XSEDE Quick Start

The NSF-funded Extreme Science and Engineering Discovery Environment (XSEDE) is a single virtual system that scientists can use to interactively share computing resources, data and expertise. The **Bridges** server is where you can manage your data, code and jobs (like your CSE online server). Here is a Quick Start user guide. You need to complete the following steps to learn how to use Bridges.

1. Registration

Sign up at the XSEDE website first, then let Becky (rjp49@psu.edu) know your account, so that you can be added to the NLP class project group that has access to the Bridges GPU server.

2. Connect to the Bridges

Connect to the server using either XSEDE or PSC credentials via **SSH**. If you are registered with XSEDE for DUO Multi-Factor Authentication (MFA), you can use this security feature in connecting to Bridges.

See the XSEDE instructions to set up DUO for MFA, first. Plus, you can use other ways to connect to the server, as found in this link (See *Connecting to Bridges*).

```
ssh -l <username> bridges.psc.xsede.org
```

```
ssh -l <username> bridges.psc.edu
```

3. Account Administration

The `projects` command will help you monitor your allocation on Bridges. You can determine what Bridges resources you have been allocated, your remaining balance, your account id (used to track usage), and more. The output below shows that this user has an allocation on Bridges AI and Bridges Large resources for computing and Bridges Pylon for file storage.

```
projects
```

```
[zjl5282@login018 test]$ projects
Your default charging project charge id  is cc5pijp.  If you would like to change t
he default charging project use the command change_primary_group ~charge_id~. Use t
he charge id listed below for the project you would like to make the default in pla
ce of ~charge_id~

Project: CCR190035P
     PI: Rebecca Passonneau
  Title: Penn State Natural Language Processing Course: Computing Resources Request

      Resource: BRIDGES GPU
    Allocation: 32,000.00
       Balance: 32,000.00
      End Date: 2020-11-24
  Award Active: Yes
   User Active: Yes
     Charge ID: cc5pijp
     *** Default charging project ***
   Directories:
       HOME /home/zjl5282

      Resource: BRIDGES PYLON STORAGE
    Allocation: 500.00
       Balance: 500.00
      End Date: 2020-11-24
  Award Active: Yes
   User Active: Yes
     Charge ID: cc5pijp
   Directories:
       Lustre Project Storage /pylon5/cc5pijp
       Lustre Storage /pylon5/cc5pijp/zjl5282

-------------------------------------------------
```

Accounting for Bridges use varies with the type of node used, which is determined by the type of allocation you have: "Bridges GPU", for Bridges' K80 and P100 GPU nodes;

4. Transferring Files

For all file transfer methods other than *cp*, you must always use the full path for your

Bridges files. The start of the full paths for your Bridges directories are:

Home directory /home/<username>
Pylon5 directory /pylon5/Unix-group/<username>

scp

To use scp for a file transfer, you must specify a source and destination for your transfer. The format for either source or destination is
username@machine-name:path/filename
For transfers involving Bridges, username is your PSC username. The machine-name should be given as data.bridges.psc.edu. This is the name for a high-speed data connector at PSC. We recommend using it for all file transfers using scp involving Bridges. Using it prevents file transfers from disrupting interactive use on Bridges login nodes.

```
scp -r filename rjp49@data.bridges.psc.edu:/home/rjp49/
```

sftp

To use sftp, first connect to the remote machine: *sftp username@machine-name*
When Bridges is the remote machine, use your PSC userid as username. The Bridges machine-name should be specified as data.bridges.psc.edu. This is the name for a high-speed data connector at PSC. We recommend using it for all file transfers using sftp involving Bridges. Using it prevents file transfers from disrupting interactive use on Bridges login nodes.You will be prompted for your password on the remote machine. If Bridges is the remote machine enter your PSC password.You can then enter sftp subcommands, like put to copy a file from the local system to the remote system, or get to copy a file from the remote system to the local system.

5. Programming Environment

Bridges provides a rich programming environment for the development of applications. Here anaconda is used as example; see more about anaconda or other software at the Pittsburgh Supercomputing Center. Multiple versions of anaconda are available on

Bridges. In addition, environments tailored for AI applications which include `anaconda` and other popular AI/Machine Learning/Big Data packages, such as TensorFlow, Theano, Keras, pandas, opencv, scikit-learn, and more are set up for you to use.

Be sure to check the help for each version to see useful information about how that version was compiled and whether there are any conflicts with other modules. Type

```
module help anaconda-version
```

Various add-on packages such as numpy, scipy, and the MKL libraries are installed with individual versions. You can see what is available in a given version by first loading the module and then using the `pip` command:

```
module load anaconda-version
```

Note that anaconda**2** modules use python**2** and anaconda**3** modules use python**3**. The configuration is similar when you use GPU functions CUDA. To use CUDA, first you must load the CUDA module. To see all versions of CUDA that are available, type:

```
module avail cuda
```

Then choose the version that you need and load the module for it.

```
module load cuda/8.0
```

CUDA 8 code should run on both types of Bridges GPU nodes with no issues. CUDA 7 should only be used on the K80 GPUs (Phase 1). Performance may suffer with CUDA 7 on the P100 nodes (Phase 2).

6. Running Jobs

You can run jobs in Bridges in several ways:

- interactive mode - where you type commands and receive output back to your

screen as the commands complete

- batch mode - where you first create a batch (or job) script which contains the commands to be run, then submit the job to be run as soon as resources are available
- through OnDemand - a browser interface that allows you to run interactively, or create, edit and submit batch jobs and also provides a graphical interface to tools like RStudio, Jupyter notebooks, and IJulia. See the OnDemand section for more information.

1) Interactive mode

You can do your production work interactively on Bridges, typing commands on the command line, and getting responses back in real time. But you must be allocated the use of one or more Bridges' compute nodes by SLURM to work interactively on Bridges. You cannot use the Bridges login nodes for your work.

Sample `interact` command for GPU

```
interact --gpu
```

Then you are in a GPU node where you can run your python jobs. Furthermore, the GPU statu can be checked by the following command:

```
python
```

```
In [1]: import torch

In [2]: torch.cuda.current_device()
Out[2]: 0

In [3]: torch.cuda.device(0)
Out[3]: <torch.cuda.device at 0x7efce0b03be0>

In [4]: torch.cuda.device_count()
Out[4]: 1
```

```
In [5]: torch.cuda.get_device_name(0)
Out[5]: 'Tesla P100-PCIE-16GB'

In [6]: torch.cuda.is_available()
Out[6]: True
```

A more specific command to start a GPU job on 4 P100 nodes for 30 minutes is

```
interact -p GPU gres=gpu:p100:2 N 4 t 30:00
```

where:

- "p" indicates the intended partition
- "gres=gpu:p100:2" requests the use of 2 P100 GPUs
- "N" requests 4 nodes
- "t" is the walltime requested in the format HH:MM:SS

2) Batch jobs

To run a batch job, you must first create a batch (or job) script, and then submit the script using the sbatch command.

A batch script is a file that consists of SBATCH directives, executable commands, and comments. SBATCH directives specify your resource requests and other job options in your batch script. You can also specify resource requests and options on the sbatch command line. Any options on the command line take precedence over those given in the batch script. The SBATCH directives must start with "#SBATCH" as the first text on a line, with no leading spaces.Comments begin with a '#' character.The first line of any batch script must indicate the shell to use for your batch job.

Sample `sbatch` command for GPU

This command requests the use of one K80 GPU node for 45 minutes:

```
sbatch p GPU gres=gpu:k80:4 N 1 t 45:00 myscript.job
```

where:

- "`p`" indicates the intended partition
- "`gres=gpu:k80:4`" requests the use of 4 K80 GPUs
- "`N`" requests one node
- "`t`" is the walltime requested in the format HH:MM:SS
- "`myscript.job`" is the name of your batch script

Sample bash script for GPU partition

```
#!/bin/bash
#SBATCH N 2
#SBATCH p GPU
#SBATCH ntaskspernode 28
#SBATCH t 5:00:00
#SBATCH gres=gpu:p100:2

#echo commands to stdout
set x

#move to working directory
# this job assumes:
#  all input data is stored in this directory
#  all output should be stored in this directory
cd /pylon5/groupname/username/pathtodirectory

#run GPU program
./mygpu
```

Notes: The value of the --gres-gpu option indicates the type and number of GPUs you want.For *groupname, username* and *path-to-directory* you must substitute your group, username and appropriate directory path.