北京超级云计算中心 N22 区

GPU 资源用户使用手册

2021 年 4 月

目录

1 系统资源简介

1.1 计算资源（节点配置）

登陆节点：主机名 ln01（16 核 32G），用于提供用户登陆服务。请不要在登录节点直接运行作业（日常查看、编辑、编译等除外），以免影响其余用户的正常使用。

GPU 计算节点：主机命名规则：g0001-gxxxx，可参考 2.3 节内容申请 GPU 计算资源进行计算任务提交。

BSCC-N22 的 gpu 队列配置为： 每台机器配置 8 块型号为 NVIDIA Tesla V100-SXM2 32GB 显存的 GPU，每块 GPU 卡默认分配 8 CPU 核及 36GB 内存，即 GPU：CPU：内存配比为 1GPU 卡：8CPU 核：36GB 内存。如果 CPU、内存需求用量超出该配比可以增加 CPU 核数的申请，也可以通过增加 GPU 卡数或 CPU 核数的申请获得更多的内存配额，CPU：内存配额比例为 1：4.5GB。

登录节点可以联网，计算节点不能联网。如果程序运行过程需要下载数据请在登录节点下载好数据再提交作业到计算节点。如果有特殊需求请联系客服。

单作业允许申请 1~8 卡运行。gpu 分区不允许提交纯 CPU 作业。有特殊需求请与管理员联系开放 CPU 资源。

1.2 存储资源（文件系统）

BSCC-N22 根据数据的使用频次将数据分成三类：

家目录：用户登录上后即在家目录（home），用于存放用户环境设置等文件，不要在此目录下安装软件及存放文件，默认配额为 1GB。

作业运行数据存放目录：~/run 用于存放计算所需或访问频次较高的数据，读写性能较好，请将计算时需要使用的数据存储到该目录下并在此目录进行计算。

归档数据存放目录：用于存放存档数据或访问不频繁的数据。

默认配额为 200GB。

```
[scv0001@ln01 ~]$ ls
archive  run
[scv0001@ln01 ~]$
```

温馨提示：

如果数据量超出磁盘配额，可以联系客户经理增加配额。

2 应用及软件管理

该章节将对算例管理的全生命周期进行详细叙述，包括算例运行前的环境搭建、代码编译、算例提交、算例取消和查看算例状态。

2.1 使用 BSCC-N22 已部署的应用软件及开发环境

2.1.1 简介

“BSCC-N22”已部署多款应用软件及软件开发运行环境。

由于不同用户在“BSCC-N22”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-N22”上安装了 module 工具用于统一管理应用软件。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。

2.1.2 基本命令

常用命令如下：

命令	功能	示例
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 cuda/10.0
module show [modulesfile]	查看对应软件的环境（安装路径、库路径等）	module show cuda/10.0
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 cuda/10.0

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。

2.1.3 已部署软件及开发环境列表

已安装软件开发运行环境包括：

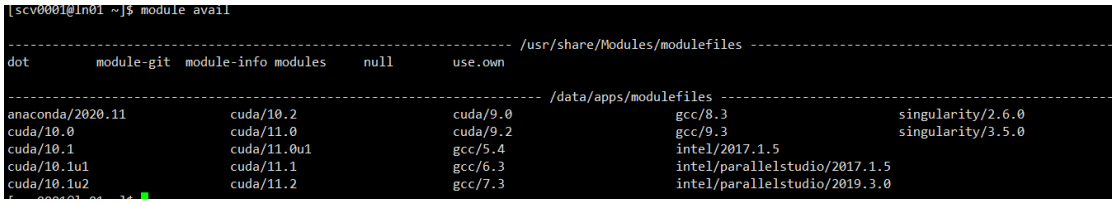
软件名称	版本
anaconda	2020.11
gcc	5.4、6.3、7.3、8.3、9.3
Intel Parallel Studio	2017.1.5、2019.3.0
CUDA	9.0~11.2 全版本
openmpi	持续更新中

2.1.4 示例说明

用户需要使用 cuda10.0 环境

操作步骤：

- 执行 module avail 查看系统中可用软件，查询到 cuda10.0 的 module 环境名称为 cuda/10.0



- 执行 module load cuda/10.0 加载 cuda10.0 环境
- 执行 module list 查看已加载的环境

```
$ module list
Currently Loaded Modulefiles:
  1) cuda/10.0
```

2.2 软件安装/环境搭建

如果执行 module avail 没有查询到所需要的软件，可以上传软件安装包，在~/run 目录下自定义安装所需软件。

软件安装基本流程：

1．上传软件安装包。

2．打开【SSH】命令行窗口，cd 到对应目录下，进行软件、工具、库等安装部署和算例运行环境搭建。

2.2.1通过 Anaconda 进行自定义 python 环境安装

Anaconda 是一个开源的 Python 发行版本，其包含了 conda、Python 等 180 多个科学包及其依赖项。支持 Linux, Mac, Windows 系统，利用 conda 工具/命令来进行包管理与环境管理，可以很方便地解决多版本 python 并存、切换以及各种第三方包安装问题。并且已经包含了 Python 和相关的配套工具。

加载 anaconda/2020.11 环境后，默认的 python 是 3.8.5（环境名称为 base）。

假设我们需要安装 python3.7，此时，我们需要做的操作如下：

1) 加载 anaconda/2020.11 环境

```
module load anaconda/2020.11
```

2) 创建名为 python37 的环境，指定 Python 版本是 3.7

```
conda create --name py37 python=3.7
```

注：

a）不用管是 3.7.x，conda 会为我们自动寻找 3.7.x 中的最新版本

b）conda 环境会默认安装到~/.conda 文件夹中，安装过程的安装包也会下载到此文件夹下，安装过程中会在~/.cache 目录下产生临时文件。

3) 查看已安装的环境

```
$ conda env list
```

输出如下,*代表目前激活的环境

```
# conda environments:
#
# conda environments:
#
base                  *  /data/apps/anaconda/2020.11
py37                     /data/home/username/.conda/envs/py37
```

4) 安装好后，使用 activate 激活某个环境

```
[username@ln01 ~]$ source activate py37
(py37) [username@ln01 ~]$ which  python
~/.conda/envs/py37/bin/python
(py37) [username@ln01 ~]$ python --version
Python 3.7.10
```

可以看到 Python 已切换到 3.7 的环境下。

5) 取消当前加载的环境，可执行：

```
$ source deactivate py37
```

6) 使用 conda 的包管理：

如：需安装 tensorflow-gpu 1.15，可执行

```
$ conda install tensorflow-gpu==1.15
```

conda 会从远程搜索 tensorflow-gpu 1.15 的相关信息和依赖组件及版本并自动安装。

7) 查看 conda 已经安装的 packages

```
$ conda list
```

最新版的 conda 是从 site-packages 文件夹中搜索已经安装的包，不依赖于 pip，因此可以显示出通过各种方式安装的包

注：如需客服协助安装、部署相关软件，可直接在您专属的微信服务群里@客服。

2.3 提交计算任务

本章节针对程序测试完成后的正式计算任务提交。推荐软件调试完毕后按照如下流程提交计算任务到计算节点进行计算。

编写提交脚本过程遇到困难时，可将程序测试能够成功运行的命令在群里发给客服，客服协助编写脚本。

使用本节方法提交计算任务后，调度系统会自动申请 GPU 资源，随后自动在计算节点执行用户所编辑的脚本内的命令，直到脚本执行结束作业自动退出（或者在作业运行时执行 scancel 命令取消作业后作业自动停止）。申请到资源到运行结束这段时间按卡时进行计费（费用=卡数 x 时间（精确到秒）x 单价）。

每台机器配置 8 块型号为 NVIDIA Tesla V100-SXM2 32GB 显存的 GPU，每块 GPU 卡默认分配 8 CPU 核及 36GB 内存，即: GPU\CPU\内存 配比为 1GPU 卡\8CPU 核\36GB 内存。如果 CPU 需求用量超出该配比可以增加 CPU 核数的申请，如果内存需求用量超出该配比可以通过增加 GPU 卡数或 CPU 核数的申请获得更多的内存配额，CPU\内存 配额比例为 1 核\4.5GB。如果 CPU（或内存）申请超出 1GPU 卡\8CPU 核\36GB 内存 比例，超出的每核按 1/8 卡的价格收费。

作业提交命令

sbatch --gpus=GPU 卡数 程序运行脚本

每个作业可以申请的 GPU 卡数范围为 1~8。不需要增加其它参数每卡默认分配 8 核及 36GB 内存。

申请到资源后程序必须按程序运行所需的参数指定进程数、线程数、卡数调用申请到的资源。注意：如果使用 srun 运行作业必须使用-n 参数指定进程数，或配合使用-c 参数指定每进程的线程数 （进程数 x 每进程线程数<申请的核数）。

作业提交命令示例：

```
$ sbatch --gpus=1 ./run.sh
```

执行此命令后即申请到 1GPU 卡、8CPU 核、36GB 内存资源。作业显示为 R（Runing）状态（parajobs 命令查看作业状态）后即开始执行 run.sh 脚本中的内容。

脚本 run.sh 示例 1，python 程序运行脚本示例：
sbatch --gpus=1 ./run.sh　(申请 1 卡，8 核，36GB 内存)

```
#!/bin/bash

#加载环境，此处加载 anaconda 环境以及通过 anaconda 创建的名为 pytorch 的环境
module load anaconda/2020.11
source activate pytorch

#python 程序运行，需在.py 文件指定调用 GPU，并设置合适的线程数，batch_size 大小等
python train.py
```

脚本 run.sh 示例 2，HPC 程序运行脚本示例：
sbatch --gpus=2 ./run.sh　(申请 2 卡，16 核，72GB 内存)

```
#!/bin/bash

#加载环境，此处加载编译程序使用的 intel parallelstudio 环境及 cuda 环境
module load intel/parallelstudio/2019.3.0
module load cuda/11.0

#程序运行，每卡分配 1 个进程 8 个线程
srun -n 2 -c 8  程序运行命令及参数
```

2.4 查看作业状态

- 查看已提交的作业

```
$ parajobs
JOBID   PARTITION   NAME   USER ST   TIME   NODES NODELIST(REASON)
 9987110          gpu .jobscri paraai_t   R          1:52          1 g0028
      9987110  作业 GPU 利用率为:
g0028: pci.bus_id, utilization.gpu [%], utilization.memory [%], memory.total [MiB], memory.free [MiB], memory.used [MiB]
g0028: 00000000:85:00.0, 0 %, 0 %, 16160 MiB, 16160 MiB, 0 MiB
```

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（RUNNING）表示正常运行，PD（PENDING）表示在排队，CG（COMPLETING）表示正在退出，S 是管理员暂时挂起，CD（COMPLETED）已完成，F（FAILED）作业已失败。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。

2.5 取消作业

执行 scancel 作业 ID 取消作业

```
$ scancel 2011812
```

2.6 程序测试流程（正式提交任务请参考 2.3）

本章节针对程序部署完成后的验证性测试、调试。推荐在不确定程序是否能够正常运行时使用此流程进行测试。此方法终端中断后程序作业即退出运行，正式提交作业请不要使用此方法。测试程序成功运行后正式提交计算任务请参考 2.3 提交计算任务

此方法提交有如下弊端：

1、如果用此方法运行程序，程序运行完成后作业不会自动停止，直到退出终端或 scancel 取消作业后作业才会停止，才会停止计费。sbatch 命令提交，任务运行完成后作业立刻停止。

2、终端断开作业会立刻停止，如果作业未运行完成时终端断开会导致任务运行中断。sbatch 命令提交作业终端断开连接作业依然在后台运行直到结束。

3、如果多个作业提交到相同机器，登录机器上只能看到最后一个提交作业申请的卡。

使用下面方法可以申请 GPU 资源，然后直接登录到计算节点上进行计算。成功申请到资源后即开始计费。

1．使用 salloc 命令申请 GPU。

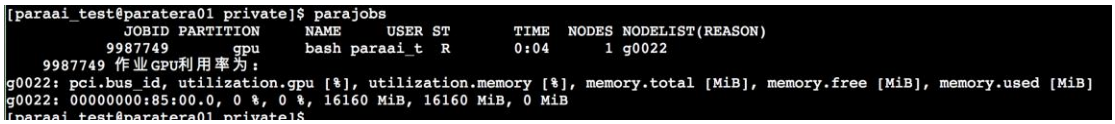
```
$ salloc --gpus=1
salloc: Granted job allocation 313
salloc: Waiting for resource configuration
salloc: Nodes g0001 are ready for job
```

如上所示，输出提示 g0001 为我们申请的主机，之后可以 ssh g0001 登录到节点进行计算。

2．也可以执行 parajobs 命令查看申请到的节点名称。

执行：

```
parajobs
```




```
$ parajobs
JOBID    PARTITION     NAME     USER ST      TIME    NODES NODELIST(REASON)
9987749          gpu        bash    paraai_t     R          0:04          1    g0022
      9987749  作业 GPU 利用率为:
g0022: pci.bus_id, utilization.gpu [%], utilization.memory [%], memory.total [MiB], memory.free [MiB], memory.used [MiB]
g0022: 00000000:85:00.0, 0 %, 0 %, 16160 MiB, 16160 MiB, 0 MiB
```

查看申请到的节点名称为 g0022，及 GPU 利用率状态。

3．登录到 g0022 进行程序运行测试。

```
ssh g0022
```

4．程序运行过程中可使用 top 查看 CPU 利用率、使用 nvidia-smi 查看 GPU 利用率。

5．退出当前终端该作业停止或执行 scancel 作业 ID 停止该作业。