北京超级云计算中心 A 分区
简明使用手册

The Right Answer in Data Center PARATERA 并行®

（2019 冬季版）

术语表

分区：对应文件系统上用来存储数据位置，可以理解为机器上的一个盘符；
队列：作业管理系统把系统的计算资源划分到不同的集合，用户的作业可以提交到这些集合排队运行，称这些集合为队列。

目录

1. 系统资源简介

1.1 计算资源（节点配置）

北京超级云计算中心 A 分区（以下简称 BSCC-A）的作业队列是 amd_256 队列，通过 sinfo 可以查看队列情况。amd_256 队列的计算节点配置是： AMD EPYC 7452，单节点 64 核，主频 2.35GHz，256GB 内存，节点间采用 ib 网互连。

1.2 存储资源（文件系统）

BSCC-A 系统的存储位于并行文件系统/public1 下面，登陆系统后通过 pwd 命令可以查看自己当前所在的分区，可以在此分区下进行编译软件，使用脚本提交作业。每个新开的账号默认存储是 500G，在家目录下执行“lfs quota -uh 超算账号 ~”即可查看使用的存储情况。用户数据占用的磁盘空间超过磁盘配额后会影响数据保存或者作业运行，建议您经常检查配额，及时清理备份不需要的数据。如果需要比较大的磁盘空间存储数据，可以联系客户经理增加配额。

2. 使用超算应用软件

2.1 简介

由于不同用户在“BSCC-A”上可能需要使用不同的软件环境，配置不同的环境变量，软件之间可能会相互影响，因而在“BSCC-A”上安装了 module 工具来对应用软件统一管理。module 工具主要用来帮助用户在使用软件前设置必要的环境变量。用户使用 module 加载相应版本的软件后，即可直接调用超算上已安装的软件。

2.2 基本命令

常用命令如下：

命令	功能	例子
module avail	查看可用的软件列表	
module load [modulesfile]	加载需要使用的软件	module load intel/19.0.3-zyq
module show [modulesfile]	查看对应软件的环境（安装路径、库路径等）	module show intel/19.0.3-zyq
module list	查看当前已加载的所有软件	
module unload [modulesfile]	移除使用 module 加载的软件环境	module unload intel/19.0.3-zyq

module 其它用法，可使用 module --help 中查询。module 加载的软件环境只在当前登陆窗口有效，退出登陆后软件环境就会失效。用户如果需要经常使用一个软件，可以把 load 命令放在~/.bashrc 或者提交脚本里面。

3. Slurm 作业管理系统

BSCC-A 使用 slurm 作业管理系统，采用节点独占模式，amd_256 队列每个节点 64 核，为避免浪费机时，使用时请尽量保证满核提交（即为单节点核数的整数倍）。

作业管理系统常用命令如下：

命令	功能介绍	常用命令例子
sinfo	显示系统资源使用情况	sinfo
squeue	显示作业状态	squeue
srun	用于交互式作业提交	srun -N 2 -n 128 -p amd_256 A.exe
sbatch	用于批处理作业提交	sbatch -N 2 -n 128 job.sh
salloc	用于分配模式作业提交	salloc -p amd_256
scancel	用于取消已提交的作业	scancel JOBID
scontrol	用于查询节点信息或正在运行的作业信息	scontrol show job JOBID
sacct	用于查看历史作业信息	sacct -u sc3001 -S 03/01/17 -E 03/31/17 --field=jobid,partition,jobname,user,nnodes,start,end,elapsed,state

3.1 sinfo 查看系统资源

sinfo 得到的结果是当前账号可使用的队列资源信息，如下图所示：

```
[deploy@ln21%bscc-a 2.2.1]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
all       inact   infinite     15 drain* j[2201-2202,2210,2301-2302,2401-2403,2405,2408,2414,2601-2602,2608,2613]
all       inact   infinite    175  down* j[1501-1516,1601-1616,1701-1716,1801-1816,1901-1916,2003,2005-2007,2101-2116,2203-2209,2211-2216,2
303-2316,2404,2406-2407,2409-2410,2412-2413,2415-2416,2501-2516,2603-2607,2609-2612,2614-2616,2701-2706,2709,2713-2716]
all       inact   infinite      2  drain j[2001-2002]
all       inact   infinite      7   down j[2004,2411,2707-2708,2710-2712]
amd_256*     up   infinite     15 drain* j[2201-2202,2210,2301-2302,2401-2403,2405,2408,2414,2601-2602,2608,2613]
amd_256*     up   infinite    175  down* j[1501-1516,1601-1616,1701-1716,1801-1816,1901-1916,2003,2005-2007,2101-2116,2203-2209,2211-2216,2
303-2316,2404,2406-2407,2409-2410,2412-2413,2415-2416,2501-2516,2603-2607,2609-2612,2614-2616,2701-2706,2709,2713-2716]
amd_256*     up   infinite      2  drain j[2001-2002]
amd_256*     up   infinite      7   down j[2004,2411,2707-2708,2710-2712]
```

其中，

第一列 PARTITION 是队列名。

第二列 AVAIL 是队列可用情况，如果显示 up 则是可用状态；如果是 inact 则是不可用状态。

第三列 TIMELIMIT 是作业运行时间限制，默认是 infinite 没有限制。

第四列 NODES 是节点数。

第五列 STATE 是节点状态，idle 是空闲节点，alloc 是已被占用节点，comp 是正在释放资源的节点，其他状态的节点都不可用。

第六列 NODELIST 是节点列表。

sinfo 的常用命令选项：

命令示例	功能
sinfo -n j12345	指定显示节点 j12345 的使用情况
sinfo -p amd_256	指定显示队列 amd_256 情况

其他选项可以通过 sinfo --help 查询

3.2 squeue 查看作业状态

squeue 得到的结果是当前账号的作业运行状态，如果 squeue 没有作业信息，说明作业已退出。

具体示例见下图：

```
[deploy@ln1%cstc9 ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            124828     v3_64 300010.s   deploy PD       0:00      6 (Resources)
[deploy@ln1%cstc9 ~]$
```

其中，

第一列 JOBID 是作业号，作业号是唯一的。

第二列 PARTITION 是作业运行使用的队列名。

第三列 NAME 是作业名。

第四列 USER 是超算账号名。

第五列 ST 是作业状态，R 表示正常运行，PD 表示在排队，CG 表示正在退出，S 是管理员暂时挂起，只有 R 状态会计费。

第六列 TIME 是作业运行时间。

第七列 NODES 是作业使用的节点数。

第八列 NODELIST(REASON)对于运行作业（R 状态）显示作业使用的节点列表；对于排队作业（PD 状态），显示排队的原因。

squeue 的 常用命令选项：

命令示例	功能
squeue -j 123456	查看作业号为 123456 的作业信息
squeue -u sc30001	查看超算账号为 sc30001 的作业信息
squeue -p amd_256	查看提交到 amd_256 队列的作业信息
squeue -w j123	查看使用到 j123 节点的作业信息

其他选项可通过 squeue --help 命令查看

3.3 srun 交互式提交作业

srun [options] program 命令属于交互式提交作业，有屏幕输出，但容易受网络波动影响，断网或关闭窗口会导致作业中断。

srun 命令示例：

```
srun -p amd_256 -w j[1100-1101] -N 2 -n 128 -t 20 A.exe
```

交互式提交 A.exe 程序。如果不关心节点和时间限制，可简写为 srun -p amd_256 -n 128 A.exe

其中，

-p amd_256 指定提交作业到 amd_256 队列；

-w j[1100-1101] 指定使用节点 j[1100-1101]；

-N 2 指定使用 2 个节点；

-n 128 指定进程数为 128，BSCC-A 超算 amd_256 队列每一个节点 64 核，建议使用单节点核数的整数倍提交作业；

-t 20 指定作业运行时间限制为 20 分钟。

srun 的一些常用命令选项：

参数选项	功能
-N 3	指定节点数为 3
-n 64	指定进程数为 64
-c 64	指定每个进程（任务）使用的 CPU 核为 64
-p amd_256	指定提交作业到 amd_256 队列
-w j[100-101]	指定提交作业到 j100、j101 节点
-x j[100,106]	排除使用 j100、j106 节点
-o out.log	指定标准输出到 out.log 文件
-e err.log	指定重定向错误输出到 err.log 文件
-J JOBNAME	指定作业名为 JOBNAME
-t 20	限制运行 20 分钟

srun 的其他选项可通过 srun --help 查看。

3.4 sbatch 后台提交作业

sbatch 一般情况下与 srun 一起提交作业到后台，需要将 srun 写到脚本中，再用 sbatch 提交脚本。这种方式不受本地网络波动影响，提交作业后可以关闭本地电脑。sbatch 命令没有屏幕输出，默认输出日志为提交目录下的 slurm-xxx.out 文件，可以使用 tail -f slurm-xxx.out 实时查看日志，其中 xxx 为作业号。

sbatch 命令示例 1（64 个进程提交 A.exe 程序）：

编写脚本 job1.sh，内容如下：

```
#!/bin/bash
srun -n 64 A.exe
```

然后在命令行执行 sbatch -p amd_256 job1.sh 提交作业。脚本中的#!/bin/bash 是 bash 脚本的固定格式。从脚本的形式可以看出，提交脚本是一个 shell 脚本，因此常用的 shell 脚本语法都可以使用。作业开始运行后，在提交目录会生成一个 slurm-xxx.out 日志文件，其中 xxx 表示作业号。

sbatch 命令示例 2（指定 2 个节点，4 个进程，每个进程 12 个 cpu 核提交 A.exe 程序，限制运行 60 分钟）：

编写脚本 job2.sh，内容如下：

```
!/bin/bash

#SBATCH -N 2

#SBATCH -n 4

#SBATCH -c 12

#SBATCH -t 60

srun -n 4 A.exe
```

然后在命令行执行 sbatch -p amd_256 job2.sh 就可以提交作业。其中#SBATCH 注释行是 slurm 定义的作业执行方式说明，一些需要通过命令行指定的设置可以通过这些说明写在脚本里，避免了每次提交作业写很长的命令行。

sbatch 命令示例 3（单节点提交多任务）

编写脚本 job3.sh，内容如下：

```
#!/bin/bash
srun -n 6 A.exe &
srun -n 6 B.exe &
srun -n 6 C.exe &
srun -n 6 C.exe
wait
```

然后在命令行执行 sbatch -N 1 -p amd_256 job3.sh，这里是单节点同时提交 4 个任务，每个任务使用 6 个进程。这里需要 4 个任务同时运行，并且全部执行完毕，作业才会退出。

sbatch 的一些常用命令选项基本与 srun 的相同，具体可以通过 sbatch --help 查看。

3.5 salloc 分配模式作业提交

salloc 命令用于申请节点资源，一般用法如下：

```
1、执行 salloc -p amd_256；
2、执行 squeue 查看分配到的节点资源，比如分配到 j100；
3、执行 ssh j100 登陆到所分配的节点；
4、登陆节点后可以执行需要的提交命令或程序；
5、作业结束后，执行 scancel JOBID 释放分配模式作业的节点资源。
```

3.6 scancel 取消已提交的作业

scancel 可以取消正在运行或排队的作业。

scancel 的一些常用命令示例：

命令示例	功能
scancel 123456	取消作业号为 123456 的作业
scancel -n testjob	取消作业名为 testjob 的作业
scancel -p amd_256	取消提交到 amd_256 队列的作业
scancel -t PENDING	取消正在排队的作业
scancel -w j100	取消运行在 j100 节点上的作业

scancel 的其他参数选项，可通过 scancel --help 查看

3.7 scontrol 查看正在运行的作业信息

scontrol 命令可以查看正在运行的作业详情，比如提交目录、提交脚本、使用核数情况等，对已退出的作业无效。

scontrol 的常用示例：

```
scontrol show job 123456
```

查看作业号为 123456 的作业详情。

scontrol 的其他参数选项，可通过 scontrol --help 查看。

3.8 sacct 查看历史作业信息

sacct 命令可以查看历史作业的起止时间、结束状态、作业号、作业名、使用的节点数、节点列表、运行时间等。

sacct 的常用命令示例：

```
sacct -u sc30001 -S 2017-09-01 -E now --field=jobid,partition,jobname,user,nnodes,nodelist,start,end,elapsed,state
```

其中，-u sc30001 是指查看 sc30001 账号的历史作业，-S 是开始查询时间，-E 是截止查询时间，--format 定义了输出的格式，jobid 是指作业号，partition 是指提交队列，user 是指超算账号名，nnodes 是节点数，nodelist 是节点列表，start 是开始运行时间，end 是作业退出时间，elapsed 是运行时间，state 是作业结束状态。sacct --helpformat 可以查看支持的输出格式。

sacct 的其他参数选项可通过 sacct --help 查看。

温馨提示：

登录节点仅供编译软件、拷贝数据，为避免登录节点负载过高，影响正常使用，请勿在登录节点运行程序。提交作业请使用调度命令发送到计算节点，请勿在登录节点运行作业，如有发现此类不规范操作，管理员将终止进程。并通知违规用户，两次警告无效后，每次违规操作，将会禁止登录账号登录一天。

4 编译器

BSCC-A 已配置 AOCC、GNU 和 Intel 编译器，支持 C、C++、Fortran77 和 Fortran90 语言程序的开发，支持 OpenMP 和 MPI 两种并行编程模式。其中 OpenMP 为共享内存方式，只能单点并行；MPI 是分布式内存并行，支持跨节点并行。

4.1 Intel 编译器

BSCC-A 可通过 module load 加载 intel 编译环境，例如 module load intel/19.0.3-zyq，如下图 1 所示：

```
[deploy@ln21%bscc-a 2.2.1]$ module avail intel

------------------------------------------------------- /public1/soft/modulefiles -------------------------------------------------------
intel/19.0.3-zyq
[deploy@ln21%bscc-a 2.2.1]$
```

图 1 module 使用示意图

通过“which”命令可以查找命令所在路径，例如“which icc”；通过“icc -v”命令可以查询 icc 的版本。Intel 编译器的详细命令行调用则可以用“icc --help”获得。

用户经常需要使用 MKL 库，通过命令 echo $MKLROOT 可以查看 MKLROOT 环境变量确认 MKL 库的位置。

4.2GCC 编译器

BSCC-A 默认的 GNU 编译器版本是 4.8.5

4.3 MPI 编译环境

BSCC-A 使用的 MPI 编译环境必须通过 module load 加载。例如 module load mpi/intel/19.0.3-zyq

温馨提示：

- BSCC-A 上部署了一些开源软件，比如 lammps 等软件，可执行 module avail 查看，module load 直接调用。
- BSCC-A 可以连外网，您可以连网安装一些软件。