=7321 #调度器端口
SelectType=select/cons_res #资源选择算法
SelectTypeParameters=CR_Core_Memory #基于Core和内存调度
SchedulerParameters=defer,sched_min_interval=10,sched_interval=30,default_queue_depth=100,bf_max_job_test=100,bf_interval=30 # https://slurm.schedmd.com/slurm.conf.html . https://slurm.schedmd.com/sched_config.html
AccountingStorageTRES=cpu,mem #TRES指标配置

主配置文件slurm.conf

```
PriorityType=priority/multifactor        #优先级策略
PriorityDecayHalfLife=30          #半衰期时长
PriorityCalcPeriod=5              #FS统计间隔
PriorityWeightFairshare=100       #FS权重
PriorityWeightPartition=1000      #分区权重
ClusterName=pia                   #集群名
PreemptMode=requeue,gang                #抢占策略
PreemptType=preempt/partition_prio    #队列优先级
DebugFlags=NO_CONF_HASH          # 调试标识
PrivateData=jobs     # 权限控制
HealthCheckInterval=60                  #检查间隔
HealthCheckProgram=/usr/sbin/nhc      #检查工具
AccountingStorageEnforce=associations,limits  #组织关联和资源限制
AccountingStorageHost=gvm04    #主用记账服务
AccountingStorageBackupHost=gvm03 #备用记账服务
AccountingStoragePort=7031          #记账服务端口
```

```
AccountingStorageType=accounting_storage/slurmdbd
#启用slurmdbd
AccountingStorageUser=root          #记账服务
AccountingStoreJobComment=YES   #记录作业注释
JobCompType=jobcomp/none    #禁止生成comp日志
JobAcctGatherFrequency=300    #作业采集间隔
JobAcctGatherType=jobacct_gather/linux    #启用Linux插件
SlurmctldDebug=3                        #日志级别
SlurmctldLogFile=/opt/gridview/slurm17/log/slurmctld.log
JobRequeue=1                  # 允许重新排队
SlurmdDebug=3                           #日志级别
SlurmdLogFile=/opt/gridview/slurm17/log/slurmd_%h.log
SuspendTime=1800                        #
include slurm_node.conf                 #引入节点配置
include slurm_partition.conf            #引入分区配置
```

记账存储服务配置slurmdbd.conf

```
AuthType=auth/munge                                    # 内部认证类型
DbdHost=gvm04                                          # slurmdbd服务节点
DbdBackupHost =gvm03                                   # 备用服务节点
DbdPort=7031                                           # 记账存储服务监控端口
SlurmUser=root                                         # 运行用户
DebugLevel=3                                           # 日志级别
PrivateData=accounts,events,jobs,reservations,usage,users    # 权限控制
LogFile=/opt/gridview/slurm17/log/slurmdbd.log         # 日志路径
StorageType=accounting_storage/mysql                   # 启用mysql
StorageHost=gvm05                                      # 数据库主机
StorageBackupHost=gvm06                                # 数据库备机
StoragePort=3308                                       # 数据库端口
StoragePass=root                                       # 密码
StorageUser=root                                       # 用户名
StorageLoc=gv_slurm_db                                 # 数据库示例
```

节点配置slurm_node.conf

```
NodeName=cmac[0011-0260] NodeAddr=cmac[0011-0260] CPUs=32 Boards=1 SocketsPerBoard=2 CoresPerSocket=16 ThreadsPerCore=1 RealMemory=385437 State=UNKNOWN
NodeName=cmac[0261-1538] NodeAddr=cmac[0261-1538] CPUs=32 Boards=1 SocketsPerBoard=2 CoresPerSocket=16 ThreadsPerCore=1 RealMemory=191913 State=UNKNOWN
```

可选参数:
MemSpecLimit： 保留内存的大小
Weight：节点权重，用于节点选择
Gres： 通用资源(如GPU)，如 GRES=gpus:2
Reason： 节点状态异常（down、drain、fail等）时的原因。
State： 可选的状态包括DOWN 、FAIL、FAILING 、UNKNOWN 、BUSY、IDLE、CLOUD、FUTURE。不要直接配置成BUSY（报错）和IDLE，而应该配置为UNKNOWN（默认）。

注意事项:
1. 节点配置的变更需要同时重启slurmctld和slurmd服务

分区配置slurm_partition.conf

```
PartitionName=serial    Nodes=cmac[0011-0034] Priority=1000  OverSubscribe=FORCE:1 Default=NO  AllowAccounts=ALL DefaultTime=15-00:00:00 MaxTime=INFINITE DefMemPerCPU=10240 LLN=YES State=UP
PartitionName=serial_op    Nodes=cmac[0011-0034] Priority=1000  OverSubscribe=FORCE:1 Default=NO  AllowAccounts=nwp,nwp_op,nwp_sp,lijuan,nwp_pd,nwp_qu DefaultTime=15-00:00:00 MaxTime=INFINITE
 DefMemPerCPU=10240 LLN=YES State=UP
PartitionName=largemem    Nodes=cmac[0035-0260] Priority=1000  OverSubscribe=FORCE:1 Default=NO  AllowAccounts=ALL QOS=normal_qos DefaultTime=15-00:00:00 MaxTime=INFINITE DefMemPerCPU=10240 S
tate=UP
PartitionName=normal    Nodes=cmac[0035-1538] Priority=1000  OverSubscribe=FORCE:1 Default=NO  AllowAccounts=ALL QOS=normal_qos DefaultTime=15-00:00:00 MaxTime=INFINITE DefMemPerCPU=5120 Stat
e=UP
PartitionName=operation Nodes=cmac[0035-1538] Priority=2000  OverSubscribe=FORCE:1 Default=NO  AllowAccounts=nwp,nwp_op,nwp_sp,lijuan,nwp_pd,nwp_qu,nwpbj_ex DefaultTime=15-00:00:00 MaxTime=IN
FINITE State=UP
```

参数简介:

OverSubscribe:

EXCLUSIVE:独占节点，要求启用了Select/cons_res

FORCE[:X]:强制节点（在X作业间）共享，忽略用户自身请求

YES:允许作业共享，考虑用户--oversubscribe 请求。

PreemptMode:

队列级的抢占模式，覆盖全局配置

State:

UP（正常） DOWN(接收不调度) DRAIN（调度不接收） INACTIVE（DOWN+DRAIN）

通用资源配置gres.conf

- 支持通用资源，必须在slurm.conf配置文件中明确指定要管理哪些资源。

参数	解释
GresTypes	e.g. *GresTypes=gpu,mic*
Gres	e.g. Gres=gpu:tesla:2,gpu:kepler:2

日志管理概述

中科曙光 Sugon

日志文件范围

主控服务slurmctld　　记账存储服务slurmdbd
计算代理服务slurmd　　认证服务munge

日志级别参数

配置参数:

Slurm.conf:　　SlurmctldDebug　SlurmdbdDebug
slurmdbd.conf:　　DebugLevel

日志级别:

调度系统支持的日志级别包括:

quiet	fatal	error	info	verbose	debug	debug 2	debug 3	debug 4	debug 5
0	1	2	**3**	4	5	6	7	8	9

日志转储设置

通过操作系统的logrotate工具管理实现自动的日志滚动保存。

日志文件说明

日志文件:

- **主进程日志slurmctld.log**

  存在于调度系统管理节点的/opt/gridview/slurm17/log目录，记录主进程运行日志，涉及作业提交、作业调度、作业控制、状态监控等各个方面的正常和异常信息。

- **记账存储服务日志slurmdbd.log**

  存在于调度系统管理节点的/opt/gridview/slurm17/log目录，主要记录跟数据库相关的各种操作日志。

- **计算代理日志slurmd_{hostname}.log**

  存在于调度系统计算节点的/opt/gridview/slurm17/log目录，记录计算节点服务的运行日志。

- **认证服务日志munged.log**

  存在于每一个调度相关节点的/opt/gridview/munge/log/munge/munged.log目录，主要记录调度系统各组件通信过程中产生/销毁各种凭证（credential）的日志。

日志转储设置

转储配置：

通过三个logrotate配置文件分别实现slurmctld/slurmdbd、slurmd、munge服务的日志文件转储。

日志转储配置	节点分布	对应服务
/etc/logrotate.d/slurm	管理节点	主控服务slurmctld 记账存储服务slurmdbd
/etc/logrotate.d/slurmd	计算节点	计算代理slurmd
/etc/logrotate.d/munge	所有节点	认证服务munged

手工测试：

```
logrotate  -f   /etc/logrotate.d/slurmd
```

日志转储设置-主服务转储配置

```
/opt/gridview/slurm17/log/slurmdbd.log
/opt/gridview/slurm17/log/slurmctld.log
{
        compress                # 启用gzip压缩
        missingok          # 日志不存在不报错退出
        nocopytruncate        # 转储不清空
        nodelaycompress      # 转储时立即压缩
        nomail                 # 禁用邮件通知
        notifempty             # 为空时不转储
        noolddir                # 原目录保存
        rotate 3                # 保存3个转储文件
        sharedscripts          # 多个文件同时处理
        daily                   # 转储周期
        dateext                # 转储后缀为年月日
        size 200M              # 条件大于200M
```

```
        # 文件转储后的操作
        postrotate
                for daemon in $(scontrol show daemons)
                do
                        killall -SIGUSR2 $daemon
                done
                ps -fe|grep slurmdbd |grep -v grep
                if [ $? -ne 0 ]
                then
                echo "no slurmdbd process"
                else
                killall -SIGUSR2 slurmdbd
                fi
        endscript
}
```

日志转储设置-计算代理转储配置

```
/opt/gridview/slurm17/log/slurmd_*.log
{
        compress                # 启用gzip压缩
        missingok           # 日志不存在不报错退出
        nocopytruncate      # 转储不清空
        nodelaycompress     # 转储时立即压缩
        nomail              # 禁用邮件通知
        notifempty          # 为空时不转储
        noolddir            # 原目录保存
        rotate 3            # 保存3个转储文件
        sharedscripts       # 多个文件同时处理
        daily               # 转储周期
        dateext             # 转储后缀为年月日
        size 50M            # 条件大于50M
        （待续）
```

```
        （续）
        # 文件转储后的操作
        postrotate
                for daemon in $(scontrol show daemons)
                do
                        killall -SIGUSR2 $daemon
                done
        endscript


}
```

日志转储设置-认证服务转储配置

```
/opt/gridview/munge/log/munge/*.log
{
        compress                # 启用gzip压缩
        missingok           # 日志不存在不报错退出
        nocopytruncate      # 转储不清空
        nodelaycompress     # 转储时立即压缩
        nomail              # 禁用邮件通知
        notifempty          # 为空时不转储
        noolddir            # 原目录保存
        rotate 3            # 保存3个转储文件
        sharedscripts       # 多个文件同时处理
        daily               # 转储周期
        dateext             # 转储后缀为年月日
        size 50M            # 条件大于50M
        # 文件转储后的操作
        postrotate
                ps -fe|grep munged |grep -v grep
                if [ $? -ne 0 ]
                then
                        echo "no munged process"
                else
                        killall -SIGUSR2 munged
                fi
        endscript

}
```

中科曙光
Sugon

谢谢！

IT基础设施及方案的领导者

数据中国百城百行的发起者

中科院产业化联盟的推动者

安全可控信息系统的践行者

携手成就梦想