Deep Learning on NUS HPC

User Guide
December 2018

Contents

- Access
- Resources
- Workflow
  - Steps
  - PBS Job Script Template
    - GPU
  - Submitting a Job
  - Log files
- Recommendations
- Extras
  - PBS Job Script Template
    - Install Python Packages
    - CPU (with Container)
- Additional Support & Consultation

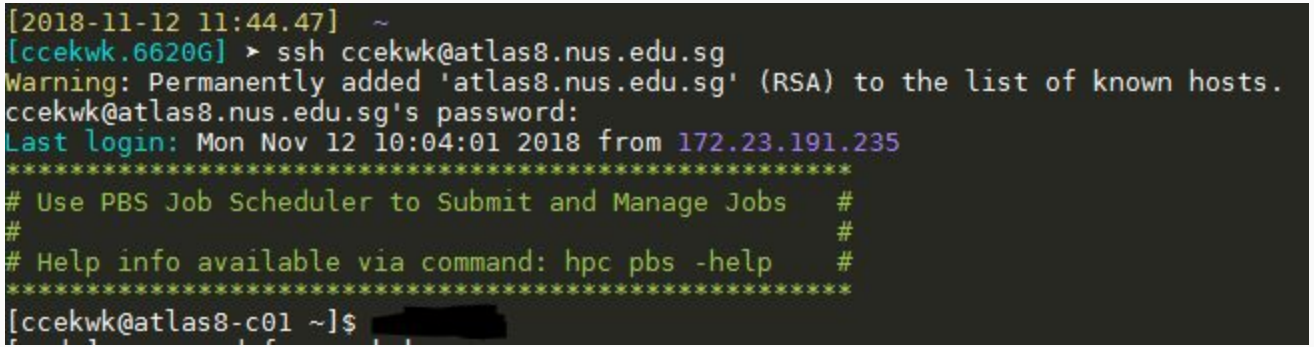
Access

Access

- Login via ssh to NUS HPC login nodes
  - atlas5-c01.nus.edu.sg
  - atlas6-c01.nus.edu.sg
  - atlas7-c10.nus.edu.sg
  - atlas8-c01.nus.edu.sg
  - gold-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

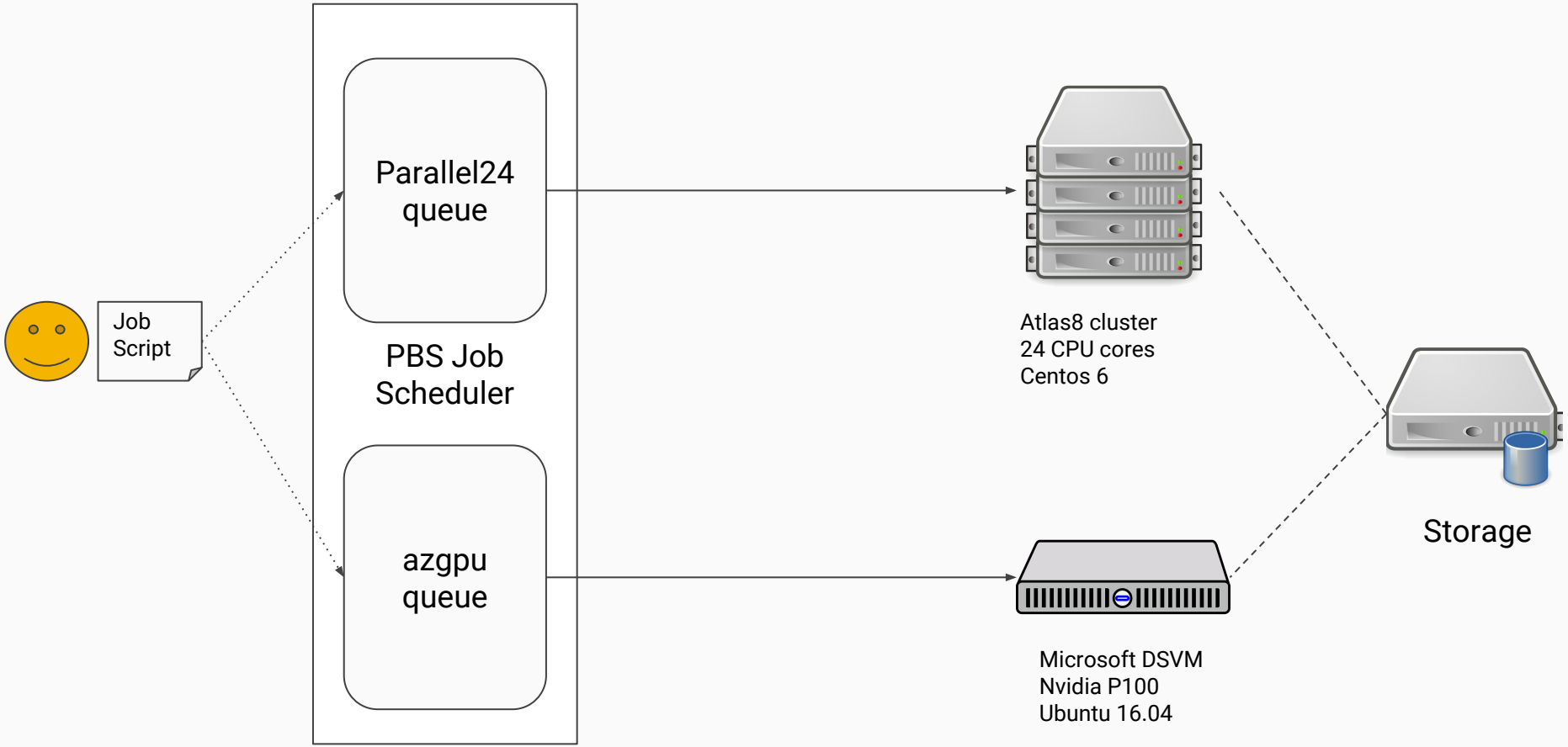
Resources: Hardware

- Standard CPU HPC Clusters
  - Atlas 8
- GPU Clusters
  - 5 nodes x 4 Nvidia Tesla V100-32GB
- Microsoft Azure Data Science VM Instance
  - Nvidia Tesla P100-16GB
  - Azgpu queue

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 cluster	500 GB	No	Working Directory. Files older than 60 days are purged automatically
`/hpctmp2/$USERID`	Local on All Atlas cluster	500 GB	No	Working Directory. Files older than 60 days are purged automatically

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


Parallel24
queue
PBS Job
Scheduler
azgpu
queue
Job
Script
Atlas8 cluster
24 CPU cores
Centos 6
Storage
Microsoft DSVM
Nvidia P100
Ubuntu 16.04

Resources: Software

- Deep Learning
  - Tensorflow
  - Keras
  - Pytorch
  - Others Available in Microsoft DSVM (azgpu)
- Machine Learning
  - Scikit-Learn
  - LightGBM
  - etc
- Reinforcement Learning
  - OpenAI Gym
- Others
  - Python 3.5
  - Matlab
  - Singularity Containers
- Computer Vision
  - OpenCV

Learn Linux Commands here.

Resources: Containers

shell$ module load singularity

Welcome to Singularity, you can find prebuilt images here:

/app1/common/singularity-img/2.5.2

usage:

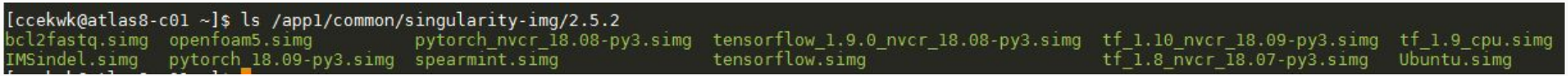
singularity exec /app1/common/singularity-img/2.5.2/Ubuntu.simg cat /etc/os-release; echo "\nHello from Container\n"

```
[ccekwk@atlas8-c01 ~]$ module load singularity
Welcome to Singularity, you can find prebuilt images here:
        /app1/common/singularity-img/2.5.2
 usage:
        singularity exec /app1/common/singularity-img/2.5.2/Ubuntu.simg cat /etc/os-release; echo "\nHello from Container\n"
```

Resources: Containers

```
shell$ ls /app1/common/singularity-img/2.5.2

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
tf_1.9_cpu.simg
tensorflow.simg
```

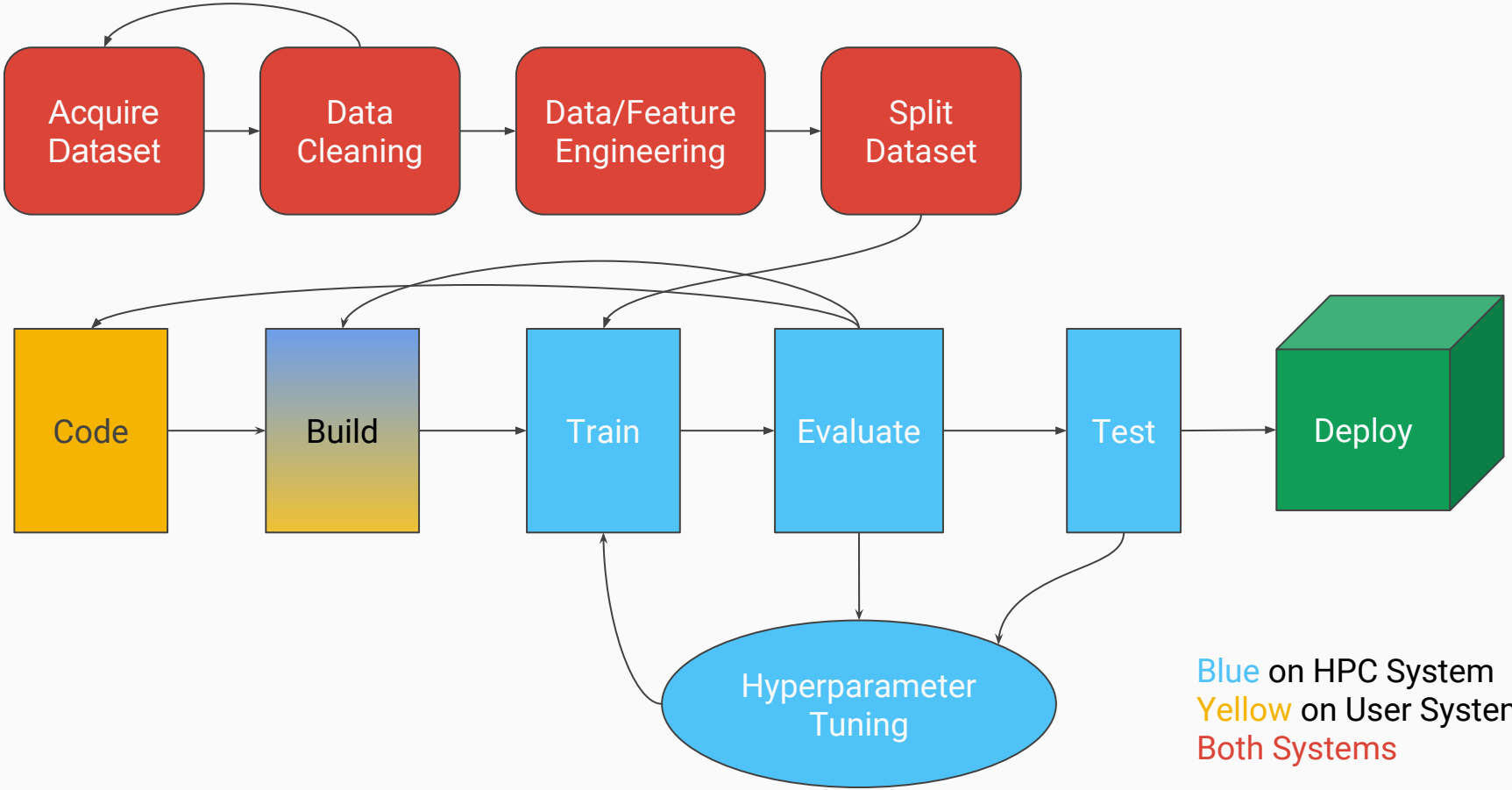


Python Packages

- User-level installable via:
  - `pip install --user package_name`
  - Please run it in a job script (login node OS/Kernel/Python different from GPU node)
- Packages that needs to be installed at system level
  - Drop us a request

Workflow

Example Workflow



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

PBS Job Script Template: Azgpu Queue (GPU)

```
#!/bin/bash
#PBS -P Project_Name_of_Job
#PBS -j oe
#PBS -N Job_Name_1
#PBS -q azgpu
#PBS -l select=1:ncpus=6:mem=60gb:ngpus=1
#PBS -l walltime=24:00:00

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

source /etc/profile.d/rec_modules.sh

# Load Azure Data Science python
module load azure

# Run your scripts
python my_deeplearningscript.py > stdout.$PBS_JOBID 2> stderr.$PBS_JOBID
```

Green is user configurable
Yellow is fixed

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   azgpu    cifar_noco    --    1   1   20gb 24:00 F   --
    --
674404.venus01  ccekwk   azgpu    cifar_noco    --    1   1   20gb 24:00 F   --
   TestVM/0
675674.venus01  ccekwk   azgpu    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)

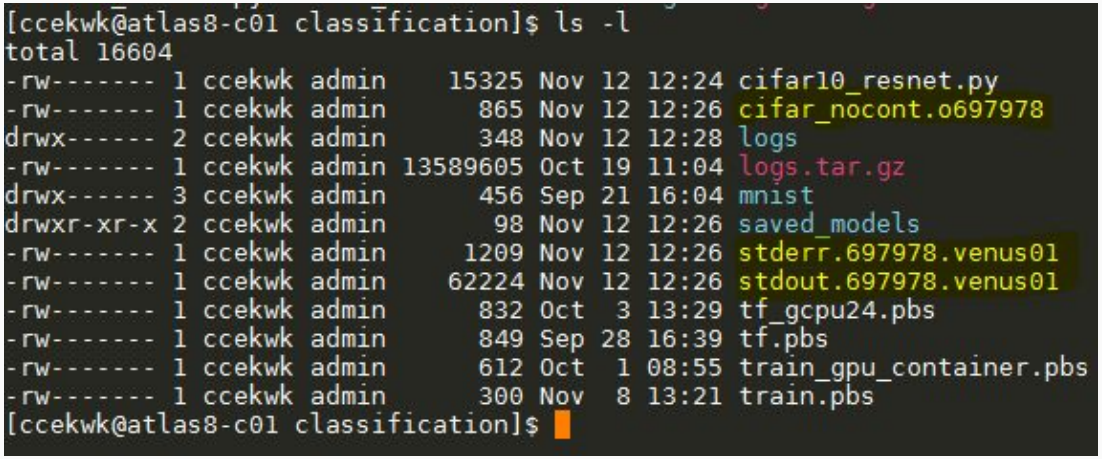
Submitting a Job

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`



GPU: In stderr file

```
shell$ cat stderr.627665.venus01
WARNING: Non existent 'bind path' source: '/hpctmp2'
2018-10-01 00:56:38.182731: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1392]
Found device 0 with properties:
name: Tesla P100-PCIE-16GB major: 6 minor: 0 memoryClockRate(GHz): 1.3285
pciBusID: e44c:00:00.0

2018-10-01 00:56:38.182778: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1471]
Adding visible gpu devices: 0
2018-10-01 00:56:39.829596: I tensorflow/core/common_runtime/gpu/gpu_device.cc:952]
Device interconnect StreamExecutor with strength 1 edge matrix:
2018-10-01 00:56:39.829645: I tensorflow/core/common_runtime/gpu/gpu_device.cc:958]
0
2018-10-01 00:56:39.829654: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 0:
N
2018-10-01 00:56:39.829940: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1084]
Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 15131 MB
memory) -> physical GPU (device: 0, name: Tesla P100-PCIE-16GB, pci bus id:
e44c:00:00.0, compute capability: 6.0)
Using TensorFlow backend.
```

GPU: In stdout file

```
shell$ tail stdout.627665.venus01
 7584/10000 [=====================>........] - ETA: 0s
 7968/10000 [======================>.......] - ETA: 0s
 8352/10000 [=======================>.....] - ETA: 0s
 8736/10000 [========================>....] - ETA: 0s
 9056/10000 [=========================>...] - ETA: 0s
 9440/10000 [==========================>..] - ETA: 0s
 9824/10000 [===========================>.] - ETA: 0s
10000/10000 [==============================] - 1s 138us/step
Test loss: 0.44839015469551086
Test accuracy: 0.9124
```

GPU: Resource Usage in Job Summary file

```
================================================================================
      Resource Usage on 2018-10-01 11:22:36.611357:
      JobId: 627665.venus01                                Project: cifar_singularity
      Submission Host: atlas8-c01.hpc.local
      Exit Status: 0
      NCPUs Requested: 1                                   NCPUs Used: 1
      Memory Requested: 20gb                               Memory Used: 2847756kb
      Vmem Used: 34965284kb
      CPU Time Used: 04:31:33
      Walltime requested: 24:00:00                         Walltime Used: 02:26:44
      Submit Time:  Mon Oct  1 08:55:45 2018               Start Time: Mon Oct  1 08:55:50
2018
      End Time: Mon Oct  1 11:22:36 2018
      Execution Nodes Used: (TestVM:ncpus=1:mem=20971520kb:ngpus=1)

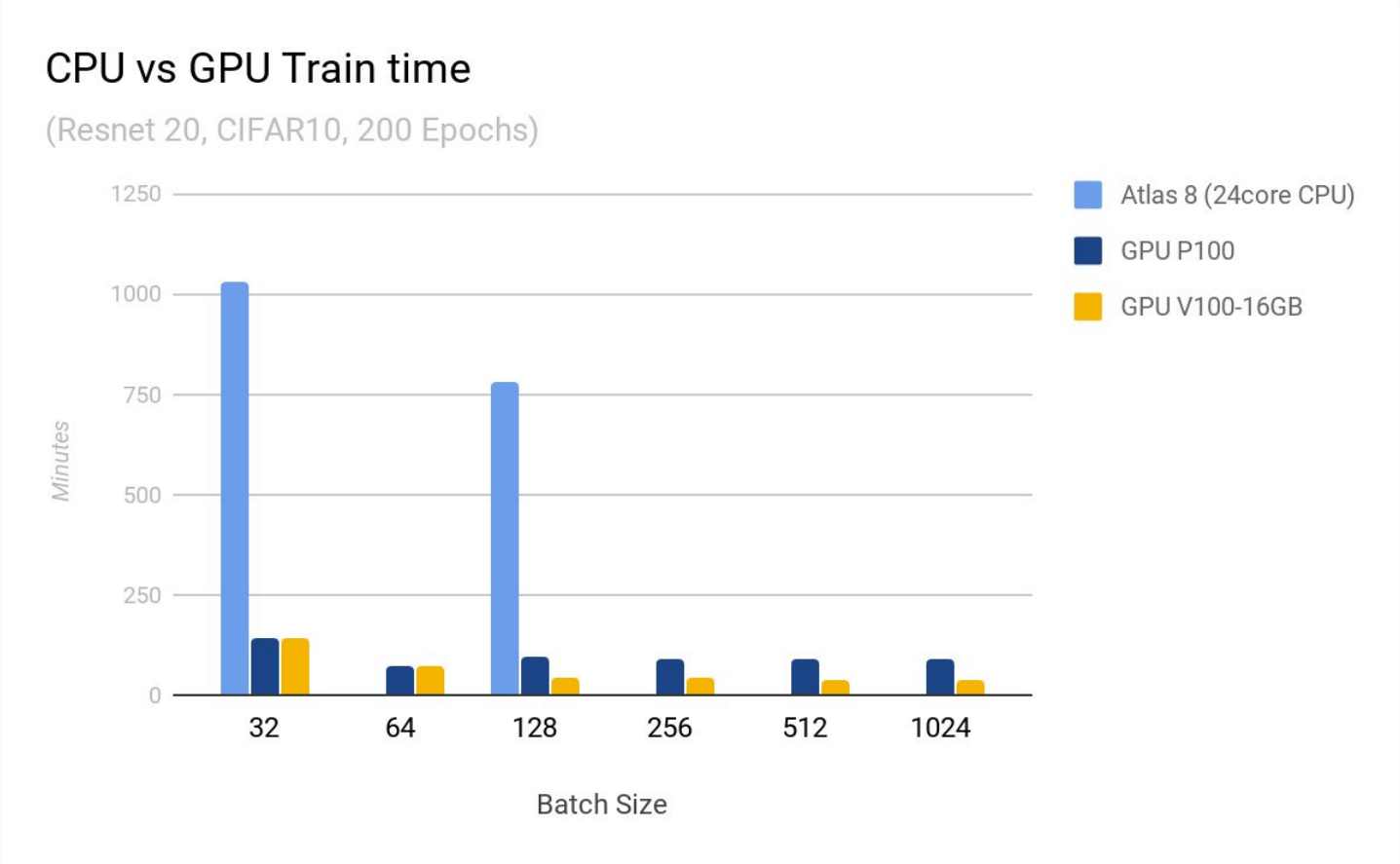
================================================================================
```

Recommendations

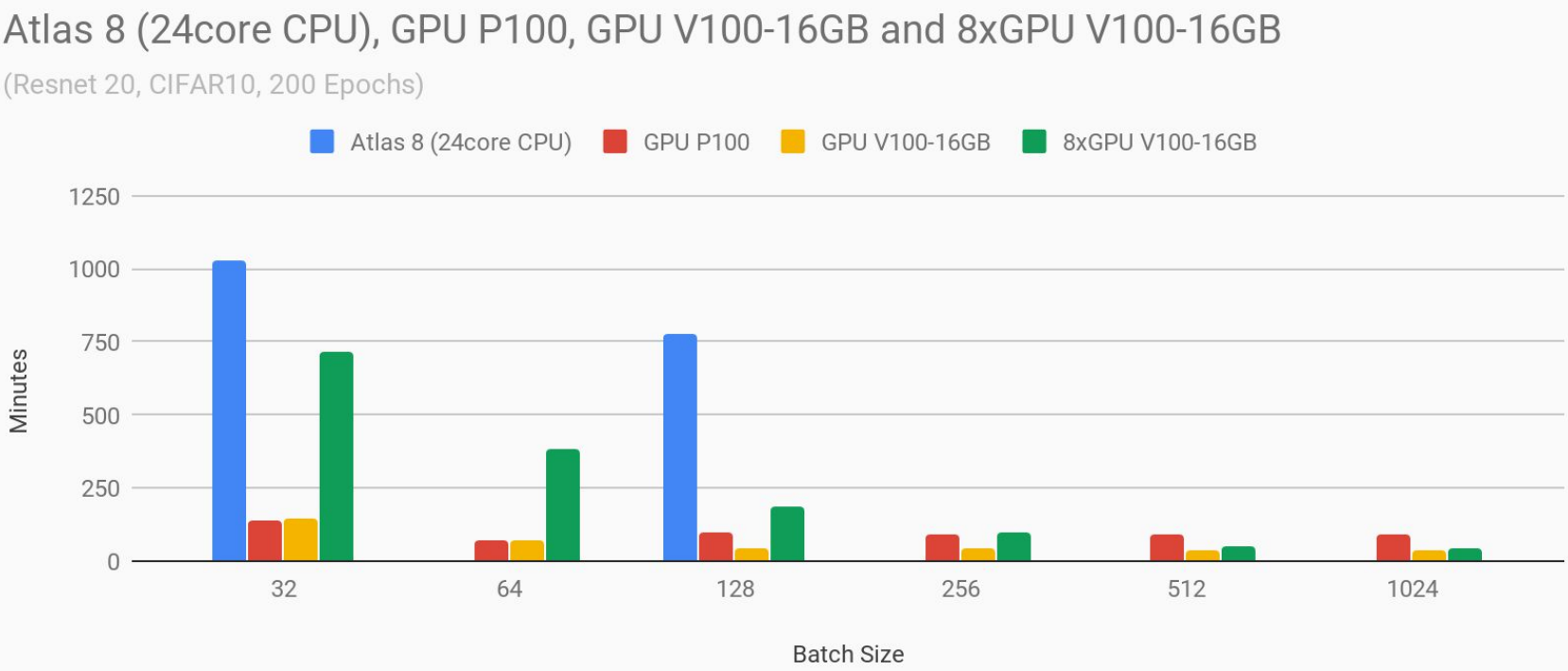
Recommendations

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

Deep Learning on CPU vs GPU



Deep Learning on CPU vs GPU vs GPUs



Extras

PBS Job Script Template: Install Python Package

```
#!/bin/bash
#PBS -P Project_Name_of_Job
#PBS -j oe
#PBS -N install_python_package
#PBS -q azgpu
#PBS -l select=1:ncpus=1:mem=5gb:ngpus=1
#PBS -l walltime=24:00:00

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

source /etc/profile.d/rec_modules.sh

module load azure
pip install --user python_packagename
```

Green is user configurable
Yellow is fixed

PBS Job Script Template: Atlas8 Queue (CPU)

```
#!/bin/bash
#PBS -P Project_Name_of_Job
#PBS -j oe
#PBS -N Job_Name_1
#PBS -q parallel24
#PBS -l select=1:ncpus=24:mem=48gb
#PBS -l walltime=00:24:00

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

source /etc/profile.d/rec_modules.sh

module load singularity
image="/app1/common/singularity-img/2.5.2/tf_1.9_cpu.simg"
singularity shell -B /hpctmp $image  << EOF > stdout.$PBS_JOBID 2> stderr.$PBS_JOBID
python cifar10_resnet.py
EOF
```

Green is user configurable
Yellow is fixed

Additional Support

Email

AI/ML/DL/Analytics:
dataengineering@nus.edu.sg