Intro to Jarvis

Scott Krieder & Eduardo Berrocal

Outline

- SSH Connections
  - How to login
  - passwordless ssh
- Linux Basics
  - cd ls mkdir touch
  - TMUX
  - Makefile
- Command Line Editors
  - emacs, vim
- Version Control
- Jarvis cluster
  - Compilers
- SGE
  - How to submit jobs
  - How to run interactive jobs

Login to jarvis cluster

- Linux & MacOS

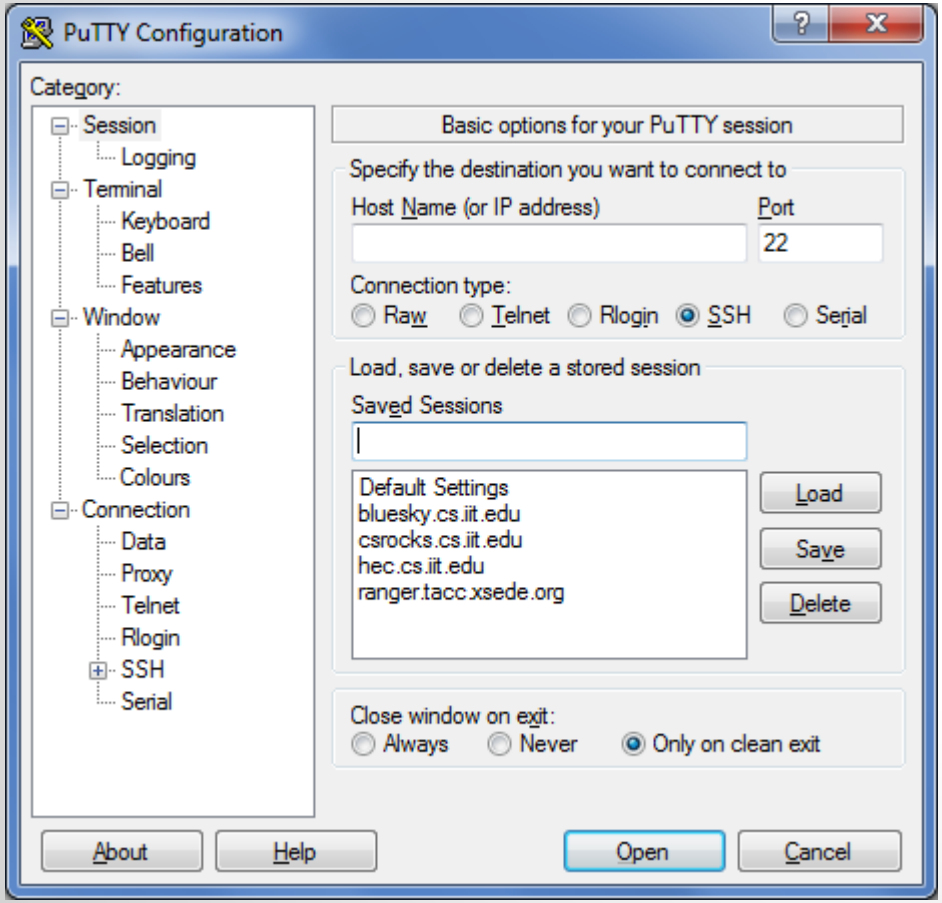
  Add the following to ~/.ssh/config:

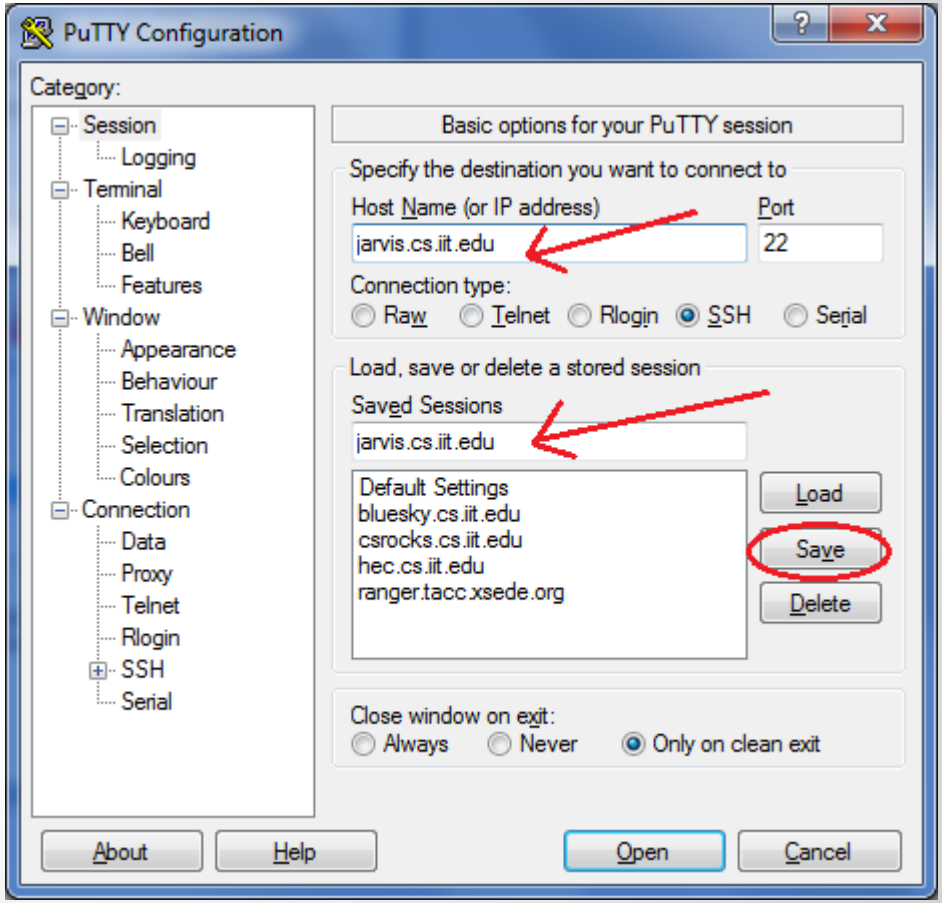
```
Host jarvis
    HostName jarvis.cs.iit.edu
    User iit_username
    IdentityFile ~/.ssh/id_rsa
```

  - **Login:** ssh *jarvis*
  - **CP Files:** scp **file** *jarvis:~/*

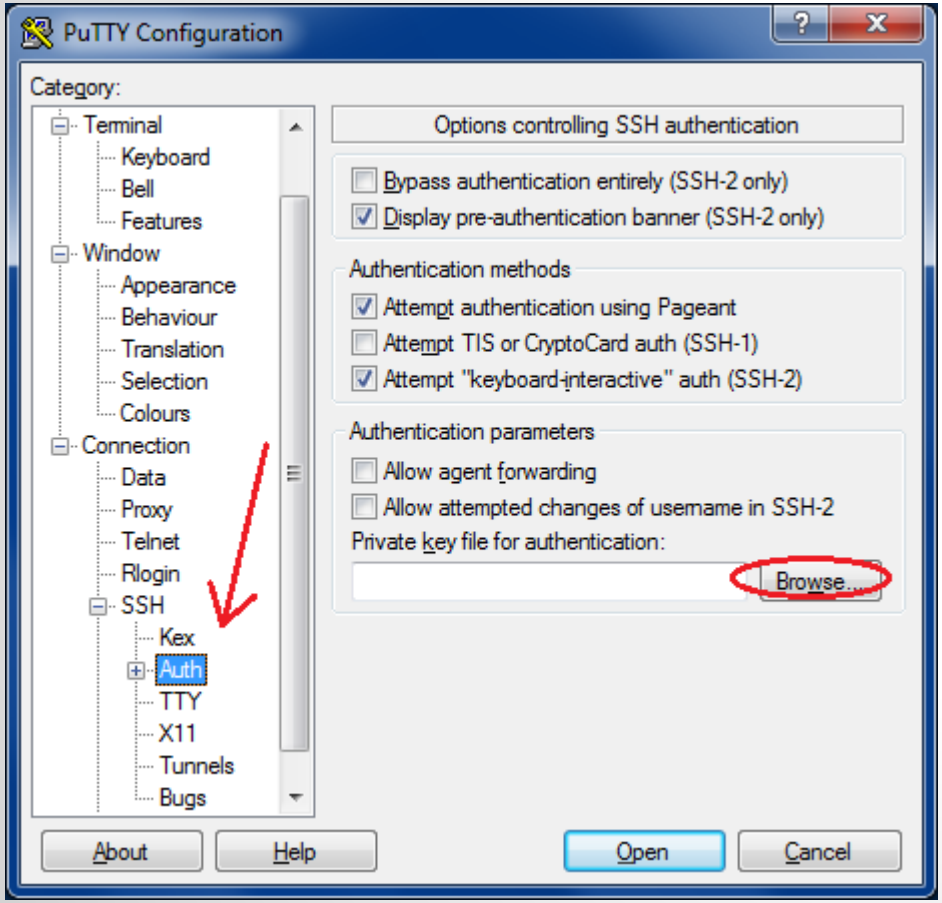
- Windows
  - PuTTY: http://www.putty.org/
  - WinSCP: http://winscp.net/eng/download.php

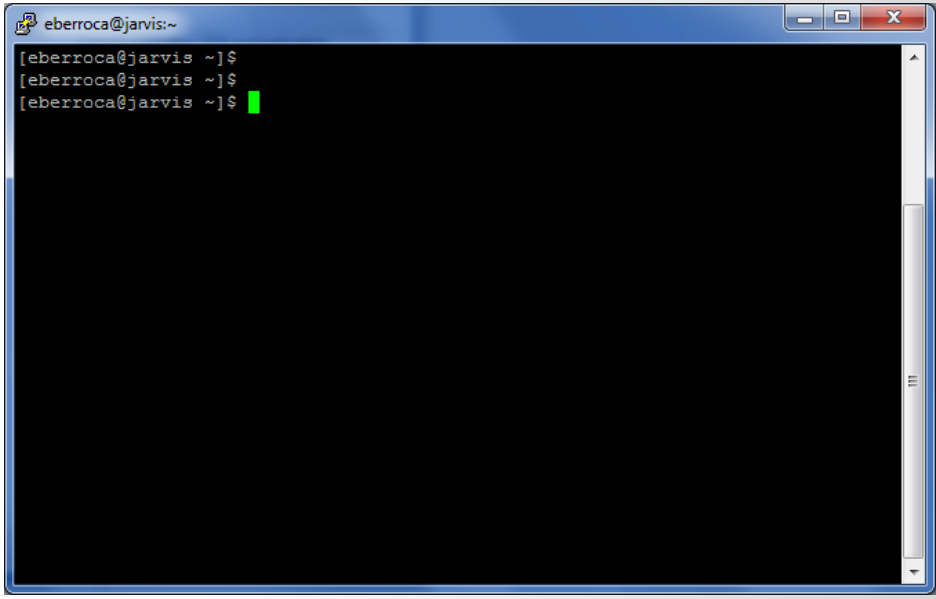
Login to jarvis (windows)



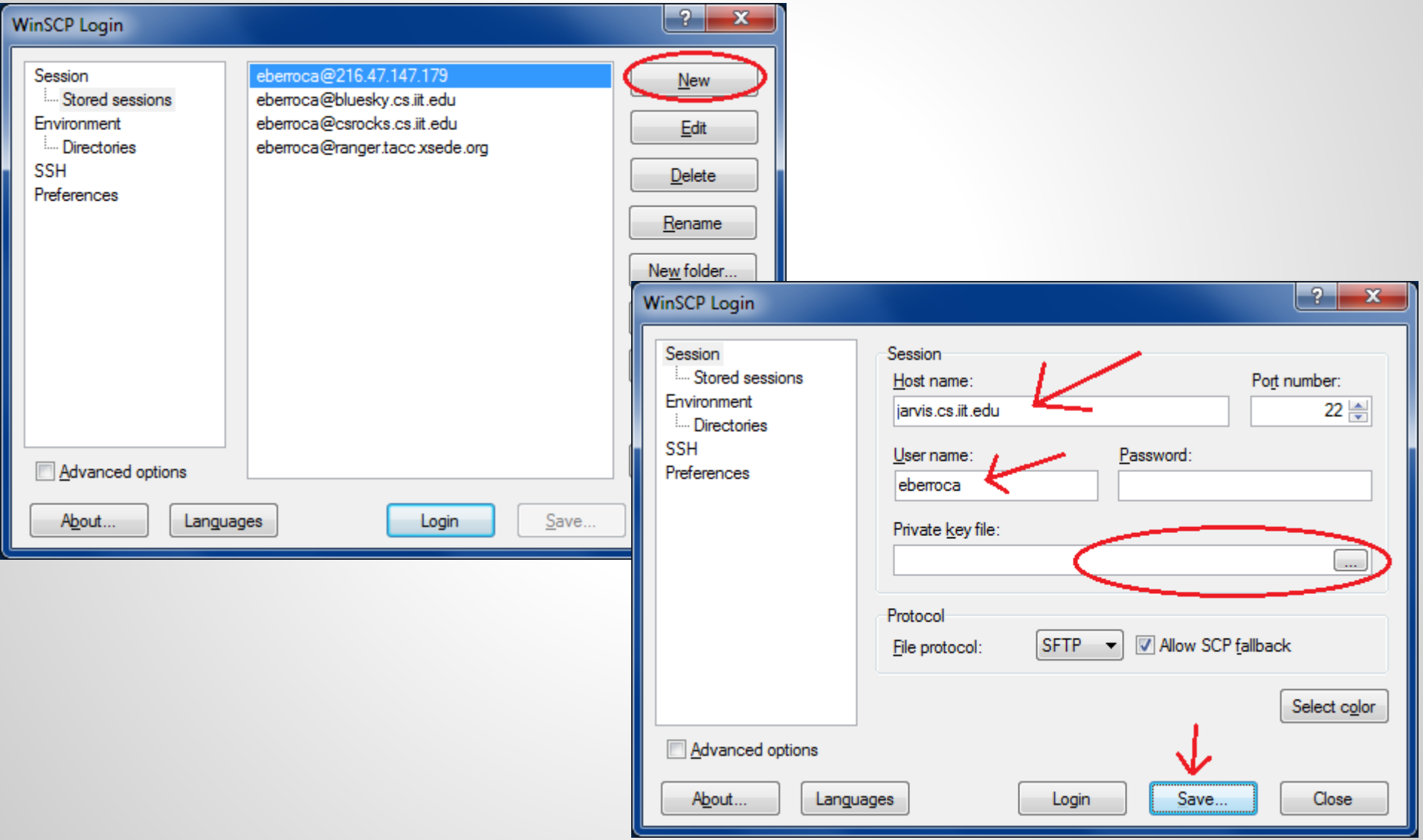


Login to jarvis (windows)

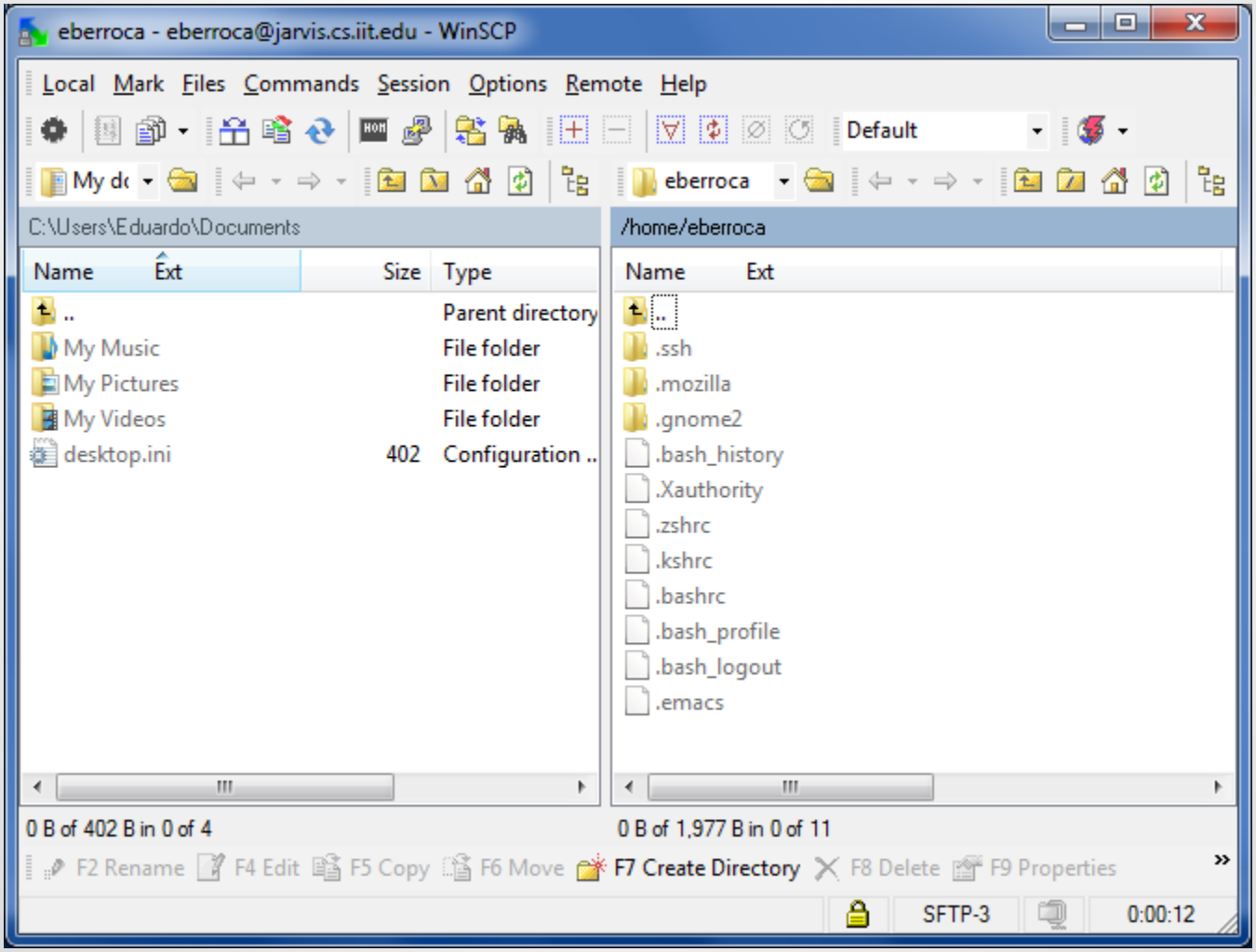




Copying files to jarvis (windows)



Copying files to jarvis (windows)



Linux Basics

- ls
- cd
- mkdir
- Touch
- Makefile

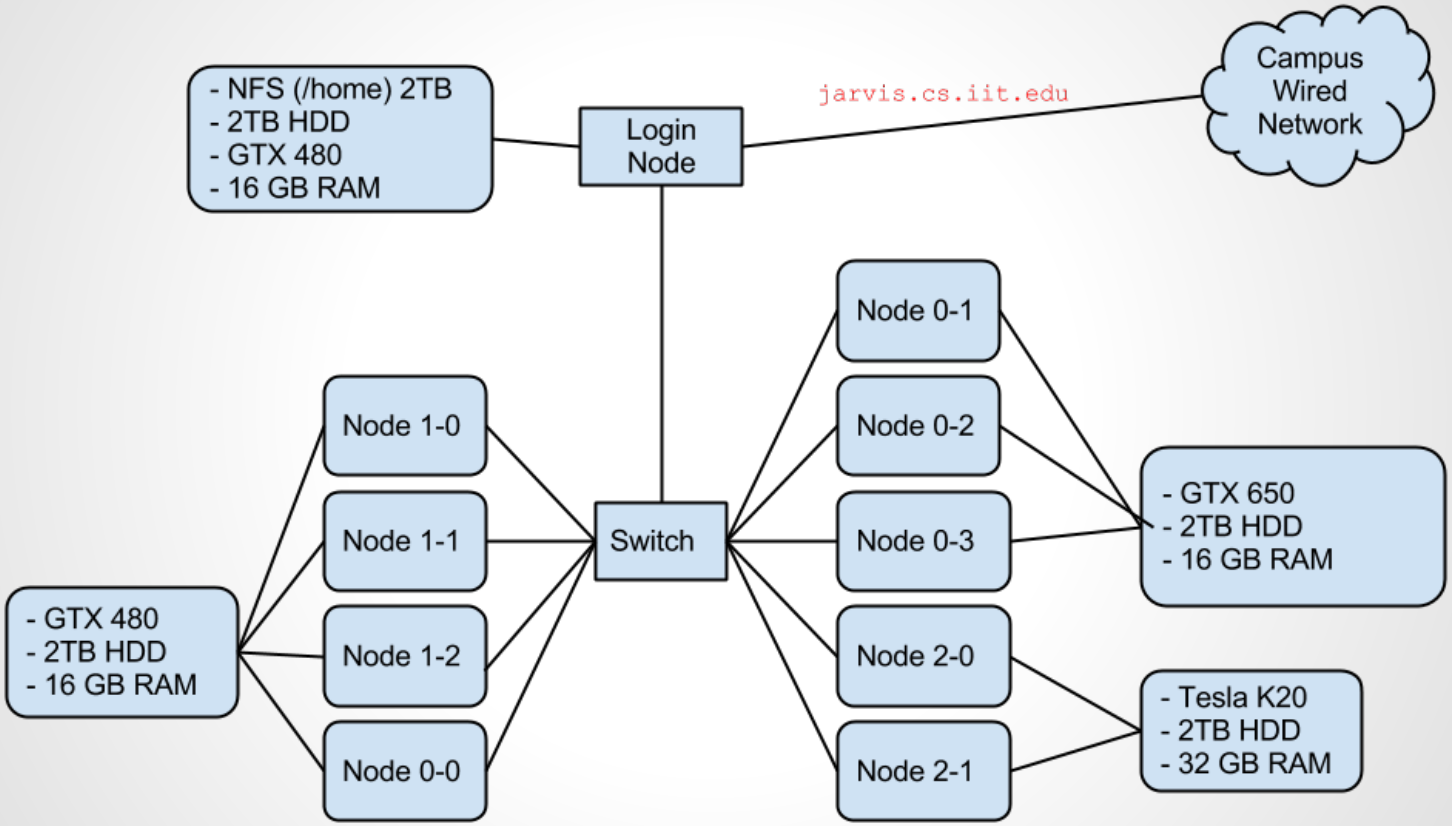
Version Control

- Create an account on (GitHub or Bitbucket)
- Initialize with a readme
- Clone the repo on Jarvis
- add a file
- write a commit message
- push to the repo
- view changes in browser

The Jarvis cluster

- Rocks 6.1 cluster (10 nodes)
- Linux CentOS 6.3
- /home mounted through NFS, accessible everywhere in the cluster (2TB)
- Each node:
  - 8-cores CPU
  - 16 GB RAM
  - 2TB HDD (local to each node at /state/partition1/*username*). Data here will be deleted once a week
  - Nvidia GPU
    - 4+1 Nvidia GTX 480
    - 3 Nvidia GTX 640
    - 2 Nvidia Tesla K20

The Jarvis cluster



Compilers available at jarvis

- C/C++/OpenMP
  - gcc 4.4.6
- Java
  - JRE 1.7.0_13
- MPI
  - OpenMPI
  - MPICH2 1.4.1p1
- perl
  - v 5.10.1
- python
  - v 2.6.6
- CUDA

SGE (How to submit jobs)

- Login node for programming, compiling and small programs only
- We will kill any process that takes too much time/too many resources (CPU, MEM,...)
- To run jobs, you need to use SGE
- Open source scheduler for clusters
- In SGE, you create job submission scripts with all the desired configurations
- The job is submitted to a specific queue, where it will run when the scheduler so decides

SGE (How to submit jobs)

Example (MPI)

- For more information please check “man qsub”. Read about SGE (http://gridscheduler.sourceforge.net/htmlman/manuals.html)

Simple MPI code (test.c)

```
#include <stdio.h>
#include <mpi.h>


int main (argc, argv)
     int argc;
     char *argv[];
{
  int rank, size;

  MPI_Init (&argc, &argv);    /* starts MPI */
  MPI_Comm_rank (MPI_COMM_WORLD, &rank);       /* get current process id */
  MPI_Comm_size (MPI_COMM_WORLD, &size);       /* get number of processes */
  printf( "Hello world from process %d of %d\n", rank, size );
  MPI_Finalize();
  return 0;
}
```

Compile: `$ mpicc test.c`

job submission script (testrun.sh)

```
#!/bin/bash

mpirun -npernode 4 /home/your/iit/username/a.out
```

Using only 4 process per node. Nodes in the cluster have 8 CPUs, but you can also run more than 8 processes in each node

Submit job

Optional: If not specified, scheduler will run on next available nodes from any queue

```
qsub -cwd -P cs550_f13_project -pe mpich 2  -hard -q nvidia480.q  testrun.sh
```

- cwd: Make it write output files to current working directory
- P: Project name (all students should use that project)
- pe: Parallel environment. In this case mpich for mpi
- 2: Number of nodes (remember each node has 8 CPUs)
- hard: If job can't run on specific queue, it won't run (alternative: `-soft`)
- q: queue. It is submitting to the nodes with nvidia GTX 480
- testrun.sh: Script to submit

Checking job in queue (jobid = 91)

```
$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
     91 0.00000 testrun.sh eberroca      qw     09/05/2013 02:05:24                                2
```

Output files

```
$ ls -lh | grep 91
-rw-r--r-- 1 eberroca nfsnobody      0 Sep  5 02:05 testrun.sh.e91
-rw-r--r-- 1 eberroca nfsnobody  256 Sep  5 02:05 testrun.sh.o91
-rw-r--r-- 1 eberroca nfsnobody      0 Sep  5 02:05 testrun.sh.pe91
-rw-r--r-- 1 eberroca nfsnobody  118 Sep  5 02:05 testrun.sh.po91
```

- e91: Standard error for job 91
- o91: Standard output for job 91
- pe91: Standard error for parallel environment in job 91
- po91: Standard output for parallel environment in job 91

Output

```
$ cat testrun.sh.o91
Hello world from process 0 of 8
Hello world from process 1 of 8
Hello world from process 2 of 8
Hello world from process 3 of 8
Hello world from process 6 of 8
Hello world from process 7 of 8
Hello world from process 5 of 8
Hello world from process 4 of 8
```

SGE (How to submit interactive jobs)

- Nodes are assigned to you when they are available

```
[test_user@jarvis ~]$ ssh gpu-compute-1-1
NO LOGIN PERMITTED FOR USERS! SUBMIT YOUR JOB USING THE SCHEDULER!
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[test_user@jarvis ~]$
```

```
[test_user@jarvis ~]$ qlogin -P cs550_f13_project -q nvidia480.q
Your job 97 ("QLOGIN") has been submitted
waiting for interactive job to be scheduled ...
Your interactive job 97 has been successfully scheduled.
Establishing builtin session to host gpu-compute-1-1.local ...
[test_user@gpu-compute-1-1 ~]$
```

SGE (How to submit interactive jobs)

- We will start allowing all users to log in indefinitely, but setting a limited number of *slots* per node (16).
- A GPU can only be used by one user at a time. If the general feeling is that users have a hard time running CUDA on the GPUs, we will pursue other strategies: Like, for example, less *slots* on each node or allowing just one user at a time but for a limited period of time (30 minutes per login).

SGE (How to submit interactive jobs)

You can request more than one node when running an interactive job

```
[test_user@jarvis ~]$ qlogin -P cs550_f13_project -pe mpich 2
```

The good news is that you don’t need to know which nodes (the names) you have requested when running MPI (the parallel environment is ready for you)

SGE (How to submit interactive jobs)

If you want to know which nodes you have in your interactive job , run the following:

```
[test_user@gpu-compute-1-0 ~]$ qstat -f
```

This command will output a lot of information regarding all the nodes. You can search for your username:

```
---------------------------------------------------------------------------------
admin.nvidia_480.q@gpu-compute BIP   0/0/1        0.00      linux-x64
---------------------------------------------------------------------------------
[...]
nvidia480.q@gpu-compute-1-0.lo BIP   0/1/1        0.00      linux-x64
     153 0.50500 QLOGIN  test_user r    09/05/2013 16:02:31      1
---------------------------------------------------------------------------------
nvidia480.q@gpu-compute-1-1.lo BIP   0/1/1        0.00      linux-x64
     153 0.50500 QLOGIN  test_user r    09/05/2013 16:02:31      1
[...]
```