OpenPOWER GPU Development Infrastructure

• Users get an account to access the head machine “openpower.cgrb.oregonstate.edu” using SSH/SCP/SFTP.

• Users can submit jobs using the SGE based scheduler system.

• Users can get command line access to individual processing machines with GPU processing and CAPI NVMe cards on processing hardware.

• Users can request more resources if needed by responding to the email generated when their account was created.
OpenPOWER GPU Development Infrastructure

openpower.cgrb.oregonstate.edu
(ssh port XXX)

(openpower)

(compute nodes)

(storage nodes)

/local/cluster/…
/raid1/…

commands ("Jobs")

RAM CPUS

RAM CPUS

RAM CPUS

RAM CPUS

RAM CPUS

RAM CPUS
Running Jobs on the OpenPOWER GPU Development Infrastructure

• Use “SGE_Avail” to determine what computational resources are available to your account.

• Submit jobs with “SGE_Batch” using quotes around the command you want to run

• Use “qstat” to monitor jobs that have been submitted to the cluster.

• If you would like to directly access a machine with a GPU use “qrsh” to check that machine out.

• Programs and tools are located under “/local/cluster” mount point.

• Tensorflow and other tools can be accessed through miniconda or can be installed by the user.
Welcome to the Center For Genome Research Genome Cluster

System Summary (collected Sun May 20 21:46:02 PDT 2018)
- Hostname = openpower
- CPU Usage (average) = 0%
- Memory in use (real) = 51%
- Total Memory = 565 GB
- Swap in use = 0 GB
- Total Swap = 7 GB
- Load Average = 0.23, 1.42, 3.02
- Total Space = 16 TB
- Total Free Local Space = 719.571 GB
- Total Used Local Space = 166.455 GB
- Private IP = 128.193.83.27
- Public IP = 128.193.83.27

--{ this information can be obtained by running the command systat }--

[[sullichr@openpower ~]$ uname -r
3.10.0-693.11.6.el7.ppc64le
[[sullichr@openpower ~]$ cat /etc/redhat-release
CentOS Linux release 7.4.1708 (AltArch)
Command To Run

Submit Job using "SGE_Batch"

Monitor Jobs with "qstat"

Example commands:

- To run a command:
  ```bash
date; sleep 20; date
  ```

- To submit the job using SGE_Batch:
  ```bash
  SGE_Batch -c 'date; sleep 20; date' -r Date_Sleep
  ```

- To monitor the jobs with qstat:
  ```bash
  qstat
  ```
Tensorflow Example

```python
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
name: Tesla P100-SXM2-16GB
major: 6 minor: 8 memoryClockRate (GHz) 1.4885
pciBusID 0000:01:00.0
Total memory: 15.839GiB
Free memory: 15.60GiB
exists before initializing the StreamExecutor. We haven't verified StreamExecutor works with that.
name: Tesla P100-SXM2-16GB
major: 6 minor: 9 memoryClockRate (GHz) 1.4885
pciBusID 0000:01:00.1
Total memory: 15.839GiB
Free memory: 15.60GiB
2018-05-29 22:25:11.906110: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 0 and 1
2018-05-29 22:25:11.906151: I tensorflow/core/common_runtime/gpu/gpu_device.cc:832] Peer access not supported between device ordinals 1 and 0
2018-05-29 22:25:11.908088: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1038] Creating TensorFlow device (/gpu:0) -> (device: 0, name: Tesla P100-SXM2-16GB, pci bus id: 0000:01:00.0)
>>> print(sess.run(hello))
Hello, TensorFlow!
```