

超算云 Q&A 汇总

北京超级云计算中心
北京北龙超级云计算有限责任公司

2023 年 5 月 16 日

目录

1. slurm 作业调度系统常见命令汇总.....	1
1.1. sinfo 查看系统资源.....	1
1.2. squeue 查看作业状态.....	2
1.3. srun 交互式提交作业.....	3
1.4. sbatch 后台提交作业.....	4
1.5. salloc 分配模式作业提交.....	7
1.6. scancel 取消已提交的作业.....	7
1.7. scontrol 查看正在运行的作业信息.....	7
1.8. sacct 查看历史作业信息.....	8
2. module 的使用和介绍.....	9
2.1. 简介.....	9
2.2. 基本命令.....	9
2.3. module 加载编译器示例.....	10
2.4. module 加载开源软件示例.....	11
3. 按节点计费资源和按核计费资源使用注意事项.....	13
3.1. 按节点计费资源使用注意事项.....	13
3.1.1. 按节点计费分区.....	13
3.1.2. 注意事项.....	13
3.2. 按核计费资源使用注意事项.....	13
3.2.1. 按核计费分区.....	13
3.2.2. 独占节点参数.....	14
3.2.3. 单节点多任务.....	14

1. slurm 作业调度系统常见命令汇总

超算使用 slurm 作业管理系统，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
srund	用于交互式作业提交	srund -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 \$USER -S 2023-05-01 -E 2023-05-31 --field=jobid,partition,jobname, user,nnodes,nodelist,start,end,elapsed,state -T -X

1.1. 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：

```
[blsc~]sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
all        inact    infinite   5 drain* hb2004,ja[0315,0410,1010,1115]
all        inact    infinite   1 down* ja1207
all        inact    infinite   1 dmgp ja1507
all        inact    infinite   29 drain qa[1215,1303,1307,1411,2002],qb[0208,0403,0606,0613,0615,0714-0716,0802,0804,0806-0807,1006],hb[2103,2201],ja[0110,0207-0208,0307,0714,1210,1613,1801,1908]
all        inact    infinite  1092 alloc qa[0303-0304,0306-0316,0403,0405-0406,0408-0416,0502,0506-0507,0513,0515-0516,0606-0607,0610-0616,0702-0703,0707,0709-0716,0801-0802,0906-0810,0812-0816,0902-0903,0909-0916,1001-1009,1012-1016,1101-1106,1108-1112,1114-1116,1201-1214,1216,1301-1302,1304-1306,1308-1316,1401-1410,1412-1416,1501-1516,1601-1610,1612-1613,1615-1616,1701-1716,1801-1816,1901-1915,2001-2003,2013,2015-2016,2101-2116,2201-2209,2213-2216,2301-2305,2307-2315,2401-2416,2501-2516,2601-2612,2615-2616,2701-2713,2716],qb[0101-0103,0106-0108,0201-0207,0209-0216,0301-0306,0308,0310-0315,0401,0406-0416,0501,0503-0505,0507-0516,0601-0602,0604-0605,0607-0612,0616,0701-0702,0704-0713,0801,0805,0808-0809,0811,0813-0816,0901,0903-0906,0908-0915,1003,1005,1009-1016,1101-1103,1105-1106,1108-1112,1114-1116,1201-1204,1209-1216,1301,1508-1512,1514-1516,1606-1613,1616,1701],gc[0601-0607,0609-0610,0612-0616,0701-0705,0801-0806,0808-0816,0901-0916,1001-1016,1101-1116,1201-1216,1301-1316,1401-1416,1501-1516,1601-1616],hd[1808,1902],hb[1908-1910,1912-1913,2001-2003,2005-2010,2012-2013,2101-2102,2104-2113,2115,2202-2213,2305-2311,2401-2413,2501-2512,2601-2606,2609,2611-2613,2701,2706-2709,2712],ja[0101,0103,0105-0109,0111-0116,0201-0206,0209-0216,0301-0306,0308-0314,0316,0401-0409,0411-0416,0501-0507,0509-0516,0601-0608,0610,0612,0615-0616,0701-0713,0715-0716,0802-0816,0901-0916,1001-1009,1011-1016,1101-1114,1116,1201-1206,1208-1209,1211-1216,1501-1506,1508-1509,1513-1516,1601-1608,1610-1612,1614-1616,1701-1716,1803-1809,1811,1814-1816,1901-1907,1909-1912,1914-1916,2001-2016,2101-2108,2110-2113,2115-2116,2201-2203,2205-2216,2304-2305,2307-2308,2310-2316,2401-2407,2409-2416,2501-2507,2510-2516,2605-2616,2701-2702,2705,2710,2714-2715]
all        inact    infinite   159 idle qa[0301-0302,0305,0401-0402,0404,0407,0501,0503-0505,0508-0512,0514,0601-0605,0608,0701,0704-0706,0708,0803-0805,0901,0904,1010-1011,1107,1113,1611,1614,2210-2212,2306,2316,2613-2614,2714-2715],qb[0104-0105,0307,0309,0316,0402,0404-0405,0502,0506,0603,0614,0703,0803,0812,0902,0907,0916,1001-1002,1004,1007-1008,1104,1107,1113,1205-1208,1302-1304,1501-1507,1513,1601-1603,1614-1615,1702-1703],gc[0608,0611,0807],hb[1911,2011,2112,2301-2304,2312-2313,2607-2608,2610,2702-2703,2709-2711,2713],ja[0102,0104,0508,0609,0611,0613-0614,0801,1510-1512,1609,1802,1810,1812-1813,1913,1919,2114,2204,2301-2303,2306,2309,2409,2508-2509,2601-2604,2703-2704,2706-2709,2711-2713,2716]
all        inact    infinite   6 down qa[0609,0811,1916,2014],qb0810,hb2513
amd_256*   up        infinite   5 drain* hb2004,ja[0315,0410,1010,1115]
amd_256*   up        infinite   1 down* ja1207
amd_256*   up        infinite   29 drain qa[1215,1303,1307,1411,2002],qb[0208,0403,0606,0613,0615,0714-0716,0802,0804,0806-0807,1006],hb[2103,2201],ja[0110,0207-0208,0307,0714,1210,1613,1801,1908]
amd_256*   up        infinite   992 alloc qa[0309-0310,0313-0315,0409-0414,0416,0507,0513,0515-0516,0610-0616,0707,0709-0716,0801-0802,0806,0809-0810,0812-0816,0902-0903,0905-0916,1001-1009,1012-1016,1101-1105,1108-1112,1114-1116,1201-1206,1210-1214,1216,1301-1302,1304-1306,1308-1316,1401-1410,1412-1416,1501-1516,1601-1610,1612-1613,1616,1701-1716,1801-1816,1901-1915,2001-2003,2005-2010,2012-2013,2015-2016,2101-2113,2115,2216,2301-2305,2307-2315,2401-2416,2501-2516,2601-2612,2615-2616,2701-2704,2706-2709,2712-2713,2716],qb[0101-0103,0106-0108,0201-0206,0209-0216,0301-0306,0308,0310-0311,0401,0406-0416,0501,0503-0505,0507,0509-0516,0601-0602,0604-0605,0607-0612,0616,0701-0702,0704-0713,0801,0805,0808-0809,0811,0813-0814,0816,0901,0904-0906,0908-0915,1003,1005,1009-1011,1014-1016,1101-1103,1105-1106,1108-1112,1114-1116,1201-1204,1209-1211,1214-1216,1301,1508-1512,1514-1516,1606-1610,1613-1616,1701],gc[0601-0607,0609-0610,0612-0616,0701-0705,0801-0806,0808-0816,0901-0916,1001-1016,1101-1116,1201-1216,1301-1316,1401-1416,1501-1516,1604-1612,1614-1616],hd[1808,1902],hb[1908-1910,1912-1913,2001-2003,2005-2010,2012-2013,2015-2016,2101-2113,2115,2202-2213,2305-2311,2401-2413,2501-2512,2601-2606,2609,2611-2613,2701,2706-2709,2712],ja[0101,0103,0201,0205,0209,0211,0215,0301-0304,0306,0309-0311,0313-0314,0401-0402,0404,0406-0409,0411-0416,0503-0507,0509-0512,0514-0516,0601-0602,0604-0605,0607-0703,0715-0716,0802-0807,0809-0816,0901-0904,0906-0907,0909-0910,0913-0916,1001-1009,1011,1013-1016,1101-1112,1116,1203-1206,1208-1209,1211,1214-1216,1501-1503,1508-1509,1513-1516,1601-1608,1610-1612,1614-1616,1701-1716,1803-1806,1808-1809,1811,1814-1816,1901-1907,1909-1911,1914-1916,2001,2003,2005-2016,2101-2109,2110-2113,2115-2116,2201-2203,2205-2216,2305,2307-2308,2310-2316,2401-2407,2409-2410,2412-2416,2501-2507,2510-2512,2514-2516,2605-2616,2701-2702,2705,2710,2714-2715]
amd_256*   up        infinite   60 idle qa[0314,1107,2315],qb[0307,0316,0402,0404-0405,0502,0603,0614,0703,0803,0902,0907,1001-1002,1004,1007-1008,1104,1107,1113,1205-1208,1302,1501-1507,1601-1605,1702],gc[0608,0611,0807],hb[1911,2011,2112,2312-2313,2702-2703,2709-2711,2713],ja[1802,1913,2303,2306,2408]
amd_256*   up        infinite   6 down qa[0609,0811,1916,2014],qb0810,hb2513
[blsc~]
```

图 1 sinfo 使用示意图

其中，

第一列 PARTITION 是队列名，如 amd_256

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

1.2. 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：

```
[blsc~]squeue
      JOBID PARTITION   NAME     USER ST       TIME  NODES NODELIST(REASON)
      2812463   amd_256 interact scb2483 R        0:27     1 ga1714
[blsc~]
```

图 2 squeue 使用示意图

其中，

第一列 JOBID 是作业号，作业号是唯一的，如截图里的 4714412。

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

1.3. 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。

1.4. 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。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, 脚本内容详细解释如下 (以队列 amd_256 为例) :

```
#!/bin/bash

#SBATCH -p amd_256 (指定使用队列, 执行 sinfo 可查看, 按需修改)
#SBATCH -N 1 (指定节点数, 按需修改)
#SBATCH -n 64 (指定进程数, 按核调度分区请按需修改; 节点独占分区建议满核提交, 每个分区单节点核数可从手册中找到)
#SBATCH -c 1 (指定线程数, 不设置则默认为 1)
#SBATCH -t 60 (设置运行限制时间, 非必须)
加载编译环境命令, 如 module 相关命令)
srun -n 64 A.exe (运行程序命令部分, 按需修改)
```

然后在命令行执行 `sbatch 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。

编写脚本 `job2.sh`, 提交运行程序为 `A.exe` 作业脚本模板如下:

```
#!/bin/bash
#SBATCH -p amd_256
#SBATCH -N 1
#SBATCH -n 64
source /public1/soft/modules/module.sh
module load mpi/intel/19.3.0 (编译环境)
export PATH=/path/to/bin (指定程序 A.exe 执行程序路径)
srun A.exe
```

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 2 (指定 2 个节点, 4 个进程, 每个进程 32 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 -p amd_256
#SBATCH -N 2
#SBATCH -n 4
#SBATCH -c 32
#SBATCH -t 60
```

```
srun -n 128 A.exe
```

然后在命令行执行 `sbatch job2.sh` 就可以提交作业。

`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
#SBATCH -p 队列
#SBATCH -N 1
#SBATCH -n 核数
cd 第一个作业的路径
srun -n 8 (以用 8 核为例) xxx &
sleep 20
cd 第二个作业的路径
srun -n 8 xxx &
sleep 20
cd 第三个作业的路径
srun -n 8 xxx
wait
```

然后在命令行执行 `sbatch job3.sh`, 这里是单节点同时提交 3 个任务, 每个任务使用 8 个进程。这里 3 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。

温馨提示:

- (1) 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。
- (2) 按核调度的资源分区, 请按需修改使用的核数; 节点独占模式的资源分区, 请满核提交, 避免造成核时浪费。
- (3) 需指定线程数运行的软件, 如 `python`, `R`, `matlab`, `gaussian` 等, 此时的脚本申请资源的写法应改为:

```
#SBATCH -p 队列
#SBATCH -N 1
#SBATCH -n 1
#SBATCH -c 64(具体核数)
```

1.5. 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` 可以抢占一个节点；
- (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。

1.6. 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
<code>scancel 123456</code>	取消作业号为 123456 的作业
<code>scancel -n testjob</code>	取消作业名为 testjob 的作业
<code>scancel -p amd_256</code>	取消提交到 amd_256 队列的作业
<code>scancel -t PENDING</code>	取消正在排队的作业
<code>scancel -w j100</code>	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

1.7. scontrol 查看正在运行的作业信息

scontrol 命令可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。

1.8. sacct 查看历史作业信息

sacct 命令可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 $USER -S 2023-05-01  
-E 2023-05-31 --field=jobid,partition,jobname,user,nnodes,nodelist,start,end,elapsed,state -T -X
```

其中, -u \$USER 是指查看当前账号的历史作业, -S 是开始查询时间, -E 是截止查询时间, --field 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。

2. module 的使用和介绍

2.1. 简介

由于不同用户在超算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，因而在超算上安装了 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/19.3.0
module show [modulesfile]	查看对应软件的环境（安装路径、库路径等）	module show intel/19.3.0
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/19.3.0
module purge	清理已加载的 module 环境	

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。

若有 module 需求的用户可根据自己所使用的存储 (pwd 查看) 执行对应的环境命令，如：public1 存储执行：

```
source /public1/soft/modules/module.sh
```

具体可参考登陆界面的通知。



图 3 登录界面示意图

如提示没有 module 命令，请根据登录界面信息，加载对应 module.sh 环境。

```
[blsc~]module list
bash: module: command not found...
[blsc~]source /public1/soft/modules/module.sh
[blsc~]module list
No Modulefiles Currently Loaded.
[blsc~]
```

图 4 加载 module 环境示意图

2.3. module 加载编译器示例

超算已配置 GNU 和 Intel 编译器和部分开源软件，可直接通过 module 来调用。

其中 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。

module 加载 mpi/intel 编译环境示例，例如 module load mpi/intel/19.3.0，如下图所示：

```
[blsc~]source /publicl/soft/modules/module.sh
[blsc~]module avail mpi
----- /publicl/soft/modulefiles -----
mpi/intel/5.0.3.049-intel15.0.6-thc  mpi/oneAPI/2022.1  mpi/openmpi/3.1.4-icc19-thc  mpich/3.1.4-gcc4.9.2-zyq
mpi/intel/17.0.7-thc  mpi/openmpi/1.5.5-gcc-new  mpi/openmpi/3.1.6-gcc9.3-fgl  mpich/3.1.4-gcc8.1.0
mpi/intel/18.0.2-thc  mpi/openmpi/1.5.5-gcc-thc  mpi/openmpi/4.1.0-gcc9.1-cjj  mpich/3.1.4-intel-17.0.7
mpi/intel/19.3.0  mpi/openmpi/1.8.8-cjj  mpi/openmpi/4.1.0-gcc9.1-fgl  mpich/3.1.4-gcc8.1.0-wjl
mpi/intel/20.0.1-lcc  mpi/openmpi/1.10.0-wzm  mpi/openmpi/4.1.0-intel20-para  mpich/3.2a2-gcc7.3-fgl
mpi/intel/20.0.4-ls  mpi/openmpi/2.0.4-gcc-4.9.0  mpi/openmpi/4.1.1-fgl  mpich/3.3.1
mpi/intel/2017.5  mpi/openmpi/3.1.2-gcc-test  mpi/openmpi/4.1.1-gcc7.3.0  mpich/3.3.1-icc18
mpi/intel/2021.1  mpi/openmpi/3.1.2-gcc-thc  mpi/openmpi/4.1.1-intel-17  mpich/3.4.2
mpi/intel/2021.2  mpi/openmpi/3.1.3-gcc-thc  mpi/openmpi/4.1.1-para  mpich/mpich2-1.1.1p1-icc18
mpi/intel/2022.1  mpi/openmpi/3.1.3-icc18-cjj  mpich/1.2.1
mpi/mvapich2/network-fenggl  mpi/openmpi/3.1.4-gcc-thc  mpich/3.1.3_fengjy
mpi/oneAPI/2021.2  mpi/openmpi/3.1.4-icc17-cjj  mpich/3.1.4-gcc4.9.2
[blsc~]module load mpi/intel/20.0.1-lcc
[blsc~]module list
Currently Loaded Modulefiles:
 1) intel/20.0.1-lcc  2) mpi/intel/20.0.1-lcc
```

图 5 module 加载编译器使用示意图

执行 module list 可以查看到已加载编译器包括 mpi/intel/19.3.0, intel/19.3.0, 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。

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。

2.4. module 加载开源软件示例

超算上也部署了一些开源软件, 比如 anaconda/python/R/lammps/Gromacs 等, 可执行 module avail xxx(xxx 为软件名, 区分大小写)查看, module load xxx(具体软件全称)直接调用。

```
[blsc~]source /publicl/soft/modules/module.sh
[blsc~]module avail anaconda
----- /publicl/soft/modulefiles -----
anaconda/2-201907-fenggl  anaconda/2-Python2.7.16-wx1  anaconda/3-Python3.7.3-wjl  anaconda/3-Python3.7.7-wjl
anaconda/2-Python2.7.16-fenggl  anaconda/3-deepmd-kit-lcc  anaconda/3-Python3.7.4-2019.10-wx1  anaconda/3-Python3.8.8-zsq
anaconda/2-Python2.7.16-fjy  anaconda/3-Python-3.8.3-phonopy-phonopy  anaconda/3-Python3.7.4-fenggl
anaconda/2-Python2.7.16-new-wx1  anaconda/3-Python3.6.5-fgl  anaconda/3-Python3.7.4-RSeQC-wx1
[blsc~]module load anaconda/3-Python3.6.5-fgl
[blsc~]module list
Currently Loaded Modulefiles:
 1) anaconda/3-Python3.6.5-fgl
[blsc~]python
Python 3.6.5 [Anaconda, Inc.] (default, Apr 29 2018, 16:14:56)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

图 6 module 加载开源软件示意图

module show 命令使用参考如下, 可以查看软件安装路径、库路径等。

```
[blsc~]module list
Currently Loaded Modulefiles:
 1) anaconda/3-Python3.6.5-fgl
[blsc~]module show anaconda/3-Python3.6.5-fgl
-----
/public1/soft/modulefiles/anaconda/3-Python3.6.5-fgl:

module-whatis   For more information, 'module help anaconda/3-python3.7.4'
conflict        anaconda
prepend-path    PATH /public1/soft/anaconda3/anaconda3-3.6.5/bin
setenv          ANACONDA_PATH /public1/soft/anaconda3/anaconda3-3.6.5/bin
prepend-path    LD_LIBRARY_PATH /public1/soft/anaconda3/anaconda3-3.6.5/lib
prepend-path    LD_LIBRARY_PATH /public1/soft/anaconda3/anaconda3-3.6.5/pkgs/icu-58.2-h9c2bf20_1/lib
prepend-path    INCLUDE /public1/soft/anaconda3/anaconda3-3.6.5/include
-----
[blsc~]
```

图 7 module show 示意图

module unload 命令使用参考如下，可以取消加载的具体某一个软件环境。

```
[blsc~]module list
Currently Loaded Modulefiles:
 1) anaconda/3-Python3.6.5-fgl
[blsc~]module unload anaconda/3-Python3.6.5-fgl
[blsc~]module list
No Modulefiles Currently Loaded.
[blsc~]
```

图 8 module unload 示意图

module purge 命令使用参考如下,清理掉原有加载的环境，提示 No Modulefiles Currently Loaded.即没有加载任何 module 部署软件环境。

```
[blsc~]module list
Currently Loaded Modulefiles:
 1) anaconda/3-Python3.6.5-fgl  2) intel/20.0.1-lcc  3) mpi/intel/20.0.1-lcc
[blsc~]module purge
[blsc~]module list
No Modulefiles Currently Loaded.
[blsc~]
```

图 9 module purge 示意图

3. 按节点计费资源和按核计费资源使用注意事项

3.1. 按节点计费资源使用注意事项

3.1.1. 按节点计费分区

BSCC-A (64 核), **BSCC-A2** (64 核), **BSCC-A3** (64 核), **BSCC-A5** (64 核),
BSCC-A6 (64 核), **BSCC-IO** (64 核), **CSTC-9** (24 核), **NC-L** (48 核), **NC-E** (48 核),
GAUNGZHOU (24、56 核), **吕梁** (24 核), **长沙** (12 核), **无锡** (24、56 核)。

3.1.2. 注意事项

节点独占的超算资源分区, 计费方式是以节点为基本单位。

即: 如果您提交核数不足节点上的核数的话, 那么计费也是按满核计费。

比如: 超算上每个节点有 64 核资源, 您提交 12 核作业的话, 也是按照 64 核计费。所以建议您满核提交, 使用的核数要用一个节点核数的整数倍。

以一个节点是 64 核的超算为例, 脚本里面 **#SBATCH -n** 后面的数字要是节点数乘以每个节点的核数, 才是充分利用资源。比如您用 1 个节点, 就是下面写法:

```
#SBATCH -N 1
```

```
#SBATCH -n 64
```

您在提交作业的时候留意下您脚本写法, 确保是满核提交, 充分利用资源。

3.2. 按核计费资源使用注意事项

3.2.1. 按核计费分区

BSCC-M (64 核), **BSCC-T** (96 核), **BSCC-T6** (96 核), **JN** (56 核), **ZC-M6** (128 核),
ZKXD (32 核), **CC-E6** (104 核), **深圳** (12 核)。

3.2.2. 独占节点参数

按核调度的超算资源分区，是按核收费的，脚本中 n 后的数字，是总核数，请根据实际需要，更改 n 后的数字，来调整算例计算所用核数。

如果脚本里面有 `#SBATCH --exclusive` 参数，是指独占该节点，将申请整个节点核数，计费也将按照独占节点计费。如果您想按核计费，则不需要使用此参数。

3.2.3. 单节点多任务

按核计费资源不推荐单节点多任务，因为单节点多任务的脚本中设有 sleep 时间，是两个任务的间隔时间。这个时间也会计算核时。

如果是大批量提交作业，会有浪费核时的潜在风险。

请根据计算量直接在脚本中设置合理的核数。