SLURM作业调度部分常见问题分析

HPC产品事业部

2022-05-23

携手成就梦想

常见问题

1、sinfo显示报错：Zero Bytes were transmitted or received。

答：查看当前节点与管理节点的系统时间是否同步。

2、有的作业一直排不上队，如何提高作业调度的成功率

通常情况下，系统会启用回填算法（sched/backfill）。

作业排队不能运行，通常是下面的原因导致：

A、系统空闲资源不足；

B、该作业优先级不是最高，且无法满足回填条件（运行时间太长）。

因此，作业提交者想要改善自身的作业运行机会，可以从两个方面考虑：

A、尽可能申请较少的计算资源，如节点数、CPU数等；

B、尽可能指定较短的运行时间（-t参数），便于回填成功。

3、如何查看作业优先级

（1）查看作业优先级有如下几种方法：

A、通过squeue格式化字段查看作业优先级

设置squeue格式化输出变量为：

export SQUEUE_FORMAT="%.18i %.9P %.8j %.8u %.2t %.10M %.6D %.8Q %R"

相对于squeue标准输出格式，新增了PRIORITY字段。

这时squeue输出示例如下：

```
[root@admin08 ~]# squeue -t PD
             JOBID PARTITION     NAME     USER ST       TIME  NODES PRIORITY NODELIST(REASON)
           1616615 full_part     bash    licom PD       0:00    800     1005 (Resources)
1616024_[38-39,71- full_part  64I_cpu   ybyang PD       0:00    128     1001 (Priority)
[root@admin08 ~]#
```

4、如何查看和修改作业优先级

B、通过sprio查看排队作业的优先级

执行命令sprio（或sprio -l）可以查看作业优先级。

```
[root@admin08 ~]# sprio -j  1616615
             JOBID PARTITION   PRIORITY       SITE  FAIRSHARE  PARTITION        QOS
           1616615 full_part       1005          0          6       1000          0
[root@admin08 ~]#
[root@admin08 ~]# sprio -l -j  1616615
             JOBID PARTITION     USER   PRIORITY       SITE        AGE      ASSOC  FAIRSHARE    JOBSIZE  PARTITION        QOS       NICE                TRES
           1616615 full_part    licom       1005          0          0          0          6          0       1000          0          0
[root@admin08 ~]#
```

5、如何在一个SLURM脚本中运行多个作业。

答：

SLURM支持作业（job）和作业步（step）两层的资源调度。

作业步通过srun启动。可以在同一个作业脚本中并行或串行执行多个作业步。作业步运行有如下的限制条件：

（1）任何作业步申请的资源，不能超过作业申请的总资源数；

（2）作业步只能使用其它作业步未使用的剩余作业资源；

（3）并发运行的作业步的资源之和不能超过作业申请的总资源数。

多作业步示例：

5、如何在一个SLURM脚本中运行多个作业。

答：多作业步示例：

```
#!/bin/bash
#SBATCH -o log/%j
#SBATCH -J ZHANGTAO
#SBATCH -N 2
#SBATCH --tasks-per-node=32
#SBATCH -p debug
#SBATCH --mem-per-cpu=3G

# Job Step0,all job resources will be used for the first job ste
time srun    --mpi=pmix_v3 ./open_fire_v7 500000

# Job step1 and job step 2 run in parallel and background, one
step for each node
time srun -n32 -N1 --mpi=pmix_v3 ./open_fire_v7 250000 &
time srun -n32 -N1 --mpi=pmix_v3 ./open_fire_v7 250000 &
# "wait" must be added to wait for job step 1 and job step 2 to
finish
wait

# Job Step 3, all job resources will be used for the third job
step
time srun    --mpi=pmix_v3 ./open_fire_v7 500000
```

5、如何在一个SLURM脚本中运行多个作业。

答：多作业步示例：

```
#!/bin/bash
#SBATCH -o log/%j
#SBATCH -J ZHANGTAO
#SBATCH -N 2
#SBATCH --tasks-per-node=32
#SBATCH -p debug
#SBATCH --mem-per-cpu=3G

# Job Step0,all job resources will be used for the first job ste
time srun    --mpi=pmix_v3 ./open_fire_v7 500000

# Job step1 and job step 2 run in parallel and background, one
step for each node
time srun -n32 -N1 --mpi=pmix_v3 ./open_fire_v7 250000 &
time srun -n32 -N1 --mpi=pmix_v3 ./open_fire_v7 250000 &
# "wait" must be added to wait for job step 1 and job step 2 to
finish
wait

# Job Step 3, all job resources will be used for the third job
step
time srun    --mpi=pmix_v3 ./open_fire_v7 500000
```

上述作业中包含 4 个作业步（0-3），分三个阶段运行：

第一阶段：

0 号作业步单独运行，使用全部作业资源（2 个节点）；

第二阶段：

1 号和 2 号作业步同时运行，各占用 1 个节点；

第三阶段：

3 号作业步单独运行，使用全部作业资源（2 个节点）。

通过sacct可以追溯运行过程，如下图所示：

```
[slurmtest@e13r1n01 test]$ sacct -j 10228 -o JobID,JobName,Partition,Account,AllocCPUS,Start,End,Nodelist,ExitCode
       JobID    JobName  Partition    Account  AllocCPUS               Start                 End        NodeList ExitCode
------------ ---------- ---------- --------- ---------- ------------------- ------------------- --------------- --------
10228          ZHANGTAO      debug  slurmtest         64 2020-04-25T11:39:16 2020-04-25T11:43:26   e13r4n[07-08]      0:0
10228.batch        batch             slurmtest         32 2020-04-25T11:39:16 2020-04-25T11:43:26       e13r4n07      0:0
10228.extern       extern            slurmtest         64 2020-04-25T11:39:16 2020-04-25T11:43:26   e13r4n[07-08]      0:0
10228.0       open_fire+             slurmtest         64 2020-04-25T11:39:17 2020-04-25T11:40:18   e13r4n[07-08]      0:0
10228.1       open_fire+             slurmtest         32 2020-04-25T11:40:18 2020-04-25T11:42:17       e13r4n07      0:0
10228.2       open_fire+             slurmtest         32 2020-04-25T11:40:18 2020-04-25T11:42:21       e13r4n08      0:0
10228.3       open_fire+             slurmtest         64 2020-04-25T11:42:21 2020-04-25T11:43:26   e13r4n[07-08]      0:0
[slurmtest@e13r1n01 test]$
[slurmtest@e13r1n01 test]$
```

6、提交作业后，提示“Failed to allocate resources: User’s group not permitted to use this partition”。

答：用户提交作业时通常需要加“-p 分区名”，这一参数，同时该参数应写在程序名前，并可用sinfo来查看所在

7、采用srun提交作业，关闭界面后，再次登录时发现作业被killed。

答：srun是交互式提交作业模式，一旦作业提交的界面关闭作业就会被killed。若需要较长时间运行的作业，建议用户采用sbatch批处理提交方式。sbatch负责资源分配，获取资源后会在获取资源的第一个节点运行提交的脚本，当前登录shell断开后，加载作业仍可正常运行。

8、salloc分配资源，退出salloc后发现作业断掉。

答：salloc与sbatch最主要的区别是，salloc命令资源请求被满足时，直接在提交作业的节点执行相应任务，适合需要指定运行节点和其他资源限制，并有特定命令的作业。当前shell断开后，申请获得的资源以及加载作业任务会退出。

9、为什么作业节点处于comp(COMPLETING)状态

答：当一个作业的应用进程终止时，该作业和它的节点都进入COMPLETING(CG)状态。当每个节点上的Slurm守护进程确定所有与作业相关的进程（如epilog脚本、slurmstepd）都已经结束时，该节点将状态变为IDLE或其它适当的状态，供其他作业使用。同时，该作业就会将状态变为COMPLETED或其他适当的状态（如FAILED）。通常情况下，这发生在一秒钟之内。

常见问题

如果出现节点和作业长期处于comp状态，可能的原因和处理方法如下：

（1）系统配置了Epilog脚本

如果SLURM系统配置了Epilog参数，则作业进程结束时会执行该参数配置的脚本。这期间的作业状态为CG，节点状态为comp。如果脚本运行需要运行很长时间，则节点状态会一直保持comp状态。处理方法：A、杀掉执行的Epilog脚本进程，此操作可能导致计算节点下线（drain）；B、手动上线节点：scontrol update node=<node name> state=resume。

（2）某些计算节点有问题，导致计算节点与管理节点通信异常问题原因包括：

A、环境配置问题，典型问题包括hosts不完整、系统时间不一致、slurm配置文件不一致。此时，出问题的节点可能是comp节点本身，也可能是其它参与消息转发的其它计算节点。解决方法：统一检查所有节点的上述配置，并对出问题的节点进行修正，或者关闭该节点的slurmd服务。

B、网络设备异常，或者配置异常。解决方法：结合系统日志和其它工具，定位问题并修正。

（3）节点上作业有些进程不能自己退出，也不能kill掉

这种情况的典型场景是共享存储异常卡死。可以执行以下命令恢复节点状态，使得作业完全退出。具体步骤：

A、设置节点为DOWN状态scontrol update nodename=<node name> state=down reason=comp

B、尝试手工处理残留的作业进程，如果失败则需要重启节点；

C、恢复节点状态，重启slurmd服务scontrol update nodename=<node name> state=resume

systemctl restsart slurmd

10、怎样才能暂时阻止一个job的运行（例如将job状态置于保留状态）。

最简单的方法是改变job的最早开始时间（可选择在作业提交时使用—begin选项设置）。下面的例子将作业置于保留状态（阻止其启动30天），然后允许其开始。

$ scontrol update JobId=1234 StartTime=now+30days

... later ...

$ scontrol update JobId=1234 StartTime=now

11、处于完成或失败状态的job如何重新排队

Slurm支持重新安排处于完成或失败状态的job，可以使用命令：

scontrol requeue job_id

然后，该job将被重新排队，回到 PENDING 状态

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

Sugon

携手成就梦想