Computer Engineering Cluster

CMPE 685 Teaching Assistant: Nilesh Pandey np9207@rit.edu

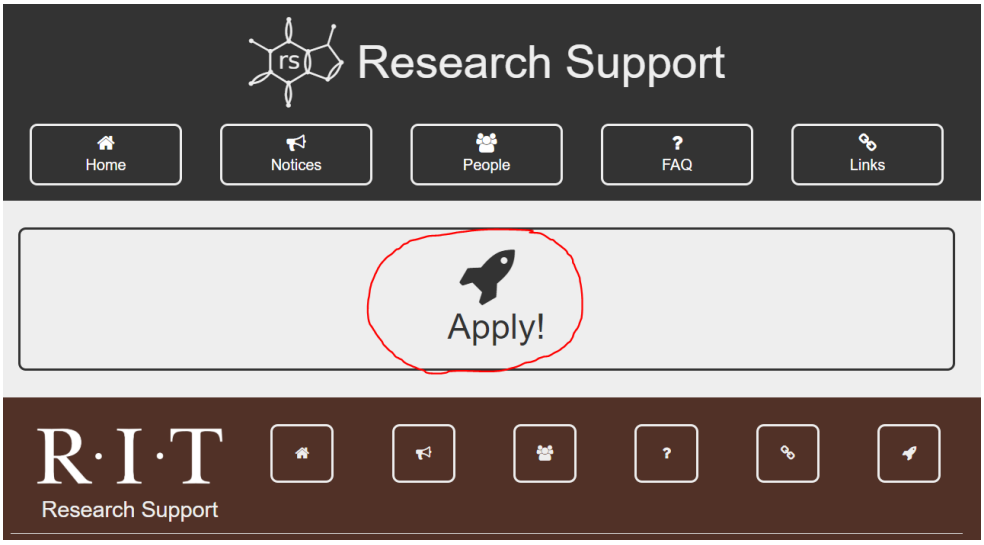
Accessing the Cluster

By: Richard Flegal and Nilesh Pandey, RIT Computer Engineering

A. Gaining Access

You can work on one of the workstations in the following labs: VLSI lab (CE), DCO lab (CE), Embedded Systems lab (CE), or the CEDA lab (EE).

1. Apply for a Research Computing Account at http://apply.rc.rit.edu



2. Login using your RIT ID

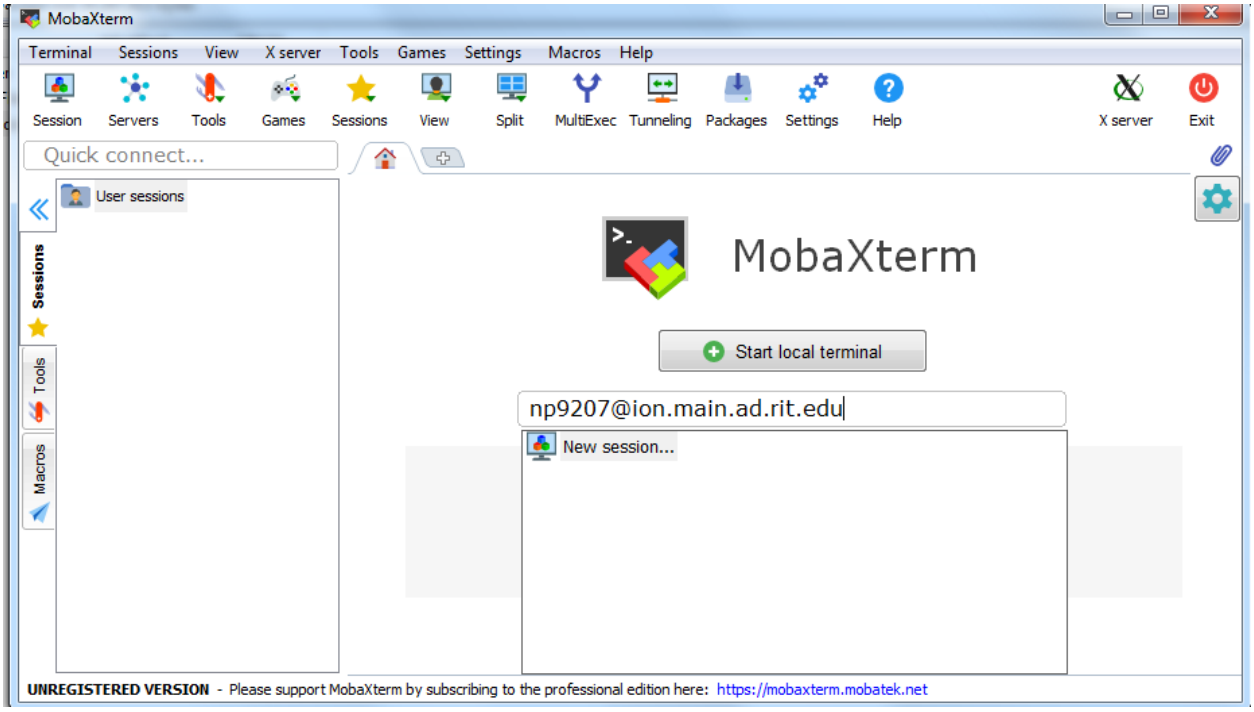
B. You should get an email confirming your access to the computer cluster

Computer Engineering Cluster

II. SSH Setup

A. Windows

1. Mobaxterm: Download Mobaxterm



2. Enter username@kgcoe-cluster01.main.ad.rit.edu or username@kgcoe-cluster02.main.ad.rit.edu
3. Click on ssh and ok to start the server.

You can upload python code, folder, and download the saved result directly from any windows machine.

Your server directory should contain

1. Data (Images) on which you are working.
2. Python code (This is code file belonging to the project.)
3. run.sh file (This bash file will load and run the following python code in your directory.)

You will be required to run **bash run.sh** to execute your program.

B. Linux/Mac

(You are required to use one of the KGCOE lab machines (on RIT network))

1. Open Terminal
2. ssh username@kgcoe-cluster01.main.ad.rit.edu or username@kgcoe-cluster02.main.ad.rit.edu

3. Enter the password to start the server
4. Open another terminal and type **hostname**
5. Switch to the terminal where ssh to the server is running.
6. cd to the directory where results are saved
7. Enter **sftp hostname.main.ad.rit.edu**
8. Type command **put result_img.jpg**
9. The result would be stored in the home directory of your host machine.

Job Scheduling

A. Checking Cluster status

1. Squeue

   Once you've logged into the cluster head node, you can check the status of the cluster by issuing the squeue command:

```
[axf2476@ion ~]$ squeue
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
13970165_[1-10000]      work run_prot   crrvcs PD       0:00      1 (Priority)
13970166_[1-10000]      work run_prot   crrvcs PD       0:00      1 (Priority)
13970167_[1-10000]      work run_prot   crrvcs PD       0:00      1 (Priority)
13970164_[6885-100      work run_prot   crrvcs PD       0:00      1 (Resources)
          14124070      work my_prog_   fxcsbi  R    5:50:22      1 magnycours-03
          14124058      work my_prog_   fxcsbi  R    5:50:26      1 magnycours-02
          14124051      work my_prog_   fxcsbi  R    5:50:27      1 magnycours-02
          14124046      work my_prog_   fxcsbi  R    5:50:29      1 interlagos-02
          14124042      work my_prog_   fxcsbi  R    5:50:31      1 interlagos-02
          14124036      work my_prog_   fxcsbi  R    5:50:33      1 magnycours-03
          14124023      work my_prog_   fxcsbi  R    5:50:37      1 interlagos-02
          14124024      work my_prog_   fxcsbi  R    5:50:37      1 magnycours-03
          14124015      work my_prog_   fxcsbi  R    5:50:42      1 haswell-01
          14124009      work my_prog_   fxcsbi  R    5:50:45      1 interlagos-01
          14124004      work my_prog_   fxcsbi  R    5:50:46      1 westmere-02
          14123999      work my_prog_   fxcsbi  R    5:50:48      1 magnycours-01
          14123997      work my_prog_   fxcsbi  R    5:50:50      1 magnycours-03
     13970164_6883      work run_prot   crrvcs  R       0:00      1 westmere-01
     13970164_6884      work run_prot   crrvcs  R       0:00      1 interlagos-02
```

The jobs that have an R in the ST column are currently in the running state. They are executing on the nodes shown. The rest of the jobs have a PD in the ST column meaning they are in a pending state. They are pending for different reasons – some do not have sufficient priority to be running yet whereas another is marked as requesting resources that are not yet available.
The USER column indicates what user owns the submitted job.
The TIME column indicates how long the jobs have been running.

2. Sinfo

   The command sinfo shows information on the partitions available for use.

Computer Engineering Cluster

```
[axf2476@ion ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
work*        up 20-00:00:0      1  drain interlagos-03
work*        up 20-00:00:0     18    mix interlagos-01,magnycours-[01-02],westmere-[01-02],woodcrest-[01,07,10-11,13-15,17-18,20-23]
work*        up 20-00:00:0      7  alloc haswell-01,interlagos-02,magnycours-03,woodcrest-[02,08-09,16]
debug        up    6:00:00      1  drain interlagos-03
debug        up    6:00:00     18    mix interlagos-01,magnycours-[01-02],westmere-[01-02],woodcrest-[01,07,10-11,13-15,17-18,20-23]
debug        up    6:00:00      7  alloc haswell-01,interlagos-02,magnycours-03,woodcrest-[02,08-09,16]
kgcoe-gpu    up 10-00:00:0      3   idle cluster-gpu-[01-03]
```

B. Running a Job

*****Important to note*****
If **sinteractive** fails to load because of server issues, you can still run the command **bash run.sh** to run your program with the default settings.

Using **sinteractive** (when it is ready by sysadmin)

a. When using CPU only, you need to use **sinteractive** command. (Project 0, Project 1, and Project 2 don't require the use of GPU).
b. When using a GPU, type: **sinteractive --gres=gpu**

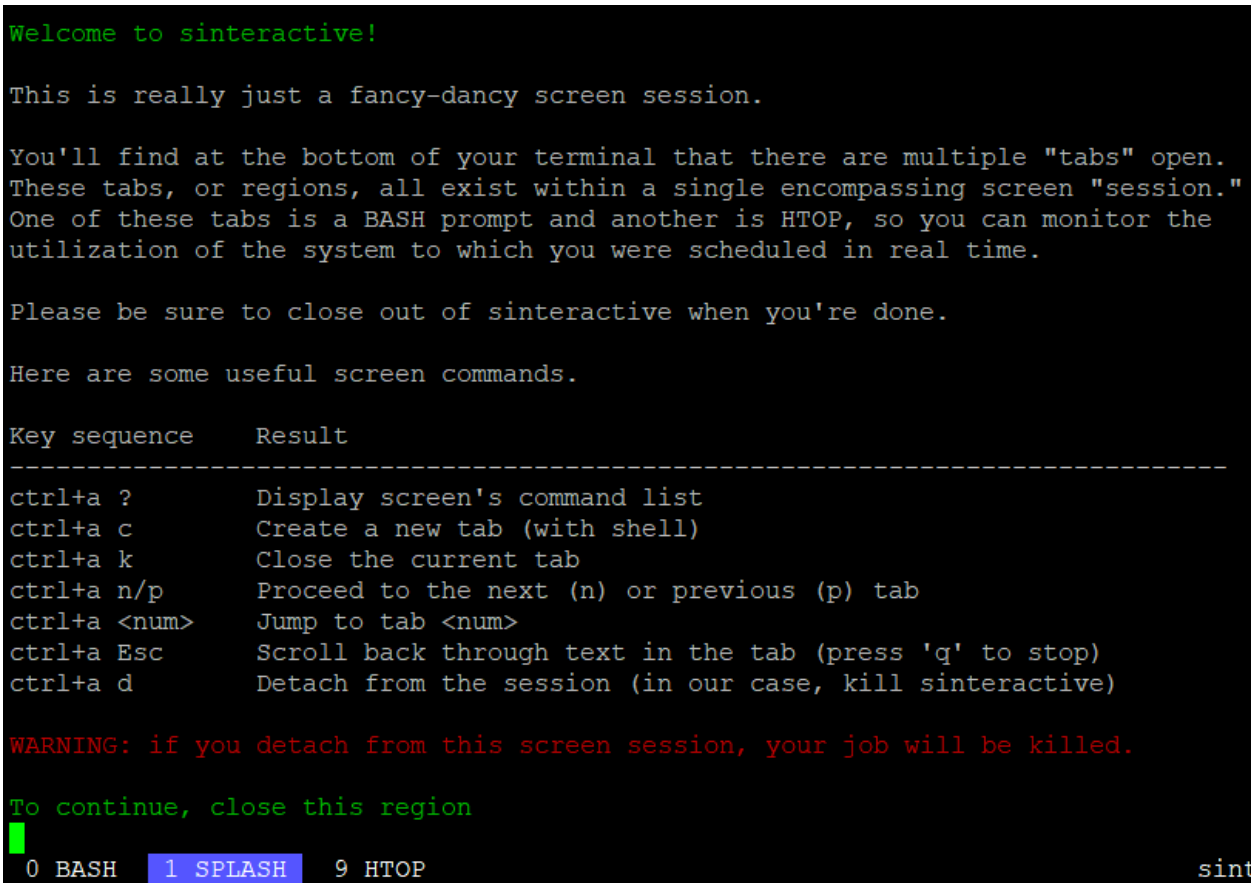
```
[axf2476@ion ~]$ sinteractive --gres=gpu
INFO: >Please enter your resource requirements.<

How many CPU cores? :: 2
How much memory (in MB)? :: 4096
How many minutes do you need? :: 120
What QOS? :: free
What partition? :: kgcoe-gpu

WARN: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
WARN: !! If you detach from the sinteractive  !!
WARN: !! screen session, it _WILL_ be killed. !!
WARN: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

INFO: Spawning job with the following resources:
INFO:      Processors: 2
INFO:          Memory: 4096 MB
INFO:   Runtime limit: 120 minutes
INFO:             QOS: free
INFO:       Partition: kgcoe-gpu

INFO: Submitting job...
INFO: Waiting for JOBID 13629275 to start
```

The session should alert you with an error message, if you are requesting resources it can't provide. If your requested resources are in use, it will schedule your job for a specific time when they are available to you.

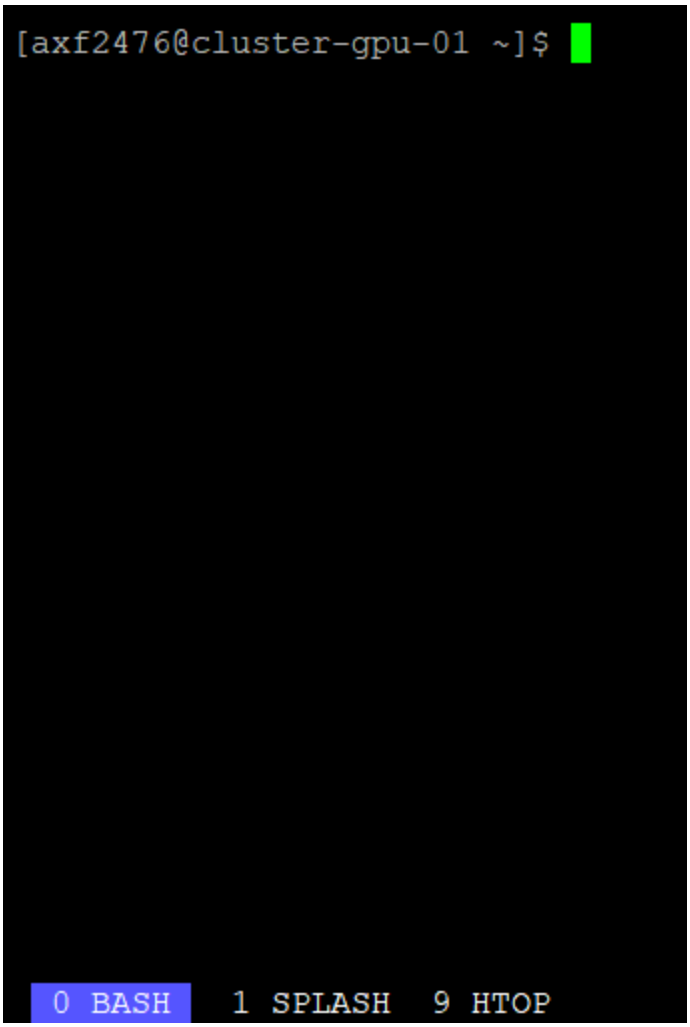
The partition should be kgcoe-gpu even when you're working with cpu only.

Computer Engineering Cluster

If kgcoe-gpu throws error, the code can still be run with the default settings just like you run on the local machine.



The following screen appears upon successful submission of a job and provides the commands necessary for your session.



On tab 0 you can run commands such a module load [module] and run your tasks in real-time with your requested resources.

Remember, once you are done with your sinteractive session to exit out of it so those resources are available to others.

C. Loading and Using Modules

1. Loading a module for use

The command: module load <program>
will work to load available modules in the tools folder module avail
The above command will give a list of all modules available for use.

Computer Engineering Cluster

```
[np9207@kgcoe-cluster02 ~]$ clear

[np9207@kgcoe-cluster02 ~]$ module avail

------------------------ /usr/share/Modules/modulefiles ------------------------
dot         module-git  module-info modules     null        use.own

------------------------------ /etc/modulefiles ------------------------------
openmpi-1.10-x86_64 tools-dir

----------------------- /tools/Modules/modulefiles.meta -----------------------
module_archive module_default module_rhel6   module_slurm
module_class   module_future  module_rhel7

----------------------- /tools/Modules/modulefiles.slurm ----------------------
slurm-partition/kgcoe-gpu slurm-partition/kgcoe-mps slurm-partition/work

------------------------- /tools/Modules/modulefiles --------------------------
altera/9.0                      libvorbis/1.3.3
ampl/20161005                   libx264/20170205
ampl/20161231                   libxml2/2.9.4
ampl/20180125                   libxslt/1.1.29
anaconda/3.16                   lr_trirls/2006
ansys/15.0.7                    lumerical-device/6.0.1157
ansys/16.1.0                    lumerical-fdtd/8.11.318
ansys/18.1.0                    lumerical-fdtd/8.15.786
ansys-em/17.0.0                 lumerical-fdtd/8.17.1057
ansys-em/18.1                   lumerical-fdtd/8.17.1102
ansys-em/19.2                   lumerical-fdtd/8.17.1157
autoconf/2.69                   lumerical-interconnect/6.5.1102
automake/1.14                   lumerical-interconnect/6.5.1157
benchmarks                      lumerical-mode/7.9.1102
boost/1.48.0                    lumerical-mode/7.9.1157
boost/1.58.0                    macs/1.4.2
bowtie/2.2.9                    mathematica/10.0
caffe/1.0.0-rc3                 matio/1.5.6
caffe/2016-03-08(default)       matlab/2016b
caffe-cpu/2015-06-25            matlab/2017a
caffe-cpu/20170302-standalone   matlab/2017b
calibre/2015.3(default)         matlab/2018a
calibre/2016.4                  matlab/2018b(default)
cflow/1.4                       medea/2.22.1
clion/1.1                       mentor/170109
cmake/3.11.0(default)           mentor/current(default)
cmake/3.7.1                     module_archive
comsol/microsystems-50          module_class
comsol/microsystems-52a         module_future
crystal/0.24.2                  mplayer/1.1.1
CST/2015                        mricron/201212
cuda/6.5(default)               ncdu-fast/1.11g
cuda/7.5                        omnetpp/4.6
cudnn/5.1                       openblas/0.2.14
cudnn/6.5                       opencv/2.4.11
```

Computer Engineering Cluster

```
File  Edit  View  Search  Terminal  Help
ffmpeg/3.2.2(default)           pycharm/4.5.3
font_server/1.0.2               python/2.7.10
freesurfer/5.3.0                python/2.7.12
g95/0.90                        python/2.7.9
gflags/2.1.2                    python/3.5.2
git/2.10.0                      pyxis/10.4
git/2.11.0                      pyxis/10.5(default)
git/2.15.1                      pyxis/10.5.7
glog/0.3.4                      rsoft/2018.06
glpk/4.45                       rubymine/7.1.4
gnuplot/5.0.1                   rvm/default
gnuplot/5.0.5                   samtools/0.1.17
go/1.10.1                       samtools/1.3.1
go/1.10.3(default)              scala/2.11.7
go/1.9.0                        scip/3.1.1
go/1.9.2                        scons/2.1.0
gtextutils/0.6                  scons/2.3.4
gurobi/4.6.1                    sentaurus/J-2014.09_SP2
gurobi/5.1.0                    silvaco/multi
harpoon/3.6a                    slang/2.3.1a
htslib/1.3.1                    snappy/2015-06-24
idea/14.1.4                     speex/1.2.0
IntelClusterSuite/2013xe        sqlitefs/1.3.1
java/8u101(default)             swig/2.0.11
java/8u40                       swig/3.0.12
jpeg-turbo/1.4.2                synopsys/160805
julia/0.3.11                    synopsys/current
julia/0.4.6                     system_defaults
julia/0.5.1                     tophat/1.3.1
kaldi/latest                    tophat/2.1.1
knitro/7.0.0                    torch7/0.0.7(default)
lame/3.99.5                     vtk/5.10.1
lammps/20160928                 xilinx/14.7
libfreetype/2.4.11-15           xml-coreutils/0.8.1
libglvnd-glx/1.0.1-0.1          xvidcore/1.3.4
libogg/1.3.1                    zimpl/3.3.2
libtheora/1.1.1

---------------------- /tools/Modules/modulefiles.rhel6 ----------------------
boost/1.57.0        gcc/4.7.4           qt/5.4.2
cadence/160805      gcc/4.8.2           qt/5.5.0
cadence/current     gcc/4.9.4           radare/2.3.0
chuffed/0.9.0       gmp/6.1.2           ruby/2.2.3
ctorrent/dnh3.3.2-1 idba/1.1.3          ruby/2.4.0(default)
dsistudio/20170706  itk/4.12.2          spades/3.10.1
dsistudio/20180720  llvm/4.0.0          usearch/9.2.64
flint/2.5.2         llvm/5.0.0          vim/8.0.1755
fsl/5.0.10          minizinc/2.1.7      vivado/2017.1
gcc/4.4             mpfr/4.0.1
gcc/4.6.4           or-tools/6.7.1

---------------------- /tools/Modules/modulefiles.class ----------------------
cmpe530
[np9207@kgcoe-cluster02 ~]$ module load python/3.5.2
[np9207@kgcoe-cluster02 ~]$
```

Example bash file (run.sh):

```
#!/bin/bash -l

# NOTE the -l flag!
#
# Please copy this file to your home directory and modify it to suit your needs.
#
# If you need any help, please email rc-help@rit.edu
#

echo "(${HOSTNAME}) CV project 0" # name of the project
module load python/3.5.2 # load the required module necessary to run code
python3 ./project1_codes/part3c.py # Path to the program you will be running
echo "(${HOSTNAME}) Done!" # Display message.
```

If you run an *sinteractive* session and the requested resources aren't available it will schedule your job for a time those resources are available.

D. Load Deep Learning Frameworks

On the server:

1. Run the following commands
   module load module_future
   module load singularity
2. Example load pytorch
   singularity run /opt/singularity/images/pytorch.simg

This will load pytorch 0.30 with python 3.6.

To check available images in singularity
Go to **opt/singularity/images**

For more information please visit Singularity