Graphics DCC Cluster User's Guide

Table of Contents

1. HOW TO ACCES THE DCC CLUSTER?

The Graphics DCC cluster is a resource of computing graphics cards accessible to all users of DCC Department.

There are three nodes in the cluster and each user should use the one that will be enumerated in the email with the account information:

- 158.109.75.50 –p 55022
- 158.109.75.51 -p 55022
- 158.109.75.52 -p 55022

We suggest you to use any ssh software like MobaXterm to connect from windows.
http://mobaxterm.mobatek.net/

From linux or mac the client is self-provided by operating system.

The home directories of each server is different in each node.

- 158.109.75.50 - /ghome
- 158.109.75.51 - /fhome
- 158.109.75.52 - /hhome

It's important to change the password of your account. To do it in the NIS domain use de command:

yppasswd

To get an account user for this cluster you can contact to Jorge Ramírez. Jorge.ramirez@uab.cat. 935813055.

2. GRAPHICS DCC CLUSTER SPECIFICATIONS.

DCC Cluster Specifications are given below:

The cluster is formed by two nodea with de following specifications:

Node 1 (3090) - 75.50

- 2 x 24 processor cores x64. Hyperthreating enabled.
- 512 GB memory.
- 8 TB disk drive.
- 1 GB network connection.

Node 2 (Titans) – 75.51

- 2 x 16 processor cores x64. Hyperthreating enabled.
- 256 GB memory.
- 3,7 TB disk drive.
- 1 GB network connection.

Node 2 (a40) – 75.52

- 1 x 24 processor cores x64. Hyperthreating enabled.
- 256 GB memory.
- 7,6 TB disk drive.
- 1 GB network connection.

10GB network truck between servers.

3. CLUSTER MANAGEMENT COMMANDS.

The system is an Ubuntu 20.04.6 LTS using Slurm 21.08.5 to manage the access to the graphics cards.

All jobs to be run using graphics cards must by processed using Slurm.

sinfo : Will show you the current list of partitions and nodes related to those partitions.

squeue : Current status of works in every queue.

scancel : To cancel a work in queue or running.

sacct : List of jobs running and already running.

sshare : Command to display statistics about priority on queues.

scontrol : Command to manage and configure the slurm software.

sreport : Make reports of all features on slurm.

4. ALIAS TO HELP YOUR DUSY TASKS.

Here I list a few alias to help on your day tasks.

```
alias sq='squeue' # To list the queue

alias sj='scontrol show job' # To show the details of any job in queue

alias sha='sshare -a' # To show the queue priority status

alias usograficas='sreport -tminper cluster utilization --tres="cpu,mem,GRES/gpu"' # Statistics on gres resources available

alias ns='nvidia-smi' # Check the current status of graphics cards
```

5. CLUSTER POLICIES AND WORKING RULES.

The cluster policies are oriented to maximize the availability and usage of resources.

The structure is the following.

3 partitions/queues available:

```
sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
mhigh        up   infinite      1    mix dcc3090
mlow         up   infinite      1    mix dcc3090
dcc          up   infinite      1    mix dcc3090
dcca40       up   infinite      1    mix dcca40
tfgm         up   infinite      1    mix etse-75-51
tfg          up   infinite      1    mix etse-75-51
```

The patitions mhigh and mlow are partitions for Master in Computer Vision students.
The partition dcc is for members of the department.
The partitions tfgm and tfg are for degree students.

The purpose of partitions is allowing users to have 1 jobs at least running at anythime even allowing them to use all resources if they are free.

mhigh, tfgm -> High priority queue. Only 1 running job from each USER. Rest of the jobs keep pending until first one finishes.
If all graphics cards are in use and the group has no running jobs in this partition, a job running in another lower priority partition may be requeued.

mlow -> Low priority queues. Any number of jobs can be executed in this queue but the priority of the next running job will be managed by the amount of time used by the account running processes in this partition. More time consumed during last 5 days -> lower priority in the partition.
If there are no free graphics card and there is any card being used by the dcc partition, the process will be requeued to priorize MCV accounts.

dcc, tfg, dcca40 -> The lowest priority partition. If there are free graphics cards in the system, users in this partition can use it with the same rules as mlow.

The way to manage preference over other jobs in MCV is done using 2 parameters.

- Partition. Parameter –p allowing you to select the partition where you are placing your work:
  mhigh
  mlow
  mhigh,mlow
- QOS. Parameter –q allowing you to select the priority of your job. Default.
  masterhigh
  masterlow

Case of usage.

A. Queue all my jobs in the mhigh partition running just 1 at once and having preference over any other job running in mlow, dcc partitions and requeuing them if required. No other job can requeue my process.
   You will need the following 2 parameters in the script.
   -p mhigh
   -q masterhigh

B. Queue all my jobs in the mlow partition and running as much as free graphics cards are available requeuing jobs in dcc partition if required. Accounting will be used for future jobs preference in this queue. Jobs launched to high partition with masterhigh qos will requeue my process.
   You will need the following 2 parameters in the script.
   -p mlow
   -q masterlow

C. Queue all my jobs in the mhigh,mlow partitions.
   The job to mhigh partition will take precedence over any other job queued but will not requeue jobs on mlow partition. Also will not be requeued by any other job.
   Also will queue jobs to mlow as in case B.
   You will need the following 2 parameters in the script.
   -p mhigh,mlow
   -q masterlow

The command sshare –a or alias sha will show you the RawUsage value that will priorize the next job to be run by a free graphics card.

The command squeue or alias sq will show you the list of jobs queued in partitions.

For the rest of the users, they should use the queue appearing in their account email confirmation.

You can find some examples in the example directory of you home directory.

6. HOW TO QUEUE JOBS IN THE CLUSTER?

Here you will find the examples about how to queue your jobs in the cluster for master users.

Example to queue your job as case A just type:

```
sbatch ~/example/mtgpuhigh.qsub
```

You will get the output of the job in /tmp directory.

```
#!/bin/bash
#SBATCH -n 4 # Number of cores
#SBATCH -N 1 # Ensure that all cores are on one machine
#SBATCH -D /tmp # working directory
#SBATCH -t 0-00:05 # Runtime in D-HH:MM
#SBATCH -p mhigh # Partition to submit to
#SBATCH -q masterhigh # Required to requeue other users mlow queue jobs
            # With this parameter only 1 job will be running in queue mhigh
            # By defaulf the value is masterlow if not defined
#SBATCH --mem 8192 # 8GB memory
#SBATCH --gres gpu:1 # Request of 1 gpu
#SBATCH -o %x_%u_%j.out # File to which STDOUT will be written
#SBATCH -e %x_%u_%j.err # File to which STDERR will be written
sleep 5
/ghome/share/example/deviceQuery
nvidia-smi
```

Example to queue your job as case C just type:

```
sbatch ~/example/mtgpulow.qsub
```

You will get the output of the job in /tmp directory.

```
#!/bin/bash
#SBATCH -n 4 # Number of cores
#SBATCH -N 1 # Ensure that all cores are on one machine
#SBATCH -D /tmp # working directory
#SBATCH -t 0-00:05 # Runtime in D-HH:MM
#SBATCH -p mhigh,mlow # Partition to submit to
#SBATCH -q masterlow # Required to requeue other users mlow queue jobs
            # With this parameter only 1 job will be running in queue mhigh
            # By defaulf the value is masterlow if not defined
#SBATCH --mem 8192 # 8GB memory
#SBATCH --gres gpu:1 # Request of 1 gpu
#SBATCH -o %x_%u_%j.out # File to which STDOUT will be written
#SBATCH -e %x_%u_%j.err # File to which STDERR will be written
sleep 5
/ghome/share/example/deviceQuery
nvidia-smi
```