AC922 DEMO System

Ing. Florin Manaila
Senior Architect and Inventor
Cognitive Systems (HPC and Deep Learning)
IBM Systems Hardware Europe
Member of the IBM Academy of Technology (AoT)

Lund, March 21, 2019

By OpenPOWER Academia

The LAB

OpenPOWER™

Part 1

PowerAI

redhat

Part 2

PowerAI Vision

IBM **Cloud** Private

kubernetes

ACCESS to the LAB

OpenVPN Access is need on your laptop to get access to the OpenPower LAB

1. OpenVPN Software

If your workstation is running Windows or Linux, you will have to install **openVPN** client software from

https://openvpn.net/community-downloads/

If you are using Mac, it is recommended to use **tunnelblick.net**.

2. VPN Certificates will be send via email, based on the form you have fill today

ACCESS to AC922 / PowerAI

OpenPOWER™

Access to AC922 system is done via SSHv2 protocol:

Host IP Address: **10.7.12.50**
Username: **openpower'n'**

NOTE: n is a number from 1 – 18, assigned to each student that will complete system username.
Ex: ssh openpower1@10.7.12.50
ssh openpower18@10.7.12.50

ACCESS

OpenPOWER™

Username default password:

```
power2openpower
```

You will require to change the password after login:

```
$passwd
```

AC922 DEMO System

Available GPUs and allocation per student groups

GROUP-1
- openpower1
- openpower2
- openpower3
- openpower4
- openpower5

GROUP-2
- openpower6
- openpower7
- openpower8
- openpower9
- openpower10

GROUP-3
- openpower11
- openpower12
- openpower13
- openpower14
- openpower15

GROUP-4
- openpower16
- openpower17
- openpower18

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 418.39       Driver Version: 418.39       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| 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   | 00000004:04:00.0 Off |                    0 |
| N/A   36C    P0    40W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000004:05:00.0 Off |                    0 |
| N/A   38C    P0    43W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  On   | 00000035:03:00.0 Off |                    0 |
| N/A   36C    P0    40W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  On   | 00000035:04:00.0 Off |                    0 |
| N/A   39C    P0    39W / 300W |      0MiB / 32480MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

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

AC922 DEMO System

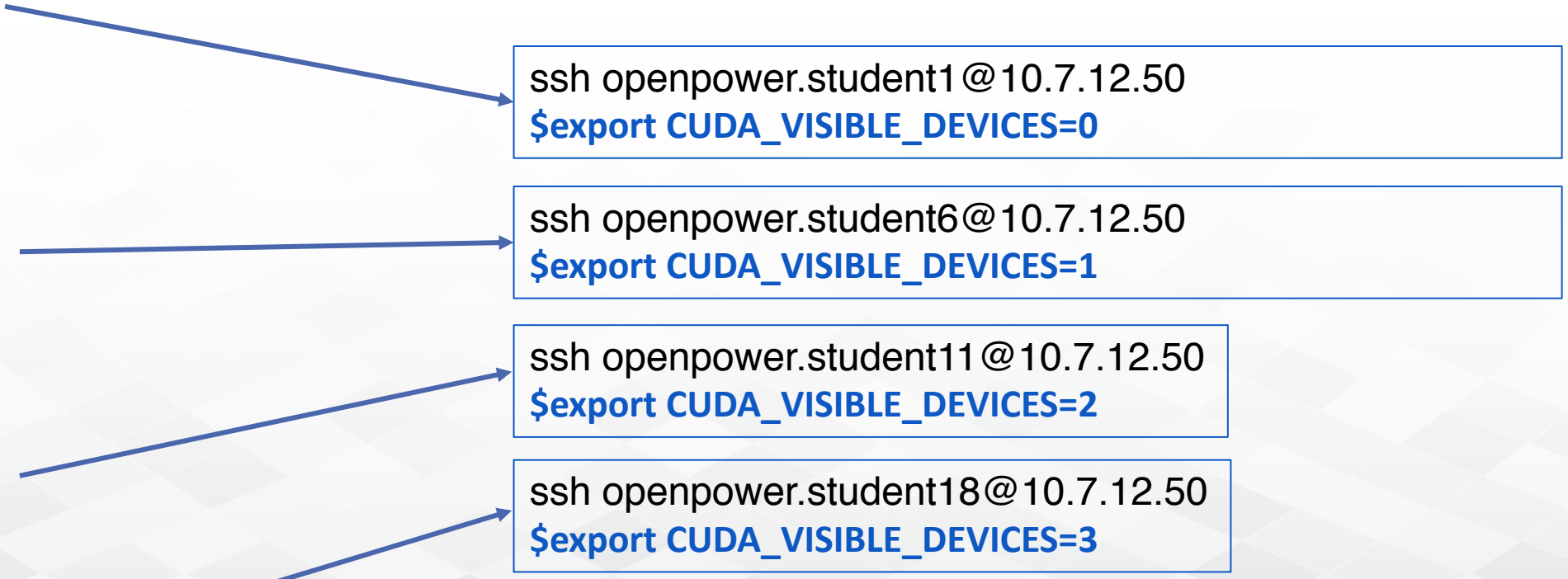
Available GPUs and allocation per student groups

GROUP-1
- openpower1
- openpower2
- openpower3
- openpower4
- openpower5

GROUP-2
- openpower6
- openpower7
- openpower8
- openpower9
- openpower10

GROUP-3
- openpower11
- openpower12
- openpower13
- openpower14
- openpower15

GROUP-4
- openpower16
- openpower17
- openpower18



Install PowerAI 1.6.0

Step 1:
wget https://repo.continuum.io/archive/Anaconda2-5.3.0-Linux-ppc64le.sh
or
wget https://repo.continuum.io/archive/Anaconda3-5.3.0-Linux-ppc64le.sh

Step 2:
sh Anaconda2-5.3.0-Linux-ppc64le.sh
or
sh Anaconda3-5.3.0-Linux-ppc64le.sh

NOTE: Do you wish the installer to initialize Anaconda3
in your /root/.bashrc ? [yes|no]
- type yes [ENTER]

Install PowerAI 1.6.0

Step 3:
source .bashrc

Step 4:
conda config --prepend channels \
https://public.dhe.ibm.com/ibmdl/export/pub/software/server/ibm-ai/conda/

Step 5:
conda create --name powerai python=3.6

Install PowerAI 1.6.0

Step 6:
conda activate powerai
conda install powerai
conda install cudf
conda install cuml

NOTE: Press Enter to continue viewing the license agreement, or, Enter "1" to accept the agreement, "2" to decline it or "99" to go back to the previous screen, "3" Print, "4" Read non-IBM terms.
- type 1 [ENTER]

Test TensorFlow Installation

Step 1 - get the install version

$python
>>> import tensorflow as tf
>>> print(tf.__version__)
1.13.0
>>> exit()

Step 2: Start your first training!

python /home/openpower”n”/anaconda3/enve/powerai/tensorflow-performance-models/scripts/tf_cnn_benchmarks/tf_cnn_benchmarks.py --batch_size=256 --num_batches=200 --num_warmup_batches=1 --data_format=NCHW --optimizer=sgd --variable_update=replicated --num_gpus=1 --model=alexnet

AC922 DEMO System

TensorFlow GPU limitations per student in a group

TensorFlow provides **two configuration** options on the session to control this.
The first is the allow_growth option when using sess.run(), which attempts to allocate only as much GPU memory based on runtime allocations, it starts out allocating very little memory, and as sessions get run and more GPU memory is needed, we extend the GPU memory region needed by the TensorFlow process.

1) Allow growth: (more flexible)

```
config = tf.ConfigProto()
config.gpu_options.allow_growth = True
session = tf.Session(config=config, ...)
```

2) Allocate fixed memory:

```
config = tf.ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.2
session = tf.Session(config=config, ...)
```

AC922 DEMO System

TensorFlow GPU limitations per student in a group

The second is when using the tf.Estimator framework the way to pass the fraction along to the implicitly created MonitoredTrainingSession

1) Estimator

```
opts = tf.GPUOptions(per_process_gpu_memory_fraction=0.200)
conf = tf.ConfigProto(gpu_options=opts) trainingConfig =
tf.estimator.RunConfig(session_config=conf, ...)
tf.estimator.Estimator(model_fn=..., config=trainingConfig)
```

2) Eager mode

```
opts = tf.GPUOptions(per_process_gpu_memory_fraction=0.200)
conf = tf.ConfigProto(gpu_options=opts)
tfe.enable_eager_execution(config=conf)
```

ACCESS to Jupiter Notebooks

OpenPOWER™

Access is done via SSHv2 tunneling:

From your Laptop:
$ ssh openpower'**n**'@10.7.12.50 -L1000**n**:localhost:1000**n**

From the AC922:
$ source /opt/DL/pytorch/bin/pytorch-activate
$ **j**upyter notebook –ip=127.0.0.1 –port=1010**n**

NOTE: n is a number from 1 – 20

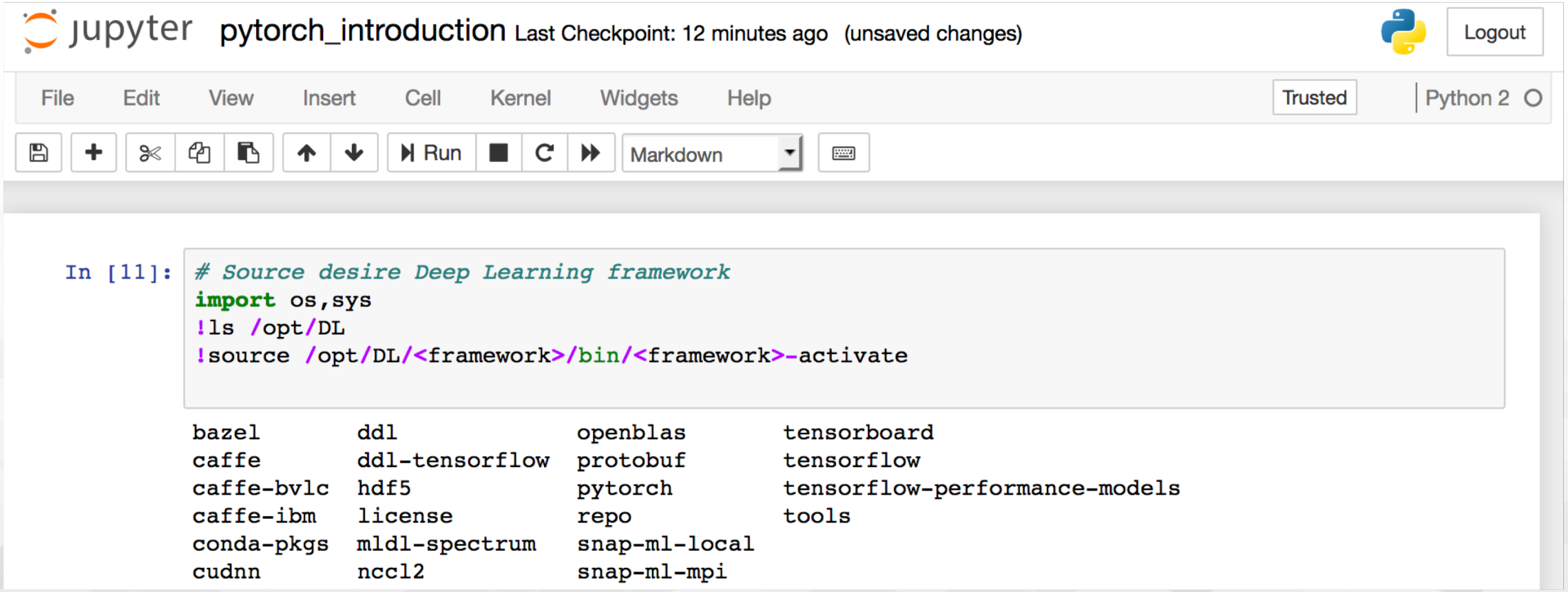
Ex: ssh openpower1@10.7.12.50 -L10001:localhost:10001
> jupyter notebook --ip=127.0.0.1 --port=10101

ssh openpower18@10.7.12.50 -L10018:localhost:10018
> jupyter notebook --ip=127.0.0.1 --port=10118

ACCESS to Jupiter Notebooks

OpenPOWER™

Point your workstation browser to: http://127.0.0.1:1000n//?token=<indicated_token_in_console>



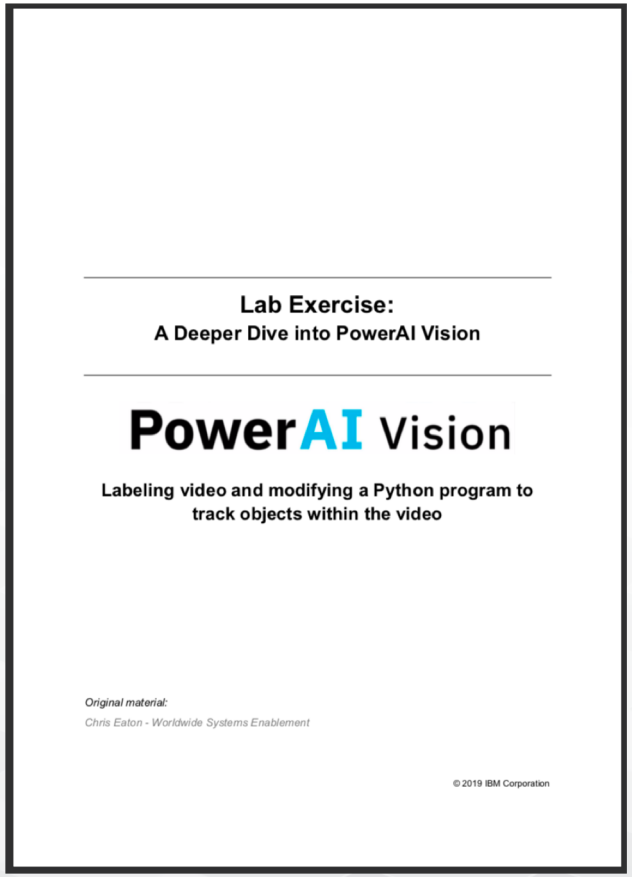
ACCESS to PowerAI Vision

OpenPOWER™

1. Read the LAB Guide
http://10.7.12.33

2. Connect to PowerAI Vision for Data Preparation, Labeling and Training
http://10.7.12.200

3. Inference and Post-Inference
Jupiter Notebooks (read LAB guide)


Lab Exercise:
A Deeper Dive into PowerAI Vision

PowerAI Vision

Labeling video and modifying a Python program to track objects within the video

Original material:
Chris Eaton - Worldwide Systems Enablement

ACCESS to PowerAI Vision

OpenPOWER™

1. PowerAI Vision URL:

https://10.7.12.200/powerai-vision-vision

2. PowerAI Vision Users:

tum01 / passw0rd
tum02 / passw0rd
tum03 / passw0rd
tum04 / passw0rd
tum05 / passw0rd
tum06 / passw0rd

IF YOU CAN CLEAN YOUR HOME DIRECTORY THEN DON'T MAKE IT DIRTY