SLURM调度系统管理和使用介绍

HPC产品事业部

吕灼恒、张涛、郝文静

目录

调度系统概述-基本概念

■ 资源（Resource）

- ✓ 作业运行过程中使用的可量化实体都是资源；
- ✓ 包括硬件资源（节点、内存、CPU 、GPU等）和软件资源（ License ）；

■ 集群（Cluster）

- ✓ 包含计算、存储、网络等各种资源实体且彼此联系的资源集合；
- ✓ 在物理上，一般由计算处理、互联通信、I/O 存储、操作系统、编译器、运行环境、开发工具等多个软硬件子系统组成；
- ✓ 节点是集群的基本组成单位，从角色上一般可以划分为管理节点、登陆节点、计算节点、存储节点等。

■ 作业（Job）

- ✓ 物理构成，一组关联的资源分配请求，以及一组关联的处理过程；
- ✓ 交互方式，可以分为交互式作业和非交互式作业；
- ✓ 资源使用，可以分为串行作业和并行作业；

调度系统概述-基本概念（续）

- **分区（Partition）**
  - ✓ 带名称的作业容器；
  - ✓ 用户访问控制；
  - ✓ 资源使用限制；
- **作业调度系统（Job Schedule System）**
  - ✓ 负责监控和管理集群中资源和作业的软件系统；
  - ✓ 通常由资源管理器、调度器、任务执行器，以及用户命令和API组成；

调度系统概述-主要作用

- **单一系统映像**
  - ✓ 解决集群结构松散问题；
  - ✓ 统一用户接口，使用简化；
- **系统资源整合**
  - ✓ 管理异构资源和异构系统；
- **多任务管理**
  - ✓ 统一管理任务，避免冲突；
- **资源访问控制**
  - ✓ 基于策略的资源访问控制；

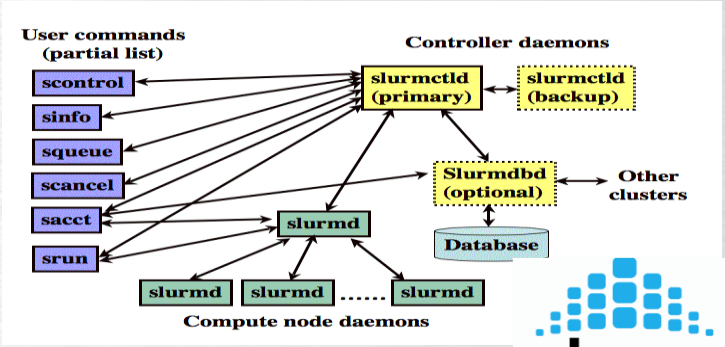
调度系统是面向集群的操作系统。

调度系统概述-功能特性

中科曙光 Sugon

IBM Blue Gene/Q	Cray XT
天河超级计算机	硅立方
IBM_Sequoia	……

slurm workload manager Version 17.02



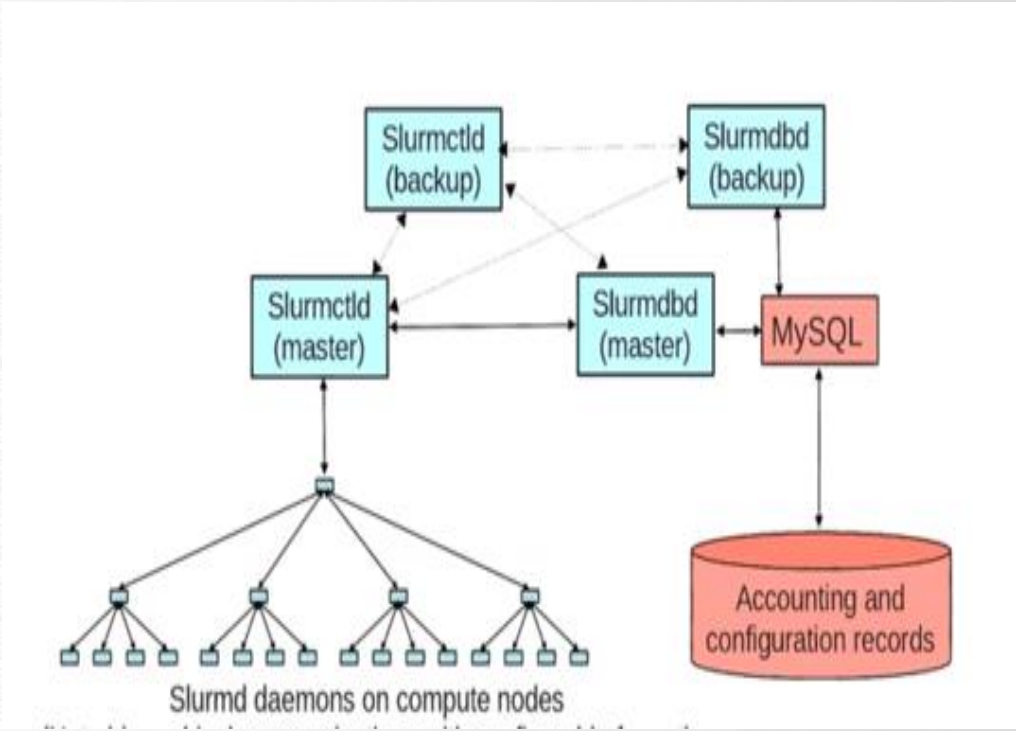
slurm workload manager Version 17.02

SLURM调度核心 -2016年6月Top500前十名有5台

- 高性能；
- 灵活性（众多插件）；
- 扩展性很高，支持数百万处理器核心的调度；
- 节点容错，高稳定性；
- 安全性高、易移植性（不修改内核）好；
- 支持各种常用的HPC操作系统(AIX、Linux、Solaris)；
- MPI支持较好，作业抢占、进程/线程绑定、作业依赖等轻松支持；
- 网络拓扑调度，内置支持Tree、3D-Tours等多种算法。

调度系统概述-运行架构

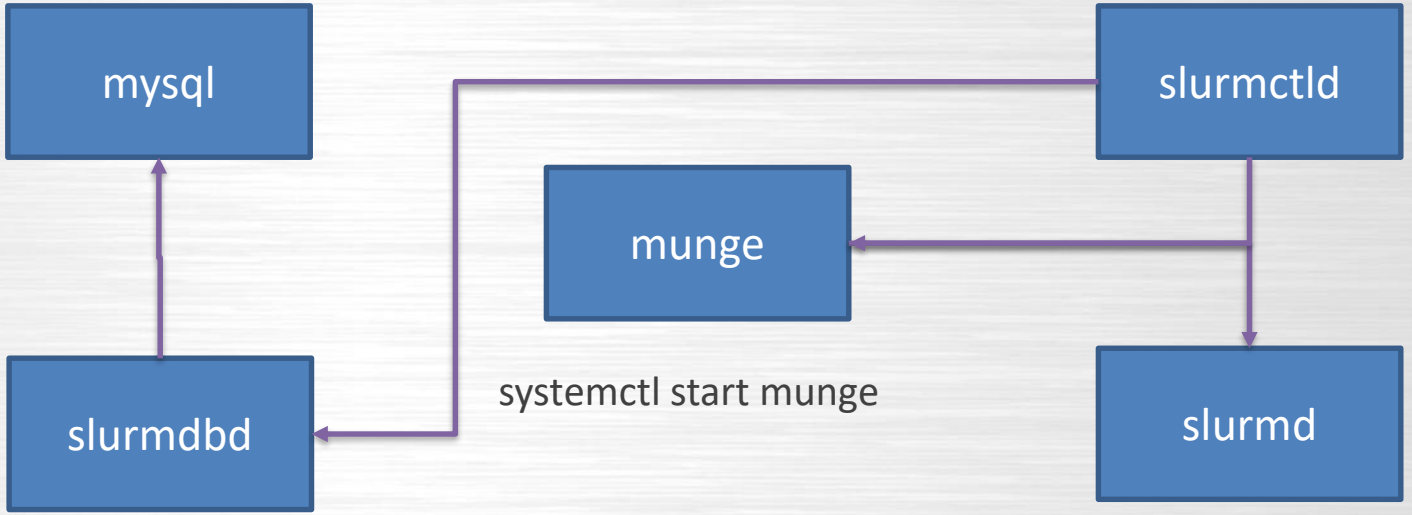
- **主控服务slurmctld**
  故障切换 资源监控
  队列管理 作业调度
- **记账存储服务slurmdbd**
  记账数据 配置信息
  故障切换
- **数据库MySQL**
  记账和配置信息存储
- **计算代理slurmd**
  启动任务 监控任务
  分层通信
- **认证服务munge**
  内部通信认证



调度系统概述-服务启动

/etc/init.d/my_mysqld start

systemctl start slurmctld



systemctl start munge

systemctl start slurmdbd

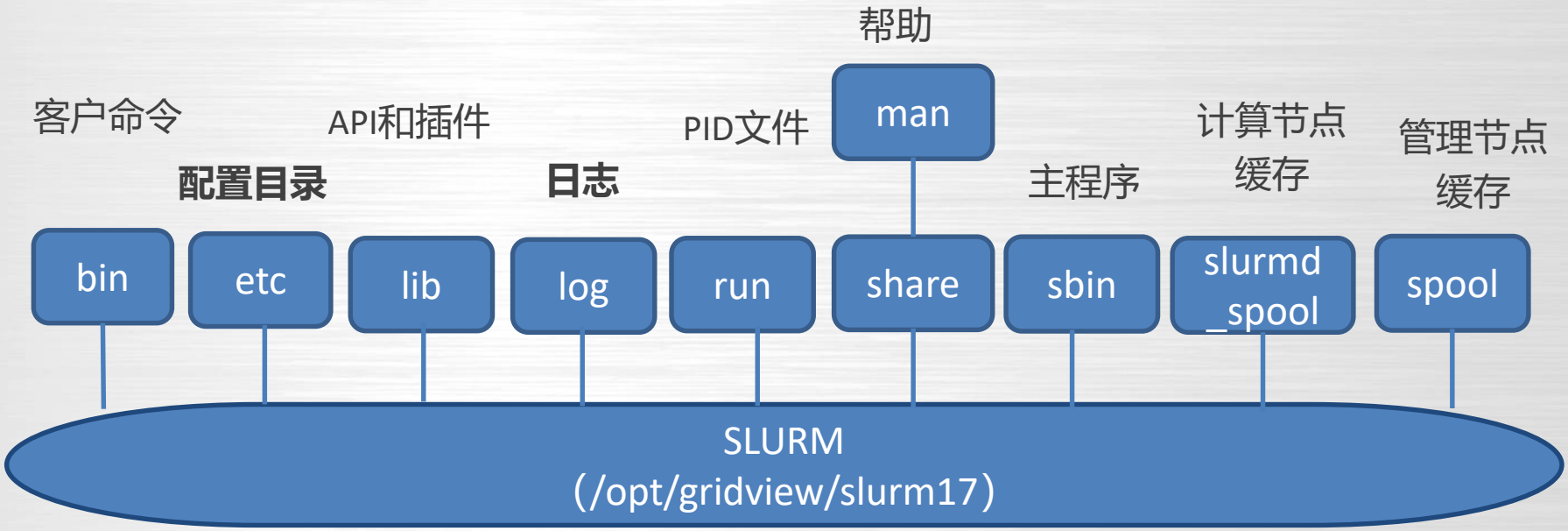
systemctl start slurmd

目录

安装部署-角色与服务

角色类型	节点范围	软件路径	服务名称
登陆节点	gv0001~gv0010,共计10个节点	1. 调度软件安装目录/opt/gridview，包括munge、slurm17	munge.service
管理节点	gvm03（主用）gvm04（备用）	1. 调度软件安装目录/opt/gridview，包括munge、slurm17 2. 数据库软件/gvdata/mysql-5.6.38-linux-glibc2.12-x86_64	munge.service slurmctld.service slurmdbd.service rsyncd和同步脚本
计算节点	gvm0011~gvm1538,共1528个节点	1. 调度软件安装目录/opt/gridview，包括munge、slurm1	munge.service slurmd.service
MySQL节点	gvm05（主用）gvm06（备用）	1. 调度软件主目录/opt/gridview，包括munge、slurm17 2. 数据库软件/gvdata/mysql-5.6.38-linux-glibc2.12-x86_64	my_mysqld

安装部署-软件目录结构



说明：

1. **所有节点的配置目录etc需要借助共享存储全局共享（/g1/gv_share/pia_etc）**
2. **主备节点之间需要同步spool目录实现高可用（/opt/gridview/slurm17/spool）**

安装部署-配置文件概述

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
拓扑配置文件	topology.conf	包含所有节点与交换机以及交换机之间的层次网络链接关系。	可选，由参数TopologyPlugin=topology/tree控制
通用资源配置文件	gres.conf	定义计算节点包含的GRES资源，如名称、类型、设备等。	可选，由参数GresTypes和Gres控制

安装部署-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             #最大作业数20万
JobSubmitPlugins=lua           #提交参数过滤
KillOnBadExit=1                    #异常作业清理
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=slurmadm                 #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
TopologyPlugin=topology/tree     #拓扑调度
```

```
TaskPlugin=task/affinity          # 任务启动cpuset
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=batch_sched_delay=3,defer,sche
d_min_interval=10,sched_interval=30,default_queue_
depth=100,bf_max_job_test=100,bf_interval=30   # 调
度参数
AccountingStorageTRES=cpu,mem  #TRES指标配置
```

安装部署-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=accounts,events,jobs,reservations,usage,us
ers      # 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
```

安装部署-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=slurmadm                                           # 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
```

安装部署-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
Feature： 节点特征，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

安装部署-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，即使启用了elect/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

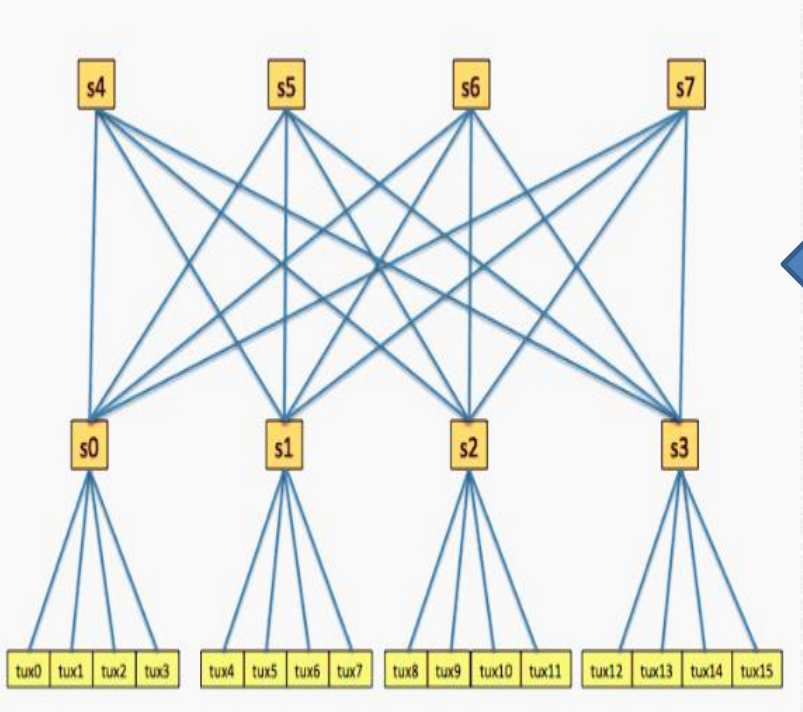
PriorityJobFactor:

用于multifactor Priority。

State:

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

安装部署-拓扑配置示例




```
# topology.conf
# Switch Configuration
SwitchName=s0 Nodes=tux[0-3]
SwitchName=s1 Nodes=tux[4-7]
SwitchName=s2 Nodes=tux[8-11]
SwitchName=s3 Nodes=tux[12-15]
SwitchName=s4 Switches=s[0-3]
```

配置格式：

（1）交换机-节点

SwitchName=X Nodes=<node name>

（2）交换机-交换机

SwitchName=X Switches=<switch name>

安装部署-拓扑配置topology.conf

```
……
SwitchName=ibsw97
Nodes=gv0942,gv0934,gv0941,gv0933,gv0940,gv0932,gv0939,gv0931,gv0946,gv0938,gv0945,gv0937,gv0944,gv0936,gv0943,gv0935,gv0891,gv0892
SwitchName=ibsw98
Nodes=gv0926,gv0918,gv0925,gv0917,gv0924,gv0916,gv0923,gv0915,gv0930,gv0922,gv0929,gv0921,gv0928,gv0920,gv0927,gv0919,gv0893,gv0894
SwitchName=ibsw99
Nodes=gv0910,gv0902,gv0909,gv0901,gv0908,gv0900,gv0907,gv0899,gv0914,gv0906,gv0913,gv0905,gv0912,gv0904,gv0911,gv0903,gv0883,gv0884
SwitchName=ibsw102 Switches=ibsw3,ibsw2,ibsw1
SwitchName=ibsw103 Switches=ibsw4,ibsw6,ibsw5
SwitchName=ibsw104 Switches=ibsw9,ibsw7,ibsw8
SwitchName=ibsw105 Switches=ibsw12,ibsw11,ibsw10
.....
```

安装部署-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

安装部署-通用资源配置gres.conf

参数	解释
Name	通用资源的名称（必须与slurm.conf中的GresTypes值匹配）
Count	此节点上可用的此类型资源数。默认值为1
CPUs	指定可以使用该资源的CPU index number
File	e.g. *File=/dev/nvidia[0-3]*
Type	指定设备类型。

安装部署-日志管理概述

- **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	debug2	debug3	debug4	debug5
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。

安装部署-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。

安装部署-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
```

安装部署-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
}
```

安装部署-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
```

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


}
```

安装部署-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

}
```

目录

命令概述(1/3)

命令分类	命令	功能介绍
作业提交和控制	sbatch	提交脚本排队执行，批处理模式。
	salloc	创建资源申请并启动shell用于运行作业，交互模式
	srun	创建资源申请并启动作业步（通常是MPI作业）。
	sattach	连接正在运行的作业步的标准输出和错误等。
	scancel	取消作业或作业步。
系统状态	sinfo	查看节点和分区的状态。
	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。

命令概述(2/3)

命令分类	命令	功能介绍
记账统计	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。
	sreport	从集群、分区、用户、账号等角度统计资源使用情况。
调度配置	sacctmgr	调度数据库配置管理工具，包括集群、用户、账号等增删，以及资源限制策略的配置。
	sprio	查看作业优先级的具体构成因素。
	sshare	查看各账号的公平共享相关数据。
	sdiag	显示调度模块的操作统计信息，包括运行周期、作业状态统计、RPC类型统计等等。

命令概述(3/3)

命令分类	令	功能介绍
其它命令	sbcast	在作业中传输特定的文件到分配的计算节点中。
	strigger	事件触发器管理工具。
	smap	图形化的查看作业、节点等状态信息。
	sview	图形化的查看系统（作业、分区、预约等）的状态，修改系统配置。

查询信息-查询命令sinfo

sinfo命令参数（1/3）	
-a, --all	查看所有分区信息（含隐藏分区），对应环境变量SINFO_ALL。
-d, --dead	查看dead状态（通信异常）的节点和分区的信息，与-r参数对应。
-h, --noheader	打印分区（或节点）信息时不打印表头。
-i <seconds>, --iterate=<seconds>	定时打印状态信息。
-l, --long	打印分区（或节点）的详细信息。
-n <nodes>, --nodes=<nodes>	查看指定节点的信息。
-N, --Node	以面向节点（默认为分区）的格式打印信息，每行一个节点。
-p <partition>, --partition=<partition>	查看指定分区的状态，对应环境变量SINFO_PARTITION。

查询信息-查询命令sinfo

sinfo命令参数介绍（2/3）	
-r, --responding	查看计算节点（内部通信）正常的节点和分区的状态，与-d参数对应。
-R, --list-reasons	查看节点不可用的原因，包括管理操作设置的异常。
-s, --summarize	查看分区中节点状态的摘要信息。
-S <sort_list>, --sort=<sort_list>	设置打印状态信息时排序的规则，如sinfo -S “#P，-t”。参数可以通过设置环境变量SINFO_SORT实现。
-t <states> , --states=<states>	查询指定节点状态的分区或节点的信息，常见状态包括：**ALLOC/ALLOCTED,COMP/COMPLETING,DOWN,DRAIN/DRAINED/DRAINING,ERR/ERROR,FAIL,IDLE,MIX/MIXED,UNK/UNKNOWN。**
--federation	显示所有集群的分区（或节点）的信息，对应变量SINFO_FEDERATION。
--local	仅显示当前集群的分区（或节点）的信息，对应环境变量SINFO_LOCAL，优先级高于--federation。
-M, --clusters=<string>	显示多个集群的分区（或节点）的 信息，对应环境变量SLURM_CLUSTERS。取值“all”代表所有集群。

查询信息-查询命令sinfo

Sinfo参数介绍（3/3）	
-o <output_format>, **--format=** **<output_format>**	按照指定的字段和格式显示信息，对应变量**SINFO_FORMAT**。 格式为“%[[.]size]<type> %[[.]size]<type> …”。 其中，点号（.）表示右对齐，size表示字段长度，type为代表特定字段的字符（或字符串）。示例如下： sinfo -o %all 以竖线分隔的形式显示所有字段。 sinfo -o "%9P %.5a %.10l %.6D %.6t %N" 显示内容：分区名-分区状态 -最大运行时间-节点数-节点状态-节点列表。
-O <output_format>, **--Format=** **<output_format>**	按照指定的格式显示状态信息。 格式为“type[:[.]size],type[:[.]size],…”。 其中，type为代表特定字段的字段名，点号（.）表示右对齐，size表示字段长度。示例如下： sinfo -O all sinfo -O Partition:9,available:.6,time:.11,nodes:.6,statecompact:.6,nodelist:.12

查询信息-查询命令sinfo

中科曙光 Sugon

格式化字段说明（1/5）		
字段名（-O）	格式化（-o）	字段说明
allocmem		节点已分配内存
allocnodes	%S	队列提交节点
available	%a	分区状态
cluster	%V	集群名
cpus	%c	节点CPU核数
cpusload	%O	节点负载
freemem	%e	节点空闲内存
cpusstate	%C	CPU统计A/I/O/T
cores	%Y	Cores/Socket
defaulttime	%L	默认运行时间

格式化字段说明（2/5）		
字段名（-O）	格式化（-o）	字段说明
disk	%d	临时盘大小MB
features	%f	节点属性
features_act	%b	Active features
groups	%g	分区用户组
gres	%G	一般性资源
maxcpuspernode	%B	单节点最大核数
memory	%m	节点总内存
nodes	%D	节点数
nodeaddr	%o	节点地址
nodeai	%A	节点统计A/I

查询信息-查询命令sinfo

格式化字段说明（3/5）		
字段名（-O）	格式化（-o）	字段说明
nodeaiot	%F	节点统计AIOT
nodehost	%n	节点主机名列表
nodelist	%N	节点名列表
oversubscribe	%h	资源超额认购
partition	%P	分区名
partitionname	%R	分区名
port		节点TCP端口
preemptmode	%M	分区抢占模式
priorityjobfactor	%I (i)	分区优先级因子
prioritytier	%p	分区调度优先级

格式化字段说明（4/5）		
字段名（-O）	格式化（-o）	字段说明
reason	%E	节点不可用原因
root	%r	仅root可用资源
size	%s	作业节点数限制
statecompact	%t	节点状态，缩写
statelong	%T	节点状态
sockets	%X	节点Socket数
socketcorethread	%z	CPU配置，S:C:T
time	%l (L)	最大运行时间
timestamp	%H	节点不可用时间
threads	%Z	单核超线程数

查询信息-查询命令sinfo

格式化字段说明（5/5）		
字段名（-O）	格式化（-o）	字段说明
user	%u	节点不可用状态的配置用户
userlong	%U	同user，额外显示用户ID
version	%v	软件版本
weight	%w	节点优先级
all	%all	全体字段，竖线分隔

查询信息-查看分区

中科曙光 Sugon

示例1: 默认格式查看分区的状态信息。

等价于 sinfo -o "%9P %.5a %.10l %.6D %.6t %N"

```
[sghpc2@login04 ~]$sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
serial       up   infinite     20   idle cmac[0011-0030]
normal       up   infinite      1 drain* cmac1174
normal       up   infinite    192  drain cmac[1091-1173,1175-1282,1304]
normal       up   infinite     54  alloc cmac[0035-0048,0057-0088,0094-0101]
normal       up   infinite   1257   idle cmac[0049-0056,0089-0093,0102-1090,1283-1303,1305-1538]
operation    up   infinite      1 drain* cmac1174
operation    up   infinite    192  drain cmac[1091-1173,1175-1282,1304]
operation    up   infinite     54  alloc cmac[0035-0048,0057-0088,0094-0101]
operation    up   infinite   1257   idle cmac[0049-0056,0089-0093,0102-1090,1283-1303,1305-1538]
sgtest       up   infinite     54  alloc cmac[0035-0048,0057-0088,0094-0101]
sgtest       up   infinite    747   idle cmac[0049-0056,0089-0093,0102-0835]
[sghpc2@login04 ~]$
```

示例2: 长格式查看分区的状态信息（--long / -l）。

等价于 sinfo -o "%9P %.5a %.10l %.10s %.4r %.8h %.10g %.6D %.11T %N"

```
[sghpc2@login04 ~]$sinfo --long
Tue Mar 27 22:51:58 2018
PARTITION AVAIL  TIMELIMIT   JOB_SIZE ROOT OVERSUBS     GROUPS  NODES       STATE NODELIST
serial       up   infinite 1-infinite   no       NO        all      1       mixed cmac0011
serial       up   infinite 1-infinite   no       NO        all     19        idle cmac[0012-0030]
normal       up   infinite 1-infinite   no       NO        all      1       down* cmac0338
normal       up   infinite 1-infinite   no       NO        all      1    draining cmac0533
normal       up   infinite 1-infinite   no       NO        all      4     drained cmac[0059-0061,1304]
normal       up   infinite 1-infinite   no       NO        all     74   allocated cmac[0037-0046,0062-0093,0530-0532,0534-0545,1122-1137,1530]
normal       up   infinite 1-infinite   no       NO        all   1423        idle cmac[0035-0036,0047-0058,0094-0337,0339-0529,0546-1121,1138-1303,1305-1400,1402-1529,1531-1538]
normal       up   infinite 1-infinite   no       NO        all      1        down cmac1401
operation    up   infinite 1-infinite   no       NO        all      1       down* cmac0338
operation    up   infinite 1-infinite   no       NO        all      1    draining cmac0533
operation    up   infinite 1-infinite   no       NO        all      4     drained cmac[0059-0061,1304]
operation    up   infinite 1-infinite   no       NO        all     74   allocated cmac[0037-0046,0062-0093,0530-0532,0534-0545,1122-1137,1530]
operation    up   infinite 1-infinite   no       NO        all   1423        idle cmac[0035-0036,0047-0058,0094-0337,0339-0529,0546-1121,1138-1303,1305-1400,1402-1529,1531-1538]
operation    up   infinite 1-infinite   no       NO        all      1        down cmac1401
sgtest       up   infinite 1-infinite   no       NO        all      1       down* cmac0338
sgtest       up   infinite 1-infinite   no       NO        all      1    draining cmac0533
sgtest       up   infinite 1-infinite   no       NO        all      3     drained cmac[0059-0061]
sgtest       up   infinite 1-infinite   no       NO        all     57   allocated cmac[0037-0046,0062-0093,0530-0532,0534-0545]
sgtest       up   infinite 1-infinite   no       NO        all    739        idle cmac[0035-0036,0047-0058,0094-0337,0339-0529,0546-0835]
[sghpc2@login04 ~]$
```

查询信息-查看分区

中科曙光 Sugon

示例3: 查看分区的摘要信息（--summarize）。

等价于 sinfo -o "%9P %.5a %.10l %.16F %N"

```
[sghpc2@login04 ~]$sinfo --summarize
PARTITION AVAIL  TIMELIMIT   NODES(A/I/O/T)  NODELIST
serial       up   infinite        1/19/0/20  cmac[0011-0030]
normal       up   infinite   91/1407/6/1504  cmac[0035-1538]
operation    up   infinite   91/1407/6/1504  cmac[0035-1538]
sgtest       up   infinite     74/723/4/801  cmac[0035-0835]
[sghpc2@login04 ~]$
```

示例4: 自定义分区摘要信息显示，添加CPU状态分布信息。

sinfo -o "%9P %.5a %.10l %.16F %.24C %N"

```
[root@login_a04 ~]#  sinfo -o "%9P %.5a %.10l %.16F %.24C  %N"
PARTITION AVAIL  TIMELIMIT   NODES(A/I/O/T)            CPUS(A/I/O/T)  NODELIST
serial       up   infinite       16/8/0/24             20/748/0/768  cmac[0011-0034]
serial_op    up   infinite       16/8/0/24             20/748/0/768  cmac[0011-0034]
largemem     up   infinite    194/32/0/226         6168/1064/0/7232  cmac[0035-0260]
normal       up   infinite 1091/413/0/1504      34806/13322/0/48128  cmac[0035-1538]
operation    up   infinite 1091/413/0/1504      34806/13322/0/48128  cmac[0035-1538]
[root@login_a04 ~]#
```

查询信息-查看分区

示例5: 查询分区的异常节点的基本信息（--list-reasons / -R）。

sinfo -o "%20E %9u %19H %N"

```
[sghpc2@login04 ~]$sinfo --list-reasons
REASON               USER      TIMESTAMP           NODELIST
Low RealMemory       slurmadm  2018-03-22T12:28:41 cmac1304
Not responding       slurmadm  2018-03-27T10:05:36 cmac0338
IB_state20           root      2018-03-26T14:45:37 cmac0533
Low RealMemory       slurmadm  2018-03-28T02:57:16 cmac[0059-0060]
Low RealMemory       slurmadm  2018-03-28T02:59:06 cmac0061
Node unexpectedly re slurmadm  2018-03-26T09:45:15 cmac1401
[sghpc2@login04 ~]$
```

示例6: 查询分区的异常节点的详细信息（--long --list-reasons / -lR）。

sinfo -o "%20E %12U %19H %6t %N"

```
[sghpc2@login04 ~]$sinfo --long --list-reasons
Wed Mar 28 03:51:54 2018
REASON               USER         TIMESTAMP           STATE  NODELIST
Low RealMemory       slurmadm(110 2018-03-22T12:28:41 drain* cmac1304
Not responding       slurmadm(110 2018-03-27T10:05:36 down*  cmac0338
IB_state20           root(0)      2018-03-26T14:45:37 drng   cmac0533
Low RealMemory       slurmadm(110 2018-03-28T02:57:16 drain  cmac[0059-0060]
Low RealMemory       slurmadm(110 2018-03-28T02:59:06 drain  cmac0061
Node unexpectedly re slurmadm(110 2018-03-26T09:45:15 down   cmac1401
[sghpc2@login04 ~]$
```

查询信息-查看分区

示例7: 自定义显示顺序（--sort= / -S ）。

排序表达式格式：[+|-]<key>,[+|-]<key>…

其中，+代表升序，-代表降序。此外，对于特殊的P，还可以是#，代表按照配置文件中的顺序显示。sinfo分区查询的默认排序方式为“#P,-t”；sinfo节点查询的默认排序方式为“N”。

```
[sghpc2@login04 ~]$sinfo --sort="t"
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
normal       up   infinite      1   down cmac1401
operation    up   infinite      1   down cmac1401
normal       up   infinite     16   idle cmac[0051-0058,0110-0114,0176-0178]
serial       up   infinite     19   idle cmac[0012-0030]
sgtest       up   infinite     16   idle cmac[0051-0058,0110-0114,0176-0178]
operation    up   infinite     16   idle cmac[0051-0058,0110-0114,0176-0178]
normal       up   infinite   1481  alloc cmac[0035-0050,0062-0109,0115-0175,0179-0337,0339-0532,0534-1303,1305-1400,1402-1538]
operation    up   infinite   1481  alloc cmac[0035-0050,0062-0109,0115-0175,0179-0337,0339-0532,0534-1303,1305-1400,1402-1538]
sgtest       up   infinite    780  alloc cmac[0035-0050,0062-0109,0115-0175,0179-0337,0339-0532,0534-0835]
serial       up   infinite      1    mix cmac0011
normal       up   infinite      3  drain cmac[0059-0061]
sgtest       up   infinite      3  drain cmac[0059-0061]
operation    up   infinite      3  drain cmac[0059-0061]
normal       up   infinite      1   drng cmac0533
sgtest       up   infinite      1   drng cmac0533
operation    up   infinite      1   drng cmac0533
normal       up   infinite      1  down* cmac0338
operation    up   infinite      1  down* cmac0338
sgtest       up   infinite      1  down* cmac0338
normal       up   infinite      1 drain* cmac1304
operation    up   infinite      1 drain* cmac1304
[sghpc2@login04 ~]$
```

查询信息-查看分区

示例8: 组合示例-查询特定分区特定状态的节点状态信息。

sinfo -p P1,P2 -t T1,T2

```
[sghpc2@login04 ~]$sinfo -p operation,normal -t idle,alloc
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
normal       up   infinite      1   drng cmac0533
normal       up   infinite   1481  alloc cmac[0035-0050,0062-0109,0115-0175,0179-0337,0339-0532,0534-1303,1305-1400,1402-1538]
normal       up   infinite     16   idle cmac[0051-0058,0110-0114,0176-0178]
operation    up   infinite      1   drng cmac0533
operation    up   infinite   1481  alloc cmac[0035-0050,0062-0109,0115-0175,0179-0337,0339-0532,0534-1303,1305-1400,1402-1538]
operation    up   infinite     16   idle cmac[0051-0058,0110-0114,0176-0178]
[sghpc2@login04 ~]$
```

SLURM作业-squeue

squeue:查询排队和运行状态的作业

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

```
[sugon@gpunode1 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                21     debug    sleep    sugon  R       0:03      2 gpunode[1-2]
```

SLURM资源-scontrol show

中科曙光 Sugon

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

SLURM队列-scontrol show

查询指定队列命令:
scontrol show partition <name>

```
[root@gpunode2 ~]# scontrol show partition debug
PartitionName=debug
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=YES QoS=N/A
   DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=UNLIMITED MaxTime=UNLIMITED MinNodes=1 LLN=NO MaxCPUsPerNode=UNLIMITED
   Nodes=gpunode[1,2]
   PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO OverSubscribe=NO
   OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=80 TotalNodes=2 SelectTypeParameters=NONE
   DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
```

SLURM节点-scontrol show

查询指定节点:
scontrol show node <name>

```
[root@gpunode2 ~]# scontrol show node gpunode1
NodeName=gpunode1 Arch=x86_64 CoresPerSocket=1
   CPUAlloc=0 CPUErr=0 CPUTot=40 CPULoad=0.37
   AvailableFeatures=(null)
   ActiveFeatures=(null)
   Gres=(null)
   NodeAddr=10.0.35.187 NodeHostName=gpunode1 Version=17.02
   OS=Linux RealMemory=30000 AllocMem=0 FreeMem=6194 Sockets=40 Boards=1
   State=IDLE ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=debug,nvidia2
   BootTime=2017-08-07T11:49:16 SlurmdStartTime=2017-08-22T15:23:13
   CfgTRES=cpu=40,mem=30000M
   AllocTRES=
   CapWatts=n/a
   CurrentWatts=0 LowestJoules=0 ConsumedJoules=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
```

SLURM作业-scontrol show

查询指定作业:
scontrol show job

```
[root@gv11 ~]# scontrol show job
JobId=119 JobName=sleep
   UserId=sugon(1000) GroupId=docker(1000) MCS_label=N/A
   Priority=4294901738 Nice=0 Account=physics QOS=testpartitionqos
   JobState=FAILED Reason=NonZeroExitCode Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=0 Reboot=0 ExitCode=1:0
   RunTime=00:00:00 TimeLimit=UNLIMITED TimeMin=N/A
   SubmitTime=2017-08-24T20:23:07 EligibleTime=2017-08-24T20:23:07
   StartTime=2017-08-24T20:23:07 EndTime=2017-08-24T20:23:07 Deadline=N/A
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   Partition=nvidia AllocNode:Sid=gv11:4295
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=gv11
   BatchHost=gv11
   NumNodes=1 NumCPUs=1 NumTasks=0 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   TRES=cpu=1,node=1
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   Gres=(null) Reservation=(null)
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=sleep
   WorkDir=/home/sugon
   Power=
```

SLURM配置-scontrol show

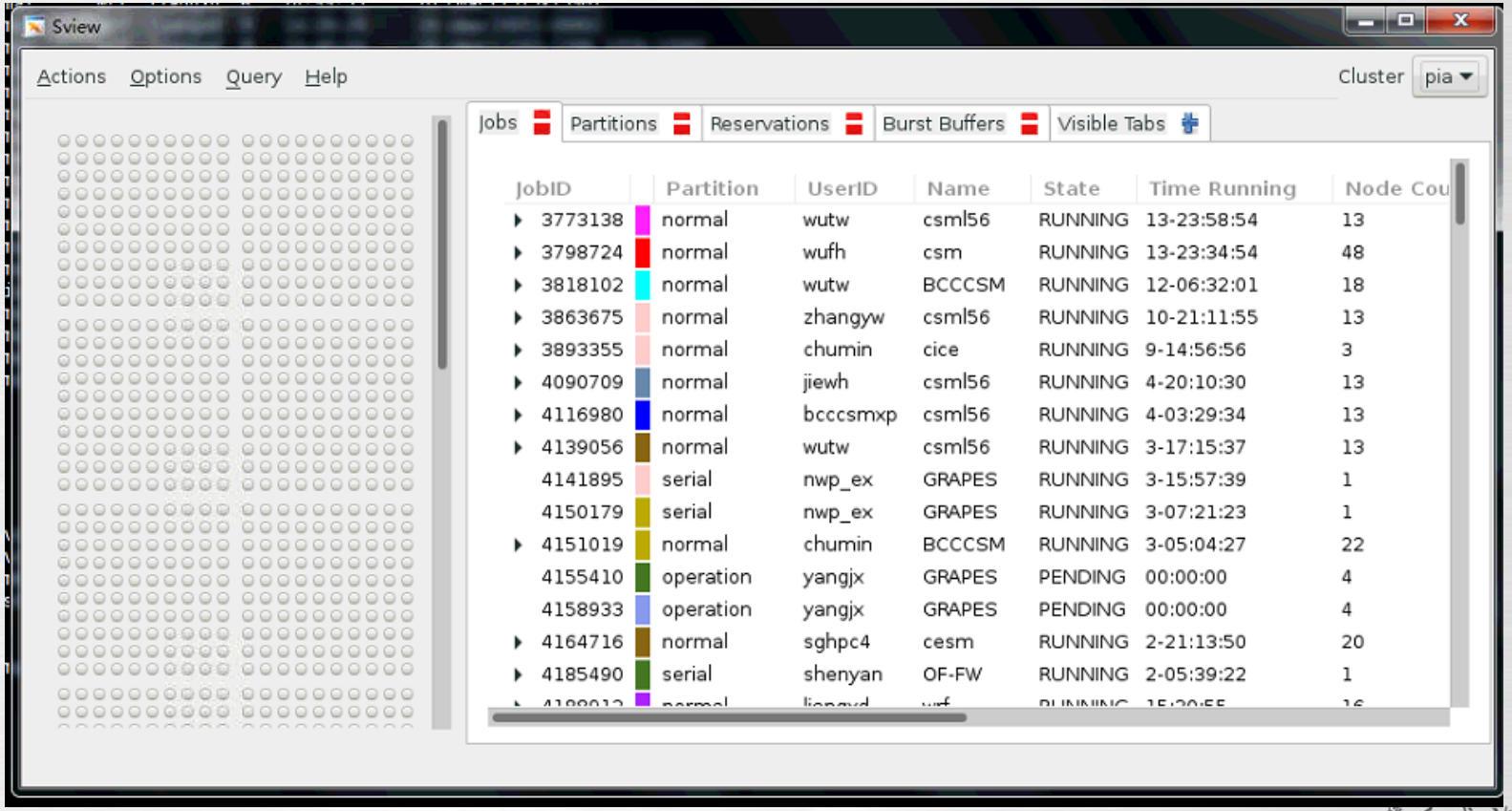
查询配置:
scontrol show config

```
[root@gv11 ~]# scontrol show config
Configuration data as of 2017-08-24T20:24:16
AccountingStorageBackupHost = (null)
AccountingStorageEnforce = associations,limits
AccountingStorageHost    = gv11
AccountingStorageLoc     = slurm_acc
AccountingStoragePort    = 3309
AccountingStorageTRES    = cpu,mem,energy,node
AccountingStorageType    = accounting_storage/mysql
AccountingStorageUser    = root
AccountingStoreJobComment = Yes
AcctGatherEnergyType     = acct_gather_energy/none
AcctGatherFilesystemType = acct_gather_filesystem/none
AcctGatherInfinibandType = acct_gather_infiniband/none
AcctGatherNodeFreq       = 0 sec
AcctGatherProfileType    = acct_gather_profile/none
AllowSpecResourcesUsage  = 0
AuthInfo                 = (null)
AuthType                 = auth/munge
BackupAddr               = (null)
BackupController         = (null)
BatchStartTimeout        = 10 sec
BOOT_TIME                = 2017-08-22T20:37:29
BurstBufferType          = (null)
CacheGroups              = 0
CheckpointType           = checkpoint/none
ChosLoc                  = (null)
```

smap命令

```
Mon Oct 29 06:15:51 2018
ID JOBID             PARTITION USER     NAME      ST      TIME NODES NODELIST
A  3773138           normal    wutw     csml56    R13-23:57:51    13 cmac[1104,1109,1115,1117-1118,1120,1122,1126,1129,1135,1184-1186]
B  3798724           normal    wufh     csm       R13-23:33:51    48 cmac[0515-0526,0531-0562,0607-0610]
C  3818102           normal    wutw     BCCCSM    R12-06:30:58    18 cmac[0748,0792,0794-0795,0797,0830-0834,0851,0855,0858,1139,1141,1143,1147,1149]
D  3863675           normal    zhangyw  csml56    R10-21:10:52    13 cmac[0579-0591]
E  3893355           normal    chumin   cice      R 9-14:55:53     3 cmac[1160,1165-1166]
F  4090709           normal    jiewh    csml56    R 4-20:09:27    13 cmac[0995-1007]
G  4116980           normal    bcccsmxp csml56    R 4-03:28:31    13 cmac[0219-0223,1334,1337-1338,1340-1344]
H  4139056           normal    wutw     csml56    R 3-17:14:34    13 cmac[0464-0466,0690,0751,1413-1414,1416-1417,1531-1533,1538]
I  4141895           serial    nwp_ex   GRAPES    R 3-15:56:36     1 cmac0011
J  4150179           serial    nwp_ex   GRAPES    R 3-07:20:20     1 cmac0011
K  4151019           normal    chumin   BCCCSM    R 3-05:03:24    22 cmac[0339-0354,0357-0358,1435-1438]
L  4164716           normal    sghpc4   cesm      R 2-21:12:47    20 cmac[0035-0050,0115-0116,0123-0124]
M  4185490           serial    shenyan  OF-FW     R 2-05:38:19     1 cmac0012
N  4188912           normal    liangxd  wrf       R   15:29:52    16 cmac[1251-1266]
O  4188913           normal    liangxd  wrf       R   15:18:54    16 cmac[1499-1514]
P  4188915           normal    liangxd  wrf       R   15:12:48    16 cmac[0129-0130,0163-0176]
Q  4188916           normal    liangxd  wrf       R   15:03:21    16 cmac[0675-0689,0752]
R  4188917           normal    liangxd  wrf       R   14:50:36    16 cmac[0127-0128,0147-0157,0160-0162]
S  4188918           normal    liangxd  wrf       R   14:49:47    16 cmac[0419-0434]
T  4188919           normal    liangxd  wrf       R   14:44:42    16 cmac[1283-1296,1319-1320]
U  4188920           normal    liangxd  wrf       R   14:44:04    16 cmac[0051-0066]
V  4188921           normal    liangxd  wrf       R   14:38:33    16 cmac[1515-1530]
W  4188922           normal    liangxd  wrf       R   14:37:35    16 cmac[1091-1103,1105-1106,1177]
X  4188923           normal    liangxd  wrf       R   14:31:34    16 cmac[0893-0894,0915-0928]
Y  4188924           normal    liangxd  wrf       R   14:30:13    16 cmac[0084-0098,0119]
```

sview命令



SLURM-提交作业

srun	sbatch	salloc
交互式作业提交	批处理作业提交	节点资源获取

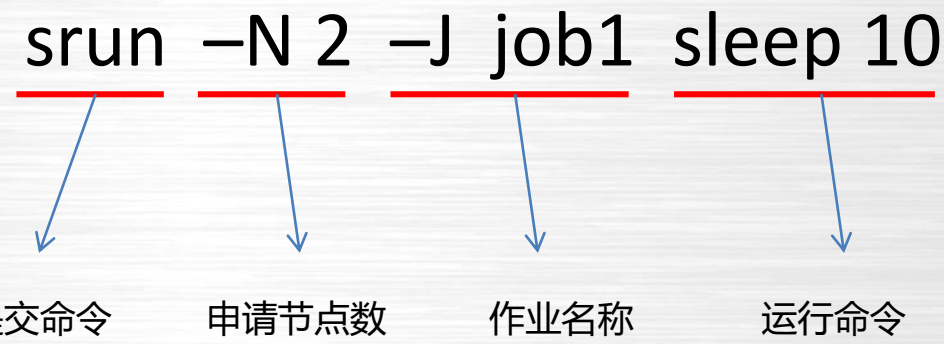
SLURM-提交作业参数

常用的作业提交参数，适用于上述三种作业提交方式。

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。

SLURM-srun

例：提交请求2个节点的并且指定作业的名称为job1



SLURM-sbatch

```
[sugon@gpunode1 ~]$ sbatch –n 4 sleep.job     //sbatch 只接收脚本
Submitted batch job 19

[sugon@gpunode1 ~]$ cat sleep.job    //脚本格式示例
#!/bin/bash
#SBATCH -J sleep              //指定作业名
#SBATCH -p debug              //指定队列
#SBATCH --time=1              //指定运行时间（分钟）
#SBATCH -N 2                  //请求节点数
#SBATCH -n 2                  //请求核心数
#SBATCH -o logs/%j.sleep      //标准输出文件
#SBATCH -e logs/%j.sleep      //错误输出文件

echo ${SLURM_JOB_NODELIST}    作业占用节点列表
echo  start on $(date)        开始时间
sleep 100                     执行命令
echo end on $(date)           结束时间
```

SLURM-salloc

```
[root@gv11 ~]# salloc –n 4                    //获取资源
salloc: Granted job allocation 117    //提交作业成功
[root@gv11 ~]# sleep 10 &                  //执行命令
[1] 1020
[root@gv11 ~]# exit                             //作业退出
exit
salloc: Relinquishing job allocation 117      //作业资源释放
[root@gv11 ~]#
```

SLURM-命令sacct的参数

sacct：查询作业

参数	解释
-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
-T, --truncate	如果一个job在 --starttime之前开始运行，开始时间将被截断为 --starttime，同样的作业结束时间 = --endtime
-o, --format	指定显示字段以逗号分隔

SLURM-命令sacct示例

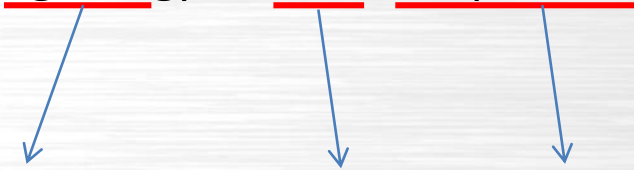
```
[root@gv11 ~]# sacct -S 2017-08-10 -E 2017-08-24
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
2                  sleep     nvidia    physics          1  COMPLETED      0:0
3              sleep.job     nvidia    physics          1  COMPLETED      0:0
3.batch            batch                physics          1  COMPLETED      0:0
4                  sleep     nvidia    physics          1  COMPLETED      0:0
5                  sleep     nvidia    physics          1  COMPLETED      0:0
6                  sleep     nvidia    physics          1  COMPLETED      0:0
7                  sleep     nvidia    physics          1  COMPLETED      0:0
8                 sleep10     nvidia    physics          1     FAILED      2:0
9                 sleep10     nvidia    physics          1     FAILED      2:0
```

```
[root@gv11 ~]# sacct -S 2017-08-10 -E 2017-08-24 -o start,end
              Start                 End
------------------- -------------------
2017-08-15T14:56:31 2017-08-15T14:56:41
2017-08-15T16:12:55 2017-08-15T16:13:06
2017-08-15T16:12:55 2017-08-15T16:13:05
2017-08-15T17:15:05 2017-08-15T17:15:15
2017-08-15T17:15:06 2017-08-15T17:15:16
2017-08-15T17:15:15 2017-08-15T17:15:25
2017-08-15T17:15:16 2017-08-15T17:15:26
2017-08-15T17:33:05 2017-08-15T17:33:05
2017-08-15T17:33:12 2017-08-15T17:33:12
2017-08-15T17:42:36 2017-08-15T17:42:46
```

SLURM-GPU请求示例

提交作业示例

srun --gres=gpu:2 -n2 sleep 10 &



作业脚本示例1—串行作业示例

串行作业的提交示例如右图所示：

示例中，calc_serial是一个串行程序，接收一个输入参数i，计算从0到（i-1）之间整数的和。

提交串行作业时，可以通过srun执行，也可以在脚本中直接调用可执行程序。

```
#!/bin/bash
#SBATCH –J TestSerial
#SBATCH -p serial
#SBATCH -N 1
#SBATCH –n 1
#SBATCH -o log/%j.loop
#SBATCH -e log/%j.loop
#SBATCH --comment=CALMET

echo   “SLURM_JOB_PARTITION=$SLURM_JOB_PARTITION”
echo   “SLURM_JOB_NODELIST=$SLURM_JOB_NODELIST”

srun  ./calc_serial 1000000
```

作业脚本示例2—MPI作业

简化的MPI作业脚本示例如右图所示：

示例中，calc_mpi是一个MPI并行程序，接收一个输入参数i，计算从0到（i-1）之间整数的和。

提交作业时，建议通过srun启动；

需要注意，如果使用intelmpi，需要事先按照示例设置I_MPI_PMI_LIBRARY变量；否则作业无法在节点内并行。

```
#!/bin/bash
#SBATCH –J mpi
#SBATCH -p normal
#SBATCH -N 2
#SBATCH –n 64
#SBATCH -o log/%j.loop
#SBATCH -e log/%j.loop
#SBATCH –comment=WRF

module load compiler/intel/composer_xe_2017.2.174
module load mpi/intelmpi/2017.2.174

export I_MPI_PMI_LIBRARY=/opt/gridview/slurm17/lib/libpmi.so
srun  ./calc_mpi 1000000
```

作业脚本示例3—单节点OpenMP作业

简化的OpenMP作业脚本示例如右图所示：

示例中，calc_openmp_init是一个单纯的OpenMP程序，接收一个输入参数i，计算从0到（i-1）之间整数的和。

提交作业时，建议通过srun启动；

需要注意，OpenMP类型的程序，需要事先设置变量OMP_NUM_THREADS，控制单进程的并发线程数。

```
#!/bin/bash
#SBATCH –J openmp
#SBATCH -p normal
#SBATCH -N 1
#SBATCH –n 1
#SBATCH –cpus-per-task=32
#SBATCH -o log/%j.loop
#SBATCH -e log/%j.loop
#SBATCH –comment=WRF

module load compiler/intel/composer_xe_2017.2.174

export OMP_NUM_THREADS=32
srun  ./calc_openmp_init 1000000
```

作业脚本示例4—MPI/OpenMP

简化的MPI/OpenMP作业脚本，示例如右图所示：

示例中，calc_openmp_init是一个单纯的OpenMP程序，接收一个输入参数i，计算从0到（i-1）之间整数的和。

提交作业时，建议通过srun启动；

需要注意，OpenMP类型的程序，需要事先设置变量OMP_NUM_THREADS，控制单进程的并发线程数。

```
#!/bin/bash
#SBATCH –J openmp_mpi
#SBATCH -p normal
#SBATCH -N 2
#SBATCH –n 4
#SBATCH –-tasks-per-node=2
#SBATCH –cpus-per-task=16
#SBATCH -o log/%j.loop
#SBATCH -e log/%j.loop
#SBATCH –comment=WRF

module load compiler/intel/composer_xe_2017.2.174
module load mpi/intelmpi/2017.2.174

export OMP_NUM_THREADS=16
srun  ./calc_openmp_mpi 1000000
```

作业脚本示例—气候模式

```
#!/bin/bash
#SBATCH -J MOD_01
#SBATCH -o log/%j.loop
#SBATCH -e log/%j.loop
#SBATCH --mem=2G
#SBATCH -p debug
#SBATCH -n 13 -N 1
#SBATCH -c 2
#SBATCH packjob
#SBATCH -J MOD_02
#SBATCH -p debug
#SBATCH --mem=2G
#SBATCH -n 2 -N 1
#SBATCH -c 1
#SBATCH packjob
#SBATCH --mem=2G
#SBATCH -J MOD_03
#SBATCH -p debug
#SBATCH -n 1 -N 1
#SBATCH -c 2

echo "SLURM_JOB_NODELIST=${SLURM_JOB_NODELIST}"
echo "SLURM_NODELIST=${SLURM_NODELIST}"

module load compiler/intel/composer_xe_2018.1.163
module load mpi/intelmpi/2018.1

export I_MPI_PMI_LIBRARY=/opt/gridview/slurm17/lib/libpmi.so

# for 16 proc, srun 10000000 = 74 sec
 runtime=100000000

#export OMP_NUM_THREADS=2

 time srun --label --mpi=pmi2 --export=OMP_NUM_THREADS=2,ALL ./calc_openmp
$runtime : --export=OMP_NUM_THREADS=1,ALL ./calc_openmp2 $runtime : --
export=OMP_NUM_THREADS=2,ALL ./calc_openmp3 $runtime
```

注意事项：

1. 各模块的作业描述以#SBATCH packjob行分隔；
2. 各模块的作业分别指定自己的描述信息；
3. 各模块的启动需要借助srun，各模块的启动命令以“:”分隔；
4. 各模块如需指定不同同一环境变量的不同取值，则该变量**不能**在srun之前指定默认值，必须要在srun中的各个模块中通过—export=<var=x,ALL>分别指定，如OMP_NUM_THREADS。

作业脚本示例—作业依赖关联

普通用户提交作业时可以指定作业之间的依赖关系，通过各种逻辑关系的设置来完成一个复杂的业务处理流程。

常用的逻辑关系包括：after（依赖作业开始运行时）、afterok（依赖作业正常结束时）、afterany（依赖作业均完成）、afternotok（依赖作业非正常结束）。

如下例所示：提交A、B、C三个作业，其依赖关系为：B作业需要在A作业正常结束后才能运行，C作业需要在作业B正常完成后开始。

- 编写业务脚本

[test@gvm03 cma]# **vim depdencysubmit**

```
#! /bin/bash

AJobId=`sbatch JobA.slurm | awk '{print $4}'`
if [ $? -ne 0 ]; then
        echo "Failed to submit JobA.slurm"
        exit 1
fi

BJobId=`sbatch --dependency=afterok:$AJobId JobB.slurm | awk '{print $4}'`
if [ $? -ne 0 ]; then
        echo "Failed to submit JobB.slurm"
        exit 1
fi

CJobId=`sbatch --dependency=afterok:$BJobId JobC.slurm | awk '{print $4}'`
if [ $? -ne 0 ]; then
        echo "Failed to submit JobC.slurm"
        exit 1
fi
```

作业脚本示例—作业依赖关联

- 提交业务脚本
  bash depdencysubmit

- 查看作业运行状态
  squeue

```
             JOBID PARTITION     NAME     USER ST       TIME  NODES
NODELIST(REASON)
                57     debug     LOOP     root PD       0:00      1 (Dependency)
                58     debug     LOOP     root PD       0:00      1 (Dependency)
                56     debug     LOOP     root  R       0:02      1 cmac0178
```

- 查看JOBID 56作业的详细信息

scontrol show jobs 56

```
JobId=56 JobName=LOOP
   UserId=root(0) GroupId=root(0) MCS_label=N/A
   Priority=1 Nice=0 Account=root QOS=normal
   JobState=COMPLETED Reason=None Dependency=(null)
   Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0
   RunTime=00:00:31 TimeLimit=UNLIMITED TimeMin=N/A
   SubmitTime=2018-01-09T18:06:16 EligibleTime=2018-01-09T18:06:16
   StartTime=2018-01-09T18:06:17 EndTime=2018-01-09T18:06:48
Deadline=N/A
   PreemptTime=None SuspendTime=None SecsPreSuspend=0
   LastSchedEval=2018-01-09T18:06:17
   Partition=debug AllocNode:Sid=cmacm03:109650
   ReqNodeList=(null) ExcNodeList=(null)
   NodeList=cmac0178
   BatchHost=cmac0178
   NumNodes=1 NumCPUs=10 NumTasks=10 CPUs/Task=1
ReqB:S:C:T=0:0:*:*
   TRES=cpu=10,mem=1M,node=1,billing=10
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=*
   MinCPUsNode=1 MinMemoryNode=1M MinTmpDiskNode=0
   Features=(null) DelayBoot=00:00:00
   Gres=(null) Reservation=(null)
   OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null)
   Command=/root/cma/JobA.slurm
   WorkDir=/root/cma
   StdErr=/dev/null
   StdIn=/dev/null
   StdOut=/dev/null
   Power=
```

作业回填

启用回填策略

SchedulerType=sched/backfill

```
[sghpc2@gv178 ~]$ sbatch -N 2 -n 15 sleep.slurm
Submitted batch job 54
[sghpc2@gv178 ~]$ sbatch -N 2 -n 32 sleep.slurm
Submitted batch job 55
```

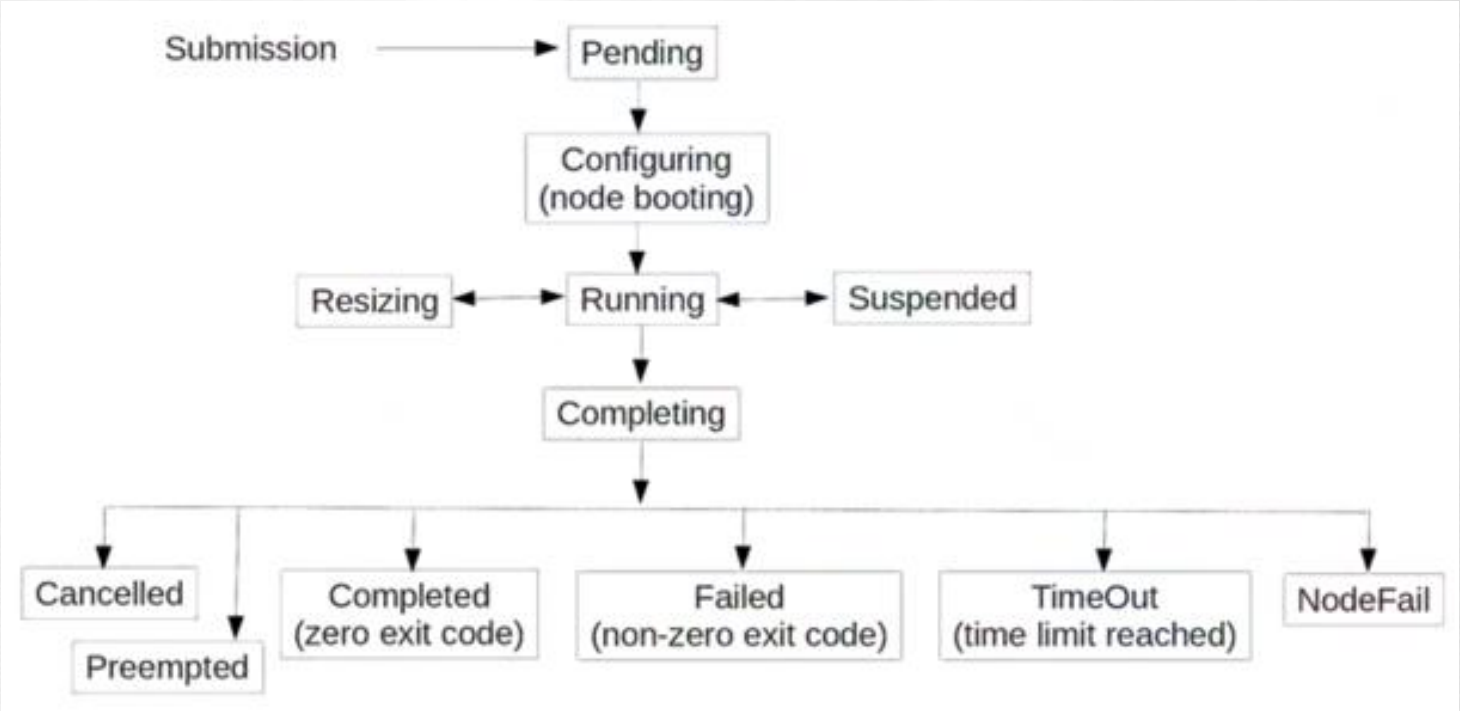
```
[sghpc2@gv178 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES   CPUS          START_TIME
          END_TIME NODELIST(REASON)
                55       low HETE-MPI   sghpc2 PD       0:00      2     32 2018-11-16T08:55:39
2018-11-16T20:55:39 (Resources)
                54       low HETE-MPI   sghpc2  R       0:50      2     15 2018-11-15T20:55:39
2018-11-16T08:55:39 gv[11,178]
```

小作业回填大作业

当前资源满足作业要求并且作业运行时间较短的作业可以在不影响前面排队作业启动的情况下调度运行，即回填生效

```
[sghpc2@gv178 ~]$ sbatch -N 1 -n 10 -t 01:00:00 sleep.slurm
Submitted batch job 56
[sghpc2@gv178 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES   CPUS          START_TIME
          END_TIME NODELIST(REASON)
                55       low HETE-MPI   sghpc2 PD       0:00      2     32 2018-11-16T08:55:39
2018-11-16T20:55:39 (Resources)
                54       low HETE-MPI   sghpc2  R       1:34      2     15 2018-11-15T20:55:39
2018-11-16T08:55:39 gv[11,178]
                56       low HETE-MPI   sghpc2  R       0:00      1     10 2018-11-15T20:57:13
2018-11-15T21:57:13 gv11
```

作业状态变化


Submission
Pending
Configuring
(node booting)
Resizing
Running
Suspended
Completing
Cancelled
Preempted
Completed
(zero exit code)
Failed
(non-zero exit code)
TimeOut
(time limit reached)
NodeFail

目录

服务控制

服务控制

服务启动

```
[root@cmacm03 ~]# systemctl start slurmctld
[root@cmacm03 ~]# echo $?
0
[root@cmacm03 ~]#
```

服务停止

```
[root@cmacm03 ~]# systemctl stop slurmctld
[root@cmacm03 ~]# echo $?
0
[root@cmacm03 ~]#
```

查看服务状态

```
[root@cmacm03 ~]# systemctl status slurmctld
● slurmctld.service - Slurm controller daemon
   Loaded: loaded (/usr/lib/systemd/system/slurmctld.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2018-01-22 04:03:20 UTC; 1min 41s ago
  Process: 86072 ExecStart=/opt/gridview/slurm17/sbin/slurmctld $SLURMCTLD_OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 86075 (slurmctld)
    Tasks: 16 (limit: 5120)
   CGroup: /system.slice/slurmctld.service
           └─86075 /opt/gridview/slurm17/sbin/slurmctld

Jan 22 04:03:20 cmacm03 systemd[1]: Starting Slurm controller daemon...
Jan 22 04:03:20 cmacm03 systemd[1]: PID file /var/run/slurmctld.pid not readable (yet?) after start.
Jan 22 04:03:20 cmacm03 systemd[1]: Started Slurm controller daemon.
```

作业管理

scontrol：控制作业命令

COMMAND	解释
scontrol suspend <jobid>	挂起作业
scontrol resume <jobid>	恢复作业
scontrol requeue <jobid>	作业重新排队
scontrol hold <id/name>	保留作业
Scontrol release <id/name>	释放作业

作业管理

```
[root@gv62 ~]# scontrol suspend 126
[root@gv62 ~]# squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               127     debug     LOOP     root  R       1:24      1 gv62
               128     debug     LOOP     root  R       1:24      1 gv62
               129     debug     LOOP     root  R       1:24      1 gv170
               130     debug     LOOP     root  R       1:24      1 gv170
               126     debug     LOOP     root  S       1:12      1 gv62
[root@gv62 ~]# scontrol resume 126
[root@gv62 ~]# squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               127     debug     LOOP     root  R       1:34      1 gv62
               128     debug     LOOP     root  R       1:34      1 gv62
               129     debug     LOOP     root  R       1:34      1 gv170
               130     debug     LOOP     root  R       1:34      1 gv170
               126     debug     LOOP     root  R       1:13      1 gv62
[root@gv62 ~]# scontrol requeue 126
[root@gv62 ~]# squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               126     debug     LOOP     root PD       0:00      1 (Dependency)
[root@gv62 ~]# scontrol hold 126
[root@gv62 ~]# squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               126     debug     LOOP     root PD       0:00      1 (JobHeldAdmin)
[root@gv62 ~]# scontrol release 126
[root@gv62 ~]# squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               126     debug     LOOP     root  R       0:14      1 gv62
```

节点管理(1/2)

查询节点：

（1）sinfo

（2）sinfo -N

控制节点：

（1）下线节点：

scontrol update nodename=gv21 state=drain reason="hardware error"

（2）上线节点

scontrol update nodename=gv21 state=idle

（3）节点清空

scontrol update nodename=gv21 state=down reason="debug"

（4）节点恢复

scontrol update nodename=gv21 state=resume

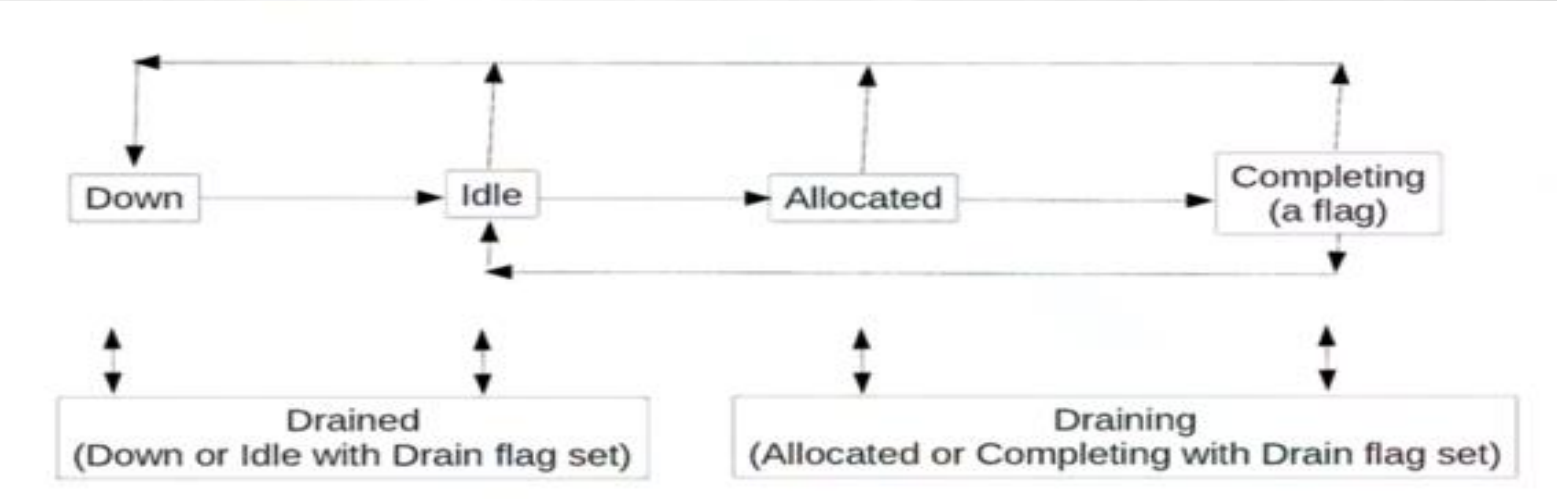
```
[zhangtao@gv21 cma]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*       up 3-00:00:00      1   idle gv21
testq        up 3-00:00:00      1   idle gv21
singer       up 3-00:00:00      1   idle gv21
double       up 3-00:00:00      1   idle gv21
double2      up 3-00:00:00      1   idle gv21
[zhangtao@gv21 cma]$
```

```
[zhangtao@gv21 cma]$ sinfo -N
NODELIST   NODES PARTITION STATE
gv21           1     debug* idle
gv21           1      testq idle
gv21           1     singer idle
gv21           1     double idle
gv21           1    double2 idle
[zhangtao@gv21 cma]$
```

```
[root@gv21 ~]#
[root@gv21 ~]# scontrol update nodename=gv21 state=down reason="debug"
[root@gv21 ~]# sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*       up 3-00:00:00      1   idle gv21
testq        up 3-00:00:00      1   idle gv21
singer       up 3-00:00:00      1   idle gv21
double       up 3-00:00:00      1   idle gv21
double2      up 3-00:00:00      1   idle gv21
[root@gv21 ~]# squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
               175     debug  MEMTEST zhangtao PD       0:00      1 (BeginTime)
```

节点管理(2/2)

状态转换图



分区管理(1/3)

查询分区：

sinfo –p debug

```
[root@gv21 ~]# sinfo -p debug
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*       up 3-00:00:00      1   idle gv21
[root@gv21 ~]#
```

分区详情：

scontrol show partion debug

```
[root@gv21 ~]# scontrol show partition debug
PartitionName=debug
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=YES QoS=N/A
   DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=UNLIMITED MaxTime=3-00:00:00 MinNodes=1 LLN=NO MaxCPUsPerNode=UNLIMITED
   Nodes=gv21
   PriorityJobFactor=1024 PriorityTier=2000 RootOnly=NO ReqResv=NO OverSubscribe=NO
   OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=40 TotalNodes=1 SelectTypeParameters=NONE
   DefMemPerCPU=4096 MaxMemPerNode=UNLIMITED

[root@gv21 ~]#
```

调整状态：

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

下线分区：

scontrol update PartitionName=debug State=Down

上线分区：

scontrol update PartitionName=debug State=Idle

```
[root@gv21 ~]# sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*       down 3-00:00:00      1   idle gv21
testq          up 3-00:00:00      1   idle gv21
singer         up 3-00:00:00      1   idle gv21
double         up 3-00:00:00      1   idle gv21
double2        up 3-00:00:00      1   idle gv21
[root@gv21 ~]#
```

分区管理(2/3)

分区用户限制

通过设置分区的可访问账号进行分区的用户限制，即只允许部分用户使用分区。

```
PartitionName=normal    Nodes=cmac[0035-1538] Priority=1000  OverSubscribe=FORCE:1 Default=NO  AllowAccounts=ALL QOS=normal_qos Default
Time=15-00:00:00 MaxTime=INFINITE DefMemPerCPU=5120 State=UP
PartitionName=operation Nodes=cmac[0035-1538] Priority=2000  OverSubscribe=FORCE:1 Default=NO  AllowAccounts=nwp,nwp_op,nwp_sp,lijuan,n
wp_pd,nwp_qu,nwpbj_ex DefaultTime=15-00:00:00 MaxTime=INFINITE State=UP
```

分区配置：

AllowAccounts=ALL　　所有账号均可以使用

AllowAccounts=accountName[,]

某些账号可以使用，如opea,nwp

```
[sghpc2@login_a01 haowj]$ sbatch -p operation test
Submitted batch job 4577265
[sghpc2@login_a01 haowj]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES   CPUS         START_TIME           END_TIME PRIORITY NODELIST(RE
ASON)
           4577265 operation     LOOP   sghpc2 PD       0:00      2     32                N/A                N/A     1100 (AccountNot
Allowed)
```

分区管理(3/3)

作业抢占

通过设置队列的优先级实现队列间作业的抢占，高优先级队列资源不够时可以抢占低优先级队列作业资源。

```
PartitionName=low Nodes=ALL Default=YES MaxTime=INFINITE Priority=1000 DefaultTime=01:00:00 State=UP
PartitionName=middle Nodes=ALL Default=NO  MaxTime=INFINITE Priority=2000 DefaultTime=01:00:00 State=UP
PartitionName=high Nodes=ALL Default=NO  MaxTime=INFINITE Priority=3000 DefaultTime=01:00:00 State=UP
```

队列优先级参数：Priority=1000

slurm.conf
抢占设置

```
PreemptMode=requeue,gang
#PreemptMode=cancel,gang
#PreemptMode=suspend,gang
PreemptType=preempt/partition_prio
```

```
[sghpc2@gv178 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES   CPUS          START_TIME            END_TIME NODELIST(REASON)
                30       low HETE-MPI   sghpc2 PD       0:00      1      3 2018-11-14T22:22:00 2018-11-14T23:22:00 (Nodes required fo
r job are DOWN, DRAINED or reserved for jobs in higher priority partitions)
                31    middle HETE-MPI   sghpc2 PD       0:00      1      3 2018-11-14T21:22:31 2018-11-14T22:22:31 (Resources)
                32      high HETE-MPI   sghpc2  R       0:33      1      3 2018-11-14T20:22:31 2018-11-14T21:22:31 gv178
```

账号管理(1/2)

查询账户：

sacctmgr show account

sacctmgr show account withassoc

添加账户：

sacctmgr add account acct03 [Parent=<root>]

修改账户：

sacctmgr modify account acct03 set GrpJobs=2

sacctmgr modify account acct03 set Parent=root

删除账号：

sacctmgr delete account acct03

```
[root@gv21 ~]# sacctmgr show account
   Account                Descr                  Org
---------- -------------------- --------------------
    acct01               acct01               acct01
    acct02               acct02               acct02
      root default root account                 root
[root@gv21 ~]#
```

```
[root@gv21 ~]# sacctmgr add account acct03
 Adding Account(s)
  acct03
 Settings
  Description     = Account Name
  Organization    = Parent/Account Name
 Associations
  A = acct03     C = cma021
Would you like to commit changes? (You have 30 seconds to decide)
(N/y): y
[root@gv21 ~]#
```

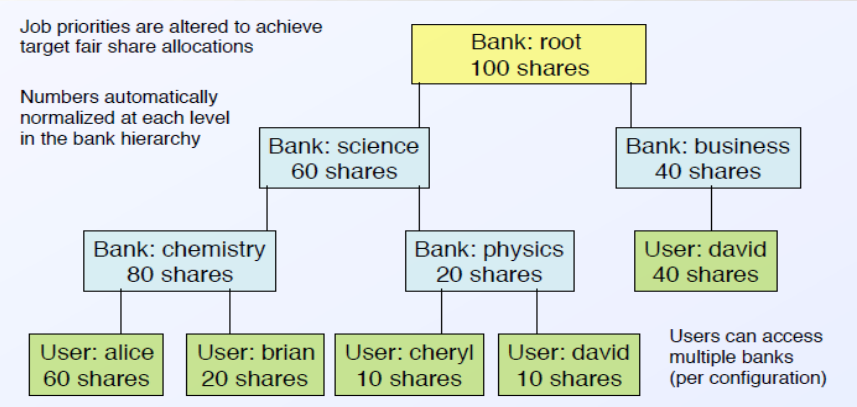
```
[root@gv21 ~]# sacctmgr show accounts
   Account                Descr                  Org
---------- -------------------- --------------------
    acct01               acct01               acct01
    acct02               acct02               acct02
    acct03               acct03               acct03
      root default root account                 root
[root@gv21 ~]#
[root@gv21 ~]# sacctmgr modify accounts acct03 set GrpJobs=2
 Modified account associations...
  C = cma021     A = acct03 of root
Would you like to commit changes? (You have 30 seconds to decide)
(N/y): y
[root@gv21 ~]#
```

账号管理(2/2)

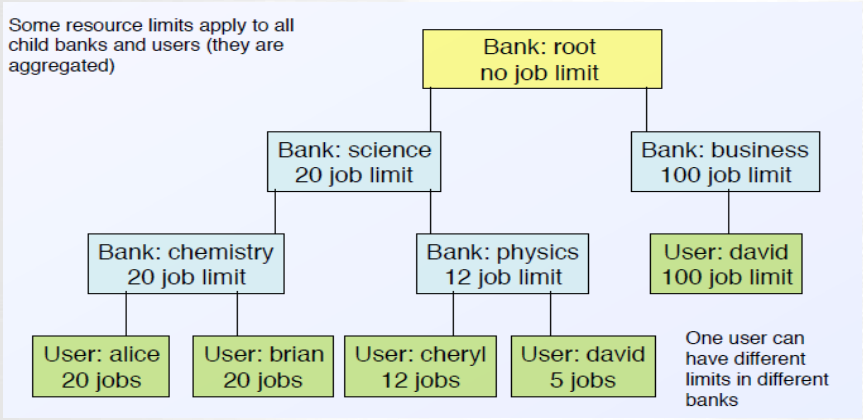
关联功能：

- 队列权限
- 公平共享
- 资源限制
- 记账统计

公平共享



资源限制



统计记账

```
[root@login_a01 ~]# sacct -A liangxd -S 2018-10-29T12:00:00 -E 2018-10-29T13:00:00
       JobID    JobName  Partition    Account  AllocCPUS      State ExitCode
------------ ---------- ---------- ---------- ---------- ---------- --------
4188930              wrf     normal    liangxd        512  COMPLETED      0:0
4188930.bat+      batch               liangxd         32  COMPLETED      0:0
4188930.0     geogrid.e+              liangxd         32  COMPLETED      0:0
4188930.1     metgrid.e+              liangxd         64  COMPLETED      0:0
4188930.2       real.exe              liangxd        128  COMPLETED      0:0
4188930.3        gsi.exe              liangxd        128  COMPLETED      0:0
4188930.4        wrf.exe              liangxd        256  COMPLETED      0:0
4188930.5        gsi.exe              liangxd        128  COMPLETED      0:0
4188930.6        gsi.exe              liangxd        128  COMPLETED      0:0
4188930.7        gsi.exe              liangxd        128  COMPLETED      0:0
4188930.8        gsi.exe              liangxd        128  COMPLETED      0:0
4188930.9     geogrid.e+              liangxd         32  COMPLETED      0:0
4188930.10    metgrid.e+              liangxd         64  COMPLETED      0:0
4188930.11      real.exe              liangxd        128  COMPLETED      0:0
4188930.12       gsi.exe              liangxd        128  COMPLETED      0:0
4188930.13       wrf.exe              liangxd        256  COMPLETED      0:0
```

QOS管理

查询QOS：

sacctmgr show qos

（normal是默认的用户QOS）

添加QOS：

sacctmgr add qos test_qos

修改QOS：

（分别设置单用户的运行作业数、使用核心数、提交作业数）

sacctmgr modify qos test_qos set MaxJobsPerUser=4

sacctmgr modify qos test_qos set MaxTRESPU="1=6"

sacctmgr modify qos test_qos set MaxSubmitJobsPerUser=6

（取消限制取值为-1）

删除QOS：

sacctmgr delete qos test_qos

```
[root@login_a01 ~]# sacctmgr show qos
      Name   Priority  GraceTime    Preempt PreemptMode                                    Flags UsageThres UsageFactor      GrpTRES
      MaxTRES MaxTRESPerNode   MaxTRESMins     MaxWall    MaxTRESPU MaxJobsPU MaxSubmitPU     MaxTRESPA MaxJobsPA MaxSubmitPA       M
---------- ---------- ---------- ---------- ---------- ------------- --------- ----------- ---------- --------- ----------- -------
    normal          0   00:00:00               cluster                                                  1.000000
                                                        cpu=8200       150         200
  test_qos        100   00:00:00               cluster                                                  1.000000
   special          0   00:00:00               cluster                                                  1.000000
                                                        cpu=40000       50         200
normal_qos          0   00:00:00               cluster                                                  1.000000
                                                                        16          50
 operation          0   00:00:00               cluster                                                  1.000000
                                                        cpu=10240      200         500
sghpc2_qos        400   00:00:00               cluster                                                  1.000000
                                                         cpu=2000       40
   normal1          0   00:00:00               cluster                                                  1.000000
                                                        cpu=12300       50         200
  special1          0   00:00:00               cluster                                                  1.000000
                                                         cpu=8200      200        2100
[root@login_a01 ~]#
```

```
[root@login_a01 ~]# sacctmgr modify qos test_qos set MaxTRESPU="1=6"
 Modified qos...
  test_qos
Would you like to commit changes? (You have 30 seconds to decide)
(N/y): y
[root@login_a01 ~]# sacctmgr show qos test_qos
      Name   Priority  GraceTime    Preempt PreemptMode                                    Flags UsageThres UsageFactor      GrpTRES
      MaxTRES MaxTRESPerNode   MaxTRESMins     MaxWall    MaxTRESPU MaxJobsPU MaxSubmitPU     MaxTRESPA MaxJobsPA MaxSubmitPA
---------- ---------- ---------- ---------- ---------- ------------- --------- ----------- ---------- --------- -----------
  test_qos        100   00:00:00               cluster                                                  1.000000
                                                             cpu=6
[root@login_a01 ~]#
```

用户管理

查询用户：

sacctmgr show user withassoc

添加用户：

sacctmgr add user sghpc2 DefaultAccount=acct02 [QOS=test_qos] [Cluster=pia]

修改用户：

sacctmgr modify user sghpc2 set QOS=normal

删除用户：

sacctmgr delete user sghpc2

```
[root@gv021 etc]# sacctmgr add user sghpc2 DefaultAccount=acct02 QOS=test_qos
There is no uid for user 'sghpc2'
Are you sure you want to continue? (You have 30 seconds to decide)
(N/y): y
 Adding User(s)
  sghpc2
 Settings =
  Default Account = acct02
 Associations =
  U = sghpc2    A = acct02     C = gv021
  U = sghpc2    A = acct02     C = gv022
 Non Default Settings
  QOS           = test_qos
Would you like to commit changes? (You have 30 seconds to decide)
(N/y): y
[root@gv021 etc]#
```

```
[root@gv021 etc]# sacctmgr modify user sghpc2 set DefaultAccount=acct03
 Can't modify because these users aren't associated with new default account 'acct03'...
  U = sghpc2 C = gv021
  U = sghpc2 C = gv022
[root@gv021 etc]# sacctmgr modify user sghpc2 set QOS=normal
 Modified user associations...
  C = gv021     A = acct02            U = sghpc2
  C = gv022     A = acct02            U = sghpc2
Would you like to commit changes? (You have 30 seconds to decide)
(N/y): y
[root@gv021 etc]#
[root@gv021 etc]# sacctmgr show users sghpc2 withassoc
      User   Def Acct     Admin    Cluster    Account  Partition     Share MaxJobs MaxNodes  MaxCPUs MaxSubmit     MaxWall  MaxCPUMins                  QOS   Def QOS
---------- ---------- --------- ---------- ---------- ---------- --------- ------- -------- -------- --------- ----------- ----------- -------------------- ---------
    sghpc2     acct02      None       gv021     acct02                  1                                                                          normal
    sghpc2     acct02      None       gv022     acct02                  1                                                                          normal
[root@gv021 etc]#
```

优先级管理

系统支持两种优先级策略：

1. priority/basic
2. priority/multifactor

Multifactor权重参数：

1. PriorityWeightAge
2. PriorityWeightFairshare
3. PriorityWeightJobSize
4. PriorityWeightPartition
5. PriorityWeightQOS
6. PriorityWeightTRES

优先级修改：

scontrol update JobID=X Priority=Y

恢复优先级：

Scontrol hold X

Scontrol release X

```
Job_priority =
        (PriorityWeightAge) * (age_factor) +
        (PriorityWeightFairshare) * (fair-share_factor) +
        (PriorityWeightJobSize) * (job_size_factor) +
        (PriorityWeightPartition) * (partition_factor) +
        (PriorityWeightQOS) * (QOS_factor) +
        SUM(TRES_weight_cpu * TRES_factor_cpu,
            TRES_weight_<type> * TRES_factor_<type>,
            ...)
```

Age_factor：

排队时间/PriorityMaxAge

Job_size_factor：

（占用节点数/总节点数+占用核心数/总核心数）/2

partition_factor:

分区JobFactor/最大分区Factor

QOS_factor

所用qos优先级/最大QOS优先级

Tres_factor

作业CPU数/队列CPU数*权重+作业内存/队列总内存*权重

预约管理

提交作业：
sbatch --reservation=test3_1 test.slurm

创建资源预约：
(1)用root用户为test3用户创建一个包含gv0011节点的预留，如下所示：

```
[root@cmacm03 ~]# scontrol create reservation user=test3 starttime=now duration
=60 flags=maint,ignore_jobs nodes=cmac0011
Reservation created: test3_1
```

(2)查看预约

```
[root@cmacm03 ~]# scontrol show reservation
ReservationName=test3_1 StartTime=2018-01-22T02:54:33 EndTime=2018-01-22T03:54:
33 Duration=01:00:00
   Nodes=cmac0011 NodeCnt=1 CoreCnt=32 Features=(null) PartitionName=(null) Fla
gs=MAINT,IGNORE_JOBS,SPEC_NODES
   TRES=cpu=32
   Users=test3 Accounts=(null) Licenses=(null) State=ACTIVE BurstBuffer=(null)
Watts=n/a
```

(3)更新预约

```
[root@cmacm03 ~]# scontrol update ReservationName=test3_1 duration=120
Reservation updated.
You have new mail in /var/spool/mail/root
[root@cmacm03 ~]# scontrol show reservation
ReservationName=test3_1 StartTime=2018-01-22T02:54:33 EndTime=2018-01-22T04:54:
33 Duration=02:00:00
   Nodes=cmac0011 NodeCnt=1 CoreCnt=32 Features=(null) PartitionName=(null) Fla
gs=MAINT,IGNORE_JOBS,SPEC_NODES
   TRES=cpu=32
   Users=test3 Accounts=(null) Licenses=(null) State=ACTIVE BurstBuffer=(null)
Watts=n/a
```

(4)删除预约

```
[root@cmacm03 ~]# scontrol delete reservation=test3_1
[root@cmacm03 ~]# scontrol show reservation
No reservations in the system
```

预约管理

SLURM具备为USER/ACCOUNT的作业预留资源（如Cores、Nodes、Licenses）的功能。一般来说，资源预留至少包括如下3项关键属性：

（1）预留哪些资源；
（2）预留多少时间；
（3）为哪些用户预留。

预约仅仅可以被root用户或配置的SlurmUser使用scontrol命令进行创建，更新，删除。

资源预约参数：
ReservationName，StartTime，EndTime，Duration，Nodes，NodeCnt，CoreCnt，Features，PartitionName，Flags，TRES，Users，Accounts，Licenses，State

Flag参数：
maint：为记账目的进行资源预约
ignore_jobs：创建预约时忽视正在运行的作业
daily：每日预约
overlap：预约重叠

```
scontrol create reservation user=sghpc2 starttime=now
duration=60 flags=maint,ignore_jobs nodes=cmbc1513
```

作业前后处理

SLURM作业调度软件支持作业运行前后的脚本运行

前处理脚本slurm.prolog
后处理脚本slurm.epilog

```
[sghpc2@cmacm02 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES   CPUS         START_TIME
  END_TIME NODELIST(REASON)
           2251239       low     long   sghpc2  R       0:00      1     20 2018-06-12T11:36:11 2018-06-
12T23:36:11 cmac1451
```

slurm.conf配置
Epilog=/opt/gridview/slurm17/etc/slurm.epilog
Prolog=/opt/gridview/slurm17/etc/slurm.prolog

```
[root@cmac1451_a302r5n1 tmp]#cat prolog.2251239.log
Tue Jun 12 11:36:11 UTC 2018
```

slurm.prolog
#!/bin/bash
date >> /tmp/prolog.${SLURM_JOB_ID}.log

```
[root@cmac1451_a302r5n1 tmp]#cat epilog.2251239.log
Tue Jun 12 11:39:32 UTC 2018
```

slurm.epilog
#!/bin/bash
date >> /tmp/epilog.${SLURM_JOB_ID}.log

目录

常见问题处理

1、节点维护

（1）下线节点：

scontrol update nodename=gv21 state=drain reason=“hardware error”

（2）上线节点：

scontrol update nodename=gv21 state=idle

2、队列维护

（1）下线队列：

接收不调度

scontrol update PartitionName=debug State=Down

调度不接收

scontrol update PartitionName=debug State=Drain

（2）上线分区：

scontrol update PartitionName=debug State=UP

常见问题处理

3、节点状态为down
sinfo -R可以看到节点down和原因

```
[root@cmbcm02 ~]# sinfo -R
REASON               USER      TIMESTAMP           NODELIST
Not responding       slurmadm  2018-11-14T11:09:11 cmbc0498
Not responding       slurmadm  2018-11-14T09:07:31 cmbc0755
```

（1）查看计算节点munge服务是否正常
systemctl status munge.service

（2）查看计算节点slurmd服务是否正常
systemctl status slurmd.service

4、节点状态为comp
（1）计算节点down
scontrol update nodename=cmbc0646 state=down reason=comp

（2）计算节点恢复：scontrol update nodename=cmbc0646 state=resume

```
[root@cmbcm02 ~]# sinfo -t comp
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
serial       up   infinite      0    n/a
serial_op    up   infinite      0    n/a
largemem     up   infinite      0    n/a
normal       up   infinite      2   comp cmbc[0646,0994]
operation    up   infinite      2   comp cmbc[0646,0994]
```

常见问题处理

5、节点状态为down

slurmd服务重启失败

处理：查看slurm服务，若有作业残留，确认该作业已结束后，强制删除残留作业，重启slurmd服务正常。

```
[root@cmbc0646_b608r5n4 ~]#systemctl status slurmd
● slurmd.service - Slurm node daemon
   Loaded: loaded (/usr/lib/systemd/system/slurmd.service; disabled; vendor preset: disabled
   Active: failed (Result: timeout) since Mon 2018-11-05 01:51:44 UTC; 8min ago
  Process: 207608 ExecStart=/opt/gridview/slurm17/sbin/slurmd $SLURMD_OPTIONS (code=killed,
 Main PID: 138139 (code=killed, signal=KILL)
    Tasks: 260
   CGroup: /system.slice/slurmd.service
           └─30345 slurmstepd: [3157341.1]

Nov 05 01:50:14 cmbc0646 systemd[1]: Starting Slurm node daemon...
Nov 05 01:51:44 cmbc0646 systemd[1]: slurmd.service start operation timed out. Terminating.
Nov 05 01:51:44 cmbc0646 systemd[1]: Failed to start Slurm node daemon.
Nov 05 01:51:44 cmbc0646 systemd[1]: Unit slurmd.service entered failed state.
Nov 05 01:51:44 cmbc0646 systemd[1]: slurmd.service failed.
```

```
[root@cmbc0852_b611r6n2 slurmd_spool]#ps -ef|grep slurm
root       3079      1  0 Nov03 ?        00:00:00 slurmstepd: [3157371.1]
slurmadm  15429      1  0 Jul05 ?        00:04:57 /opt/gridview/munge/sbin/munged --num-threads=10
root      65565      1  0 Sep26 ?        00:01:32 /opt/gridview/slurm17/sbin/slurmd
root     167225 166888  0 02:11 pts/0    00:00:00 grep --color=auto slurm
```

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