Volta Cluster Pilot User Guide

February 2019

Contents

- Access
- Resources
- Volta Cluster Resources
- Short Interactive Jobs
- Batch Jobs
- Multi-GPU Scaling
- Python Packages
  - Install Python Packages in User Package Path (Default)
  - Install Python Packages in Custom Path
- PBS Job Scheduler
- Recommendations
- Additional Support & Consultation

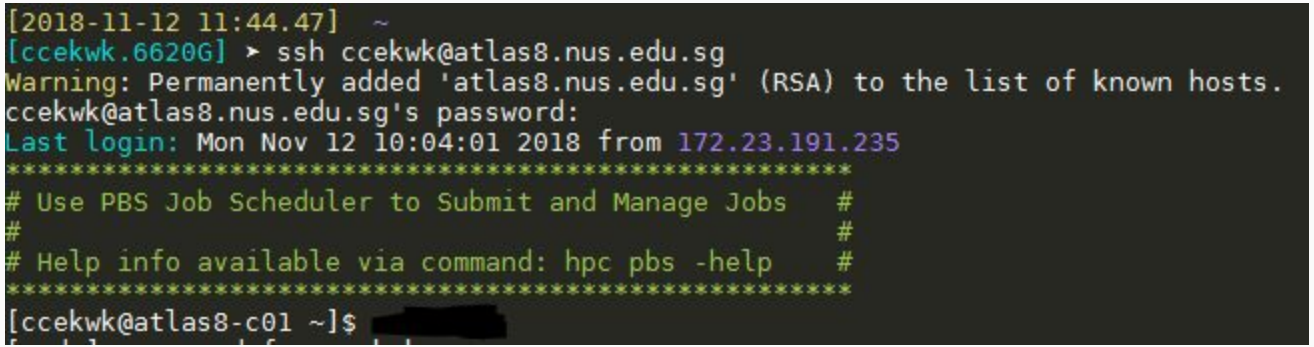
Access

Access

- Login via ssh to NUS HPC login nodes
  - atlas9
  - atlas6-c01.nus.edu.sg
  - atlas7-c10.nus.edu.sg
  - atlas8-c01.nus.edu.sg
- If you are connecting from outside NUS network, please connect to Web VPN first
  - http://webvpn.nus.edu.sg

Access

OS	Access Method	Command
Linux	ssh from terminal	`ssh nusnet_id@atlas8-c01.nus.edu.sg`
MacOS	ssh from terminal	`ssh username@hostname`
Windows	ssh using mobaxterm or putty	`ssh username@hostname`


[2018-11-12 11:44.47] ~
[ccekwk.6620G] ➤ ssh ccekwk@atlas8.nus.edu.sg
Warning: Permanently added 'atlas8.nus.edu.sg' (RSA) to the list of known hosts.
ccekwk@atlas8.nus.edu.sg's password:
Last login: Mon Nov 12 10:04:01 2018 from 172.23.191.235
Use PBS Job Scheduler to Submit and Manage Jobs
# #
Help info available via command: hpc pbs -help
[ccekwk@atlas8-c01 ~]$

Logging in

Resources

Resources: Hardware

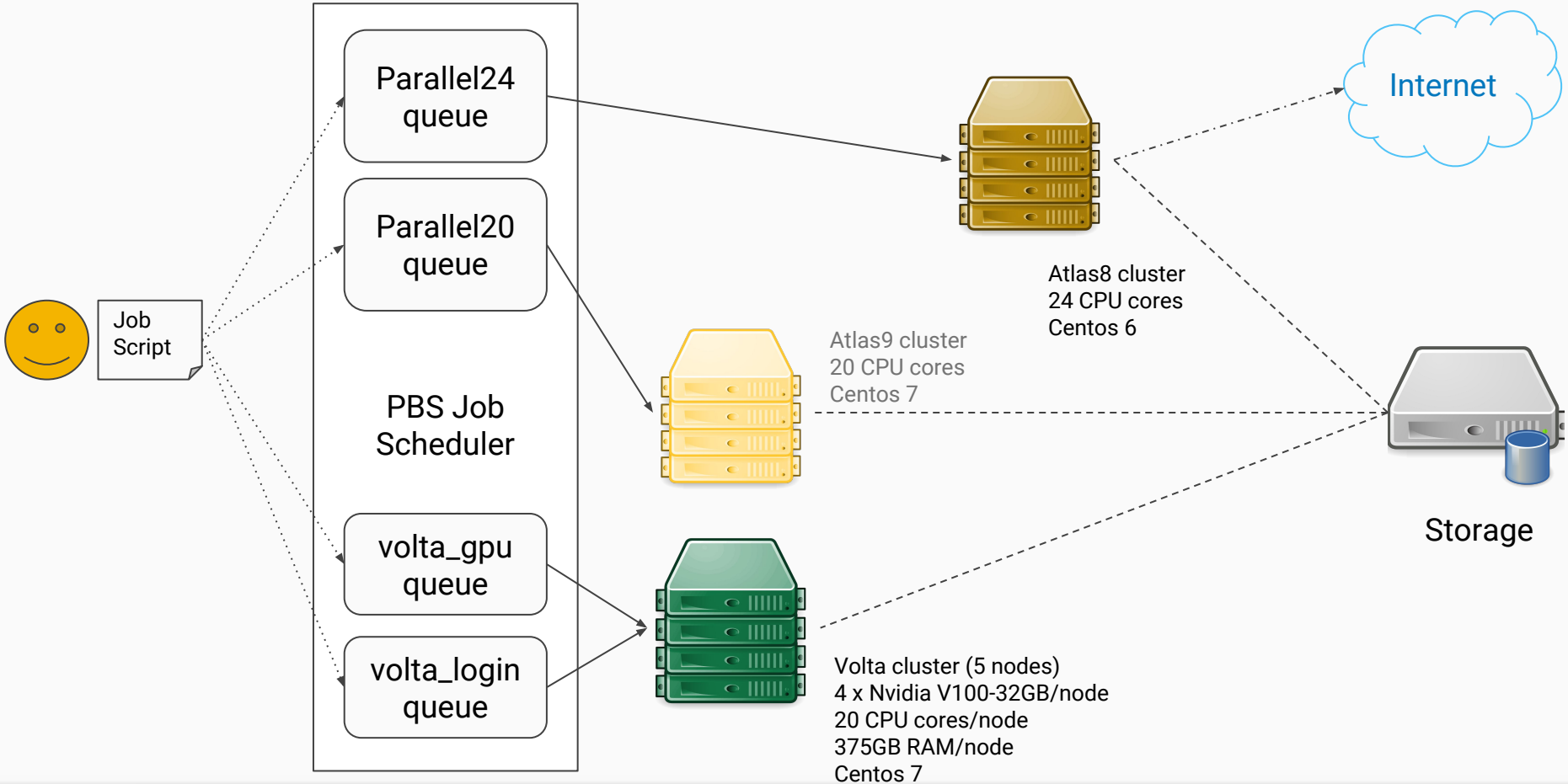
- Standard CPU HPC Clusters
  - Atlas 8
  - Atlas 9
- GPU Clusters
  - 5 nodes x 4 Nvidia Tesla V100-32GB

No internet access on Volta Servers and Atlas9

Resources: Hardware/Storage

Directories	Feature	Disk Quota	Backup	Description
`/home/svu/$USERID`	Global	20 GB	Snapshot	Home Directory. U:drive on your PC.
`/hpctmp/$USERID`	Local on All Atlas/Volta cluster	500 GB	No	Working Directory. Files older than 60 days are purged automatically
`/scratch`	Local to each Volta node		No	For quick read/write access to datasets. Routinely purged.

Note: Type “hpc s” to check your disk quota for your home directory


Parallel24 queue
Parallel20 queue
PBS Job Scheduler
volta_gpu queue
volta_login queue
Job Script
Internet
Atlas8 cluster
24 CPU cores
Centos 6
Atlas9 cluster
20 CPU cores
Centos 7
Storage
Volta cluster (5 nodes)
4 x Nvidia V100-32GB/node
20 CPU cores/node
375GB RAM/node
Centos 7

Notes

- The V100-32GB GPUs are the latest and fastest and largest GPU available
- They are most suited for large batch workloads (complex model, large datasets)
- For development and testing work, please use the volta_login interactive queue and request for 1 GPU
- Access is through PBS Job Scheduler

Volta Cluster Resources

Resources: Containers

Each Deep Learning framework has its own singularity container:

```
l[ccekwk@atlas9-c01 ~]$ ls /app1/common/singularity-img/3.0.0/
caffe_0.17.2_nvcr_19.01-py27.simg  darknet_jimmylin.simg                  lgbm_v2.simg                  rapidsai_5.0_nvcr_c10_u16.04-py36.simg
caffe2_0.8.1_nvcr_18.08-py27.simg  intel_mkl_tensorflow_1.11.0_cpu_py27.simg  openfoamplus_v1806.sig        tensorflow_1.12_nvcr_19.01-py3.simg
caffe2_0.8.1_nvcr_18.08-py35.simg  intel_mkl_tensorflow_1.11.0_cpu_py35.simg  pytorch_1.0.0_nvcr_19.01_py3.simg  tensorflow_1.8.0_nvcr_18.07-py35.simg
```

Containers are located in

- **/app1/common/singularity-img/3.0.0/**
- **/app1/common/singularity-img/2.5.2/**

All scripts must be executed within a container.

Containers are created from Nvidia GPU Cloud containers (ngc.nvidia.com).
Contact us if you have any container requests.

2.5.2

GPU

- caffe_18.08-py2.simg
- pytorch_nvcr_18.08-py3.simg
- pytorch_18.09-py3.simg
- tf_1.8_nvcr_18.07-py3.simg
- tensorflow_1.9.0_nvcr_18.08-py3.simg
- tf_1.10_nvcr_18.09-py3.simg

CPU

- tf_1.9_cpu.simg

3.0.0

GPU

- caffe_0.17.2_nvcr_19.01-py27.simg
- caffe2_0.8.1_nvcr_18.08-py27.simg
- caffe2_0.8.1_nvcr_18.08-py35.simg
- torch7_nvcr_18.08-py2.simg
- pytorch_1.0.0_nvcr_19.01_py3.simg
- tensorflow_1.7.0_nvcr_18.05-py3.simg
- tensorflow_1.8.0_nvcr_18.07-py35.simg
- tensorflow_1.9.0_nvcr_18.08-py3.simg
- tensorflow_1.12_nvcr_19.01-py3.simg
- darknet_jimmylin.simg
- rapidsai_5.0_nvcr_c10_u16.04-py36.simg
- lgbm_v2.simg

CPU

- Intel_mkl_tensorflow_1.11.0_cpu_py35.simg
- intel_mkl_tensorflow_1.11.0_cpu_py27.simg

Running a Container

```
$ singularity exec $image bash
```

Substitute **$image** with full path to container image file

E.g.:

```
$ singularity exec /app1/common/singularity-img/3.0.0/tensorflow_1.12_nvcr_19.01-py3.simg bash
```

Put the commands in a job script

Short Interactive Jobs

Short Interactive Jobs

Run short interactive jobs meant for debugging your code on GPU.
GPUs are shared for interactive jobs.

```
qsub -I -l select=1:mem=10GB:ncpus=5:ngpus=1 -q volta_login
```

- `-I` : Interactive
- Mem: Max 96GB
- ncpus: 1 to 20
- ngpus: 1 to 2
- `-q` : volta_login queue

1. Default Walltime: 1 hour
2. Max Walltime: 4 hours

```
[ccekwk@atlas9-c01 horovod]$ qsub -I -l select=1:ncpus=5:ngpus=1 -l walltime=0:05:00 -q volta_login
qsub: waiting for job 789846.venus01 to start
qsub: job 789846.venus01 ready

[ccekwk@volta01 ~]$ uname -a
Linux volta01 3.10.0-862.el7.x86_64 #1 SMP Fri Apr 20 16:44:24 UTC 2018 x86_64 x86_64 x86_64 GNU/Linux
[ccekwk@volta01 ~]$ cat /etc/centos-release
CentOS Linux release 7.5.1804 (Core)
[ccekwk@volta01 ~]$
```

Launching an interactive job

```
[ccekwk@volta01 ~]$ singularity exec /app1/common/singularity-img/3.0.0/tensorflow_1.12_nvcr_19.01-py3.simg bash
ccekwk@volta01:~$ uname -i
x86_64
ccekwk@volta01:~$ cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04.5 LTS"
```

Ubuntu 16.04 container running on Centos 7 host

Interactive Jupyter Notebook

- You can run a Jupyter Notebook instance for a short amount of time (up to 4 hours) using the **volta_login** interactive queue
- The Jupyter Notebook instance will run inside a container of your choice.

Interactive Jupyter Notebook

- Login to Atlas9 and set your Jupyter password (ONLY DO ONCE)
- In Atlas9 run the following commands:

```
bash
module load singularity/3.0.0
singularity exec $image jupyter notebook password
```

- You can replace `$image` with the container image of your choice, e.g.:
  `/app1/common/singularity-img/3.0.0/tensorflow_1.12_nvcr_19.01-py3.simg`
- Set your password as prompted

Interactive Jupyter Notebook

- Launch an Interactive Job from Atlas9
- Change walltime and other resources as required

```
qsub -I -l select=1:mem=10GB:ncpus=5:ngpus=1 -q volta_login -l walltime=0:10:00
```

```
[ccekwk@atlas9-c01 ~]$ qsub -I -l select=1:mem=10GB:ncpus=5:ngpus=1 -q volta_login -l walltime=0:10:00
qsub: waiting for job 797088.venus01 to start
qsub: job 797088.venus01 ready

[ccekwk@volta01 ~]$
```

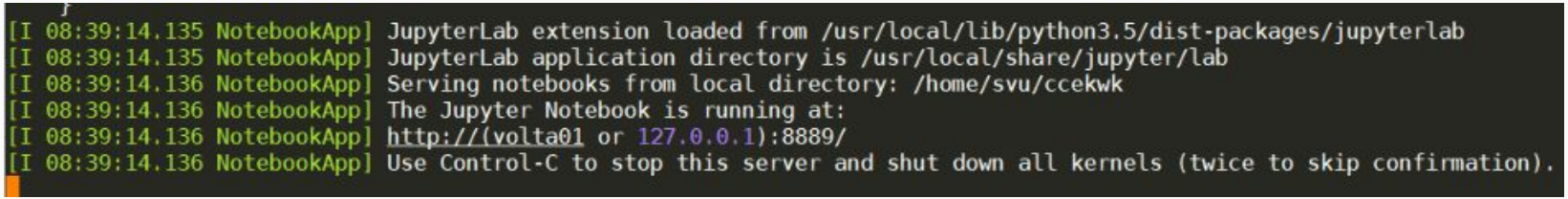
Take note of which volta node it is running in

Interactive Jupyter Notebook

- In the Volta node, launch jupyter notebook in a container of your choosing
- Execute the following command in 1 line

```
singularity exec $image jupyter notebook --no-browser --port=8889 --ip=0.0.0.0
```

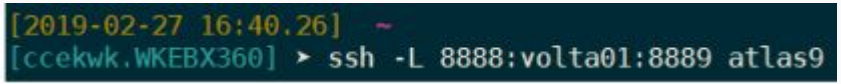
- You can replace `$image` with the container image of your choice, e.g.:
  `/app1/common/singularity-img/3.0.0/tensorflow_1.12_nvcr_19.01-py3.simg`



Interactive Jupyter Notebook

- On your personal computer in Mobaxterm run the following command
- Replace volta01 with the name of the node your interactive job is launched in

```
ssh -L 8888:volta01:8889 atlas9
```



- Open your browser and browse to http://localhost:8888

Partial Node Jobs

Partial Node

Specify resource requirements in job script:

```
#PBS -l select=1:mem=10GB:ncpus=5:ngpus=1
```

```
[ccekwk@atlas9-c01 ~]$ qsub -I -l select=1:mem=10GB:ncpus=5:ngpus=1 -q volta_login -l walltime=0:10:00
qsub: waiting for job 798234.venus01 to start
qsub: job 798234.venus01 ready

[ccekwk@volta01 ~]$ nvidia-smi
Thu Feb 28 09:00:40 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.48                 Driver Version: 410.48                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:1A:00.0 Off |                    0 |
| N/A   35C    P0    46W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
[ccekwk@volta01 ~]$
```

Partial Node

```
[ccekwk@atlas9-c01 ~]$ qsub -I -l select=1:mem=10GB:ncpus=5:ngpus=2 -q volta_login -l walltime=0:10:00
qsub: waiting for job 798249.venus01 to start
qsub: job 798249.venus01 ready

[ccekwk@volta01 ~]$ nvidia-smi
Thu Feb 28 09:25:26 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.48                 Driver Version: 410.48                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:1A:00.0 Off |                    0 |
| N/A   36C    P0    46W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000000:1C:00.0 Off |                    0 |
| N/A   33C    P0    44W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Batch Jobs

Batch Jobs

- Up to 4 GPUs per job
  - Please request only what you need
- Most applications use only 1 GPU by default
  - Requesting >1 GPUs does not make it faster. The other GPUs will not be utilised at all.
- Most applications will not scale well with > 2 GPUs
  - Use Horovod for better scaling
- 1 GPU is most likely more than enough as the V100-32GB is the latest and fastest

Sample Job Script

For Batch Jobs

Volta GPU Cluster Pilot Phase

Batch Job Script for volta_gpu

```
#!/bin/bash
#PBS -P volta_pilot
#PBS -j oe
#PBS -N tensorflow
#PBS -q volta_gpu
#PBS -l select=1:ncpus=5:mem=50gb:ngpus=1:mpiprocs=1
#PBS -l walltime=24:00:00

cd $PBS_O_WORKDIR;
np=$(cat ${PBS_NODEFILE} | wc -l);

image="/app1/common/singularity-img/3.0.0/tensorflow_1.12_nvcr_19.01-py3.simg"

singularity exec $image bash << EOF > stdout.$PBS_JOBID 2> stderr.$PBS_JOBID

python cifar10_resnet.py

# you can put more commands here
# echo “Hello World”
EOF
```

Green is user configurable
Black is fixed
mpiprocs = ngpus

Multi-GPU Scaling

Scaling for Multi GPU with Horovod

For better scaling when utilising multi-gpus, use Horovod.
Distributed training framework for TensorFlow, Keras, PyTorch, and MXNet.

To use Horovod, you need to set the following in your job script.

```
#PBS -l select=1:ncpus=5:mem=50gb:ngpus=4:mpiprocs=4
```

mpiprocs value must be the same as ngpus value

```
NCCL_DEBUG=INFO ; export NCCL_DEBUG
mpirun -np $np -x NCCL_DEBUG python keras_mnist_advanced.py
```

Use mpirun to execute python

Horovod Job Script for volta_gpu

```
#!/bin/bash
#PBS -P volta_pilot
#PBS -j oe
#PBS -N tensorflow
#PBS -q volta_gpu
#PBS -l select=1:ncpus=5:mem=50gb:ngpus=2:mpiprocs=2
#PBS -l walltime=24:00:00

cd $PBS_O_WORKDIR;
np=$(cat ${PBS_NODEFILE} | wc -l);

image="/app1/common/singularity-img/3.0.0/tensorflow_1.12_nvcr_19.01-py3.simg"

singularity exec $image bash << EOF > stdout.$PBS_JOBID 2> stderr.$PBS_JOBID

NCCL_DEBUG=INFO ; export NCCL_DEBUG
mpirun -np $np -x NCCL_DEBUG python keras_mnist_advanced.py

EOF
```

Green is user configurable
Black is fixed
mpiprocs = ngpus

Multi-Node Multi-GPU Scaling

- Not supported/allowed in NUS HPC Volta Cluster
- Use NSCC DGX-1 Cluster
  - 8 x V100-16GB, 6 Nodes

Python Packages

Installing Python Packages: Default User Package Path

On Atlas8:

```
$ module load singularity/3.0.0

$ singularity exec /app1/common/singularity-img/3.0.0/[cont.simg] bash
```

In Container:

```
$ pip install --user package_name
```

Python in the container will automatically locate packages installed with this method

Installing Python Packages: Specify Custom Path

On Atlas8:

```
$ module load singularity/3.0.0
```

```
$ singularity exec /app1/common/singularity-img/3.0.0/[cont.simg] bash
```

In Container:

```
$ pip install --prefix=/home/svu/[nusnet_id]/volta_pypkg/ package_name
```

Using User-installed Python Packages: Custom Path

In job script add the following:

```
export PYTHONPATH=$PYTHONPATH:/home/svu/[nusnet_id]/volta_pypkg/lib/python3.5/site-packages
```

Batch Job Script for volta_gpu with Custom Python Lib Path

```
#!/bin/bash
#PBS -P volta_pilot
#PBS -j oe
#PBS -N tensorflow
#PBS -q volta_gpu
#PBS -l select=1:ncpus=5:mem=50gb:ngpus=2:mpiprocs=2
#PBS -l walltime=24:00:00

cd $PBS_O_WORKDIR;
np=$(cat ${PBS_NODEFILE} | wc -l);

image="/app1/common/singularity-img/3.0.0/tensorflow_1.12_nvcr_19.01-py3.simg"

singularity exec $image bash << EOF > stdout.$PBS_JOBID 2> stderr.$PBS_JOBID

PYTHONPATH=$PYTHONPATH:/home/svu/[nusnet_id]/volta_pypkg/lib/python3.5/site-packages
export $PYTHONPATH

python cifar10_resnet.py
EOF
```

Green is user configurable
Black is fixed
mpiprocs = ngpus

PBS Job Scheduler

Steps

You have to run:

1. Prepare your python script in your working directory
2. Create a PBS job script and save it in your working directory
    a. Example job scripts are in the following 2 slides
3. Submit PBS job script to PBS Job Scheduler

Server will run:

1. Job is in PBS Job Scheduler queue
2. Job Scheduler waits for server resources to be available
3. If available, Job Scheduler runs your script on remote gpu server

Submitting a Job

Save your job script (previous slides for examples) in a text file (e.g. train.pbs) then run the following commands

```
shell$ qsub train.pbs
675674.venus01


shell$ qstat -xfn

venus01:
                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
669468.venus01  ccekwk   volta    cifar_noco    --    1   1   20gb 24:00 F   --
    --
674404.venus01  ccekwk   volta    cifar_noco    --    1   1   20gb 24:00 F   --
   TestVM/0
675674.venus01  ccekwk   volta    cifar_noco    --    1   1   20gb 24:00 Q   --
    --
```

Statuses: Q(ueue), F(inish), R(unning), E(rror), H(old)

```
[ccekwk@atlas8-c01 classification]$ qsub train.pbs
697978.venus01
[ccekwk@atlas8-c01 classification]$ qstat -xfn

venus01:
                                                                  Req'd  Req'd   Elap
Job ID          Username Queue    Jobname    SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ---------- ------ --- --- ------ ----- - -----
695126.venus01  ccekwk   azgpu    cifar_noco    --    1   4   40gb 24:00 F   --
    --
697978.venus01  ccekwk   azgpu    cifar_noco    --    1   4   40gb 24:00 R   --
   TestVM/0*4
[ccekwk@atlas8-c01 classification]$
```

Statuses: Q(ueue), F(inish), R(unning), E(rror), H(old)

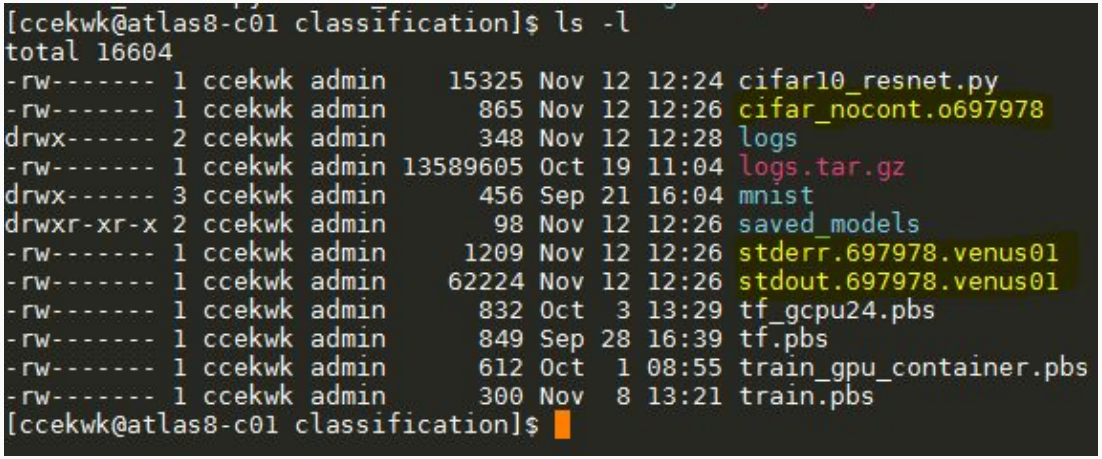
Submitting a Job

Useful PBS Commands

Action	Command
Job submission	`qsub my_job_script.txt`
Job deletion	`qdel my_job_id`
Job listing (Simple)	`qstat`
Job listing (Detailed)	`qstat -ans1`
Queue listing	`qstat -q`
Completed Job listing	`qstat -H`
Completed and Current Job listing	`qstat -x`
Full info of a job	`qstat -f job_id`

Log Files

- Output (stdout)
  - `stdout.$PBS_JOBID`
- Error (stderr)
  - `stderr.$PBS_JOBID`
- Job Summary
  - `job_name.o$PBS_JOBID`

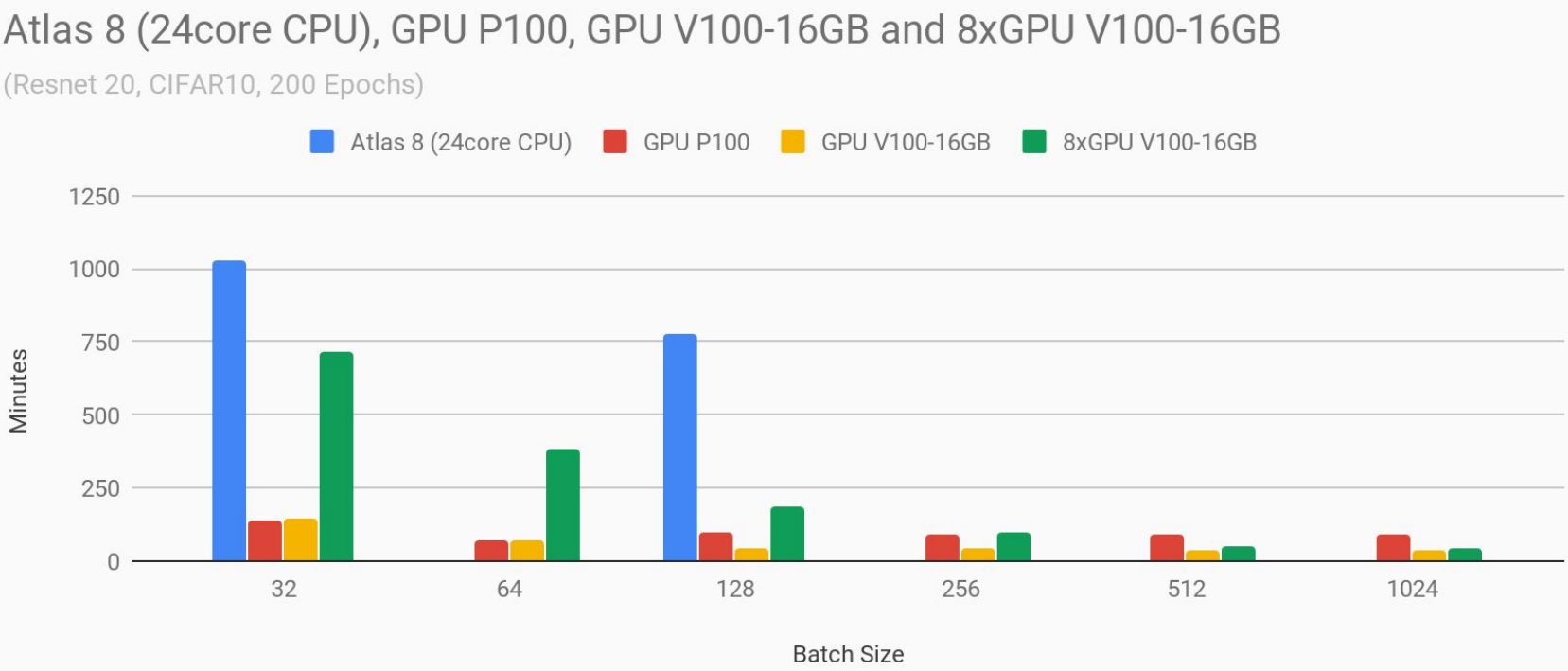


Recommendations

Recommendations

- Max walltime: 48 hours
- Default walltime: 24 hours
- Implement weights/model checkpointing
  - Save your model weights after every training epoch
- Copy large datasets to /scratch on Volta nodes
- Save weights to `/hpctmp/nusnet_id/` directory
  - Backup your important weights from /hpctmp often to your own computer

Deep Learning on CPU vs GPU vs GPUs



Additional Support

nTouch

https://ntouch.nus.edu.sg/ux/myitapp/#/catalog/home

Project Collaboration

Email

dataengineering@nus.edu.sg