多媒体计算与通讯实验室

GPU 集群 Torque 排队系统使用手册

袁平波 2016.7.4

本实验室新购进 24 块 K80 tesla GPU。为了充分利用 GPU 计算资源，我们利用 Torque 来管理同学们的计算任务队列。头结点的 IP 是 192.168.17.240。下面说明使用本 GPU 集群的几个步骤。

1. 申请帐号.

本集群有一个头结点和多个服务结点构成，因此提交计算作业需要在头结点上拥有帐号，需要使用集群的学生需要给我发一个申请邮件，同时 cc 给自己的导师，在导师批准后相应的帐号会被建立。

2. 建立 job 脚本文件

Torque 管理系统不能直接提交二进制可执行文件，需要编写一个文件的脚本文件，来描述相关参数情况。一个示例脚本文件 myjob1.pbs 如下：

```
#PBS    -N  myjob1
#PBS    -o  /home/ypb/myjob1.out
#PBS    -e  /home/ypb/myjob1.err
#PBS    -l nodes=1:gpus=1
#PBS    -r y
cd $PBS_O_WORKDIR
echo Time is `date`
echo Directory is $PWD
echo This job runs on following nodes:
cat $PBS_NODEFILE
./my_proc
```

脚本文件中定义的参数默认是以#PBS 开头的。其中：

-N 定义的是 job 名称，可以随意。

-o 定义程序运行的标准输出文件，如程序中 printf 打印信息，相当于 stdout；

-e 定义程序运行时的错误输出文件，相当于 stderr。

-l 定义了申请的结点数和 gpus 数量。nodes=1 代表一个结点，一般申请一个结点，除非采用 mpi 并行作业；gpus=1 定义了申请的 GPU 数量，根据应用实际使用的 gpu 数量来确定。队列系统的默认 job 请求时间是一周，如果运行的 job 时间估计会超过，则可以使用下面的参数：

```
#PBS    -l nodes=1:gpus=1,walltime=300:00:00
```

表示请求 300 小时的 job 时间。

-r 表示 job 立即执行。

my_proc 是用户的可执行程序。需要通过 scp 或 winscp 复制到自己的 home 目录。如果程序运行过程中需要读取数据文件和生成数据文件，也需要在运行前后上传和下传。

3. 提交作业 ：qsub

```
$qsub myjob1.pbs
```

myjob1.pbs 是前一步骤生成的脚本文件。相应可执行文件和数据文件也必须就位。

4. 查看作业：qstat -n

```
[ypb@torqueServer ~]$ qstat
Job ID                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
165.torqueServer           my_job1          ypb              00:00:00 C batch
166.torqueServer           my_job1          ypb                     0 R batch
```

上图中 165 是 jobid 运行状态有以下几种状态：

C - Job 已经运行结束
E - Job 运行结束后退出
H - Job 被挂起
Q - job 被排队，可被手动启动或路由
R - job 在运行中.
T - job 被移动
W - job 等待其执行时间到来（-a 选项设置 job 启动时间）

其中-n 参数可以列出运行 job 的结点。

其他常用命令：

1）挂起作业:qhold

Qhold 命令可以挂起作业，被挂起的作业将暂时停止执行，可以让其余的作业优先得到资源运行，被挂起的作业在 qstat 中显示为 H 状态，下面的命令将挂起 id 为 165 的 job。

```
$qhold 165
```

2）取消挂起：qrls

被挂起的作业可以重新被运行，如下面的命令将重新运行 id 为 165 的 job

```
$qrls 165
```

3）终止作业：qdel

如果用户想放弃一个作业的执行，可以使用 qdel 命令，下面的命令将终止 id 为 165 的 job。

```
$qdel 165
```

4）查看结点 pbsnodes

```
$pbsnodes
```

5) 查看空闲结点 pbsnodes -l free

$pbsnodes -l free

5. 关于集群环境的说明

应各位同学要求，集群的每一个结点都安装了 caffe 深度学习的环境。包括 Gcc4.8.5, cmake 3.1.3, python 2.7.5, blas3.4.2, numpy 1.9.1,opencv3.0.0。

头结点 torqueServer 没有编译环境，也没有 GPU 卡，如果需要测试自己的代码是否能在集群环境下运行，可以先写一个简单程序在第 7 结点上试运行：

$ssh Gpu107

$./myproc.sh #在这里运行你自己的测试程序。

另外，/opt/下面有下载好的 caffe-master.zip，可以复制到自己目录下：(以下步骤可以在 Gpu107 上完成)

$cp /opt/caffe-master.zip ~/.

a)解压:

$unzip caffe-master.zip

b)配置并修改 config 文件

$cp Makefile.config.example Makefile.config

$vi Makefile.config 修改如下参数

BLAS := atlas

BLAS_LIB := /usr/lib64/atlas

PYTHON_INCLUDE:=/usr/include/python2.7

/usr/lib/python2.7/dist-packages/numpy/core/include

PYTHON_LIB := /usr/lib64

c)编译

$make all -j12

$make test -j 12

$make runtest

d)获取数据

$ sh data/mnist/get_mnist.sh

e)重建 lmdb

$ sh examples/mnist/create_mnist.sh

f)训练数据

$ sh examples/mnist/train_lenet.sh

也可以直接复制已经解压编译好并下载了数据的文件夹（复制过程中有权限错误，忽略不会影响后序过程），这样可以免去 a-d)步骤的编译和数据下载，直接进行 e|f),如下：

$cp /opt/caffe-master ~/. -R

$ sh examples/mnist/create_mnist.sh #重建 lmdb

$ sh examples/mnist/train_lenet.sh #训练

6. 关于 GPU 集群的存储问题

用户登录 pbs 头节点(192.168.16.240)后默认的路径是/home/$USER,但/home 下的总空间只有 1T，主要用于存放代码

等重要文档，同学们在运行代码过程中用到的数据文件尽量不要放在/home 下，目前可以用于存放数据的 mount 点有/data、/data1、/data2、/data3、/data4、/data5、/data6、/data7,每个 mount 点约 1.5T 空间（使用 df -h 查看）。

同学们可以在/data$i($i=1..7)下建立自己的用户名为子目录，对于一些公共测试数据，可以不放在用户子目录下，而直接放在/datai 下，供大家使用，避免存放大量重复的数据。尤其是同一导师的学生，尽量减少重复下载和存储测试数据。

/data$i($i=1..7)是挂接在 Gpu10$i 结点上的存储，/data 挂接在头结点本地。因此如果有大量数据需要读写并且对 IO 速度有要求的应用,可以考虑把数据存放于某个 mount 点，比如/data3，然后提交 job 时使用参数

#PBS -l nodes=Gpu103:gpus=1

则可以使 job 运行在 Gpu103 结点。这样数据和代码运行于同一节点，IO 会避开 nfs 网络操作。但指点节点操作削弱了 pbs 系统的排队功能，可能会导致任务失败。因此除非有特殊要求，一般不建议这么做。