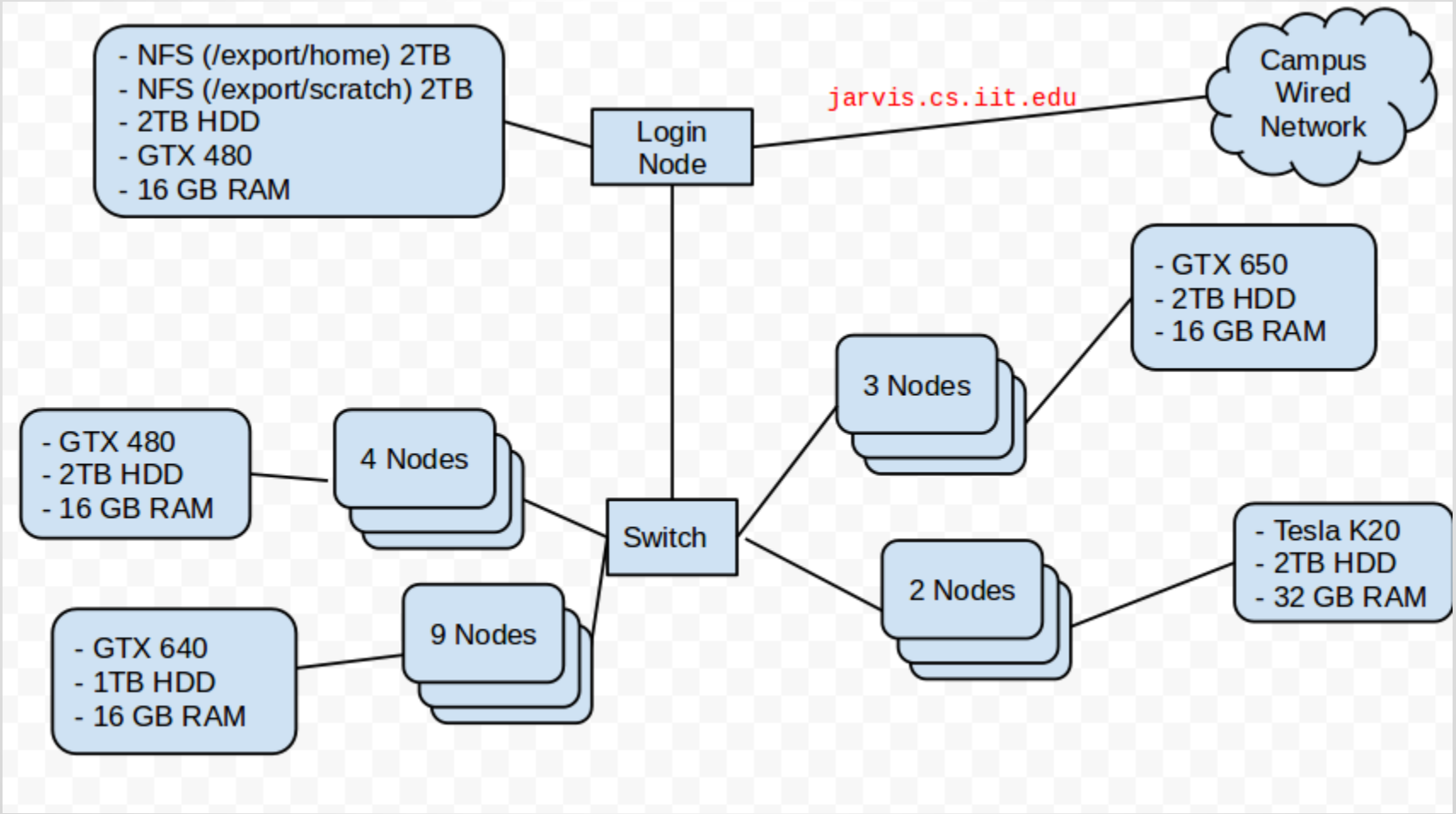
Intro to Jarvis

Scott Krieder & Eduardo Berrocal

Outline

- Intro to Jarvis
- How to request an account
- Using the cluster
  - login
  - submitting jobs
- Linux Basics
  - cs, ls, mkdir, touch, …
  - Makefile
  - emacs, vim
- Version Control

The Jarvis cluster



The Jarvis cluster

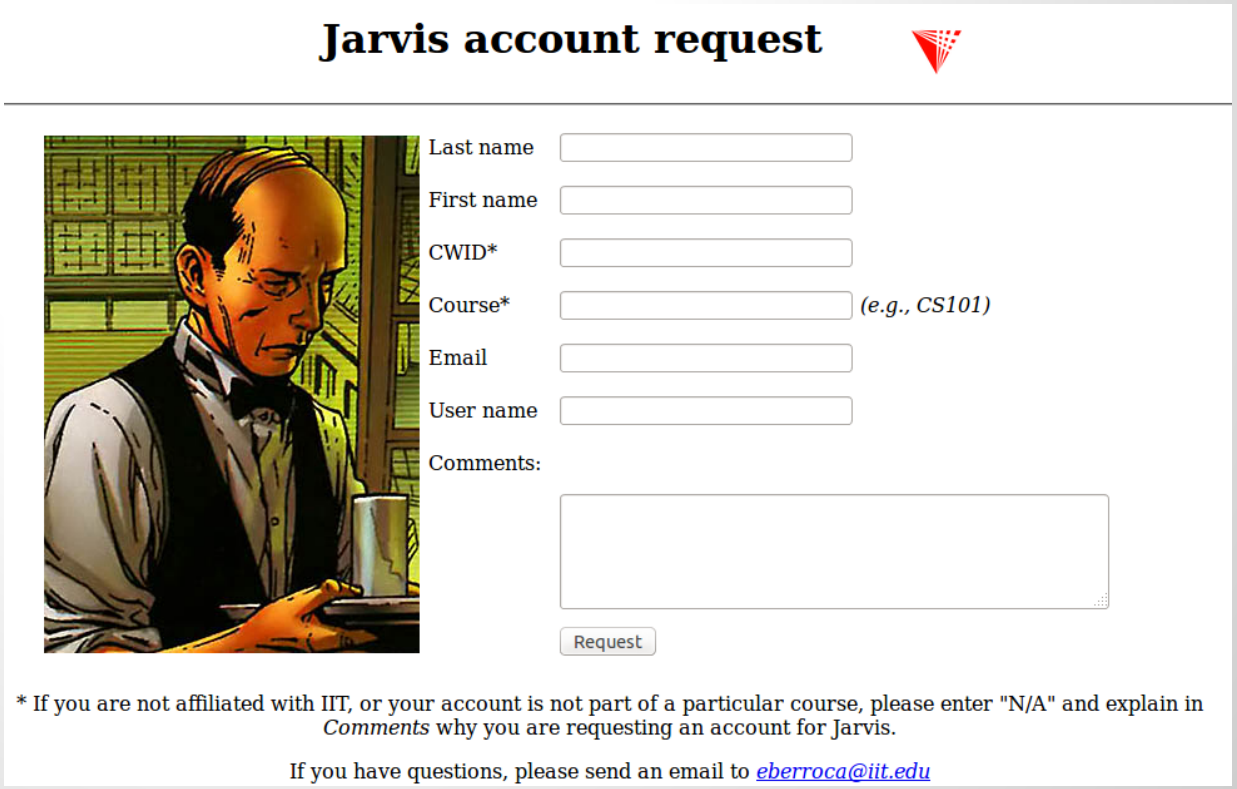
- Rocks 6.1 cluster (19 nodes)
- Linux CentOS 6.3
- ***home*** mounted through NFS, accessible everywhere in the cluster (2TB)
- Each node:
  - 8-cores CPU
  - 16/32 GB RAM
  - HOME: /export/home/*username* 2TB total
  - SCRATCH: /export/scratch/*username* 2TB total (deleted once a week) For now only accessible through login node
  - 2TB HDD (local to each node at /state/partition1).
  - Nvidia GPU
    - 4+1 Nvidia GTX 480
    - 3 Nvidia GTX 650
    - 9 Nvidia GTX 640
    - 2 Nvidia Tesla K20

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

Request an account

- https://bluesky.cs.iit.edu/jarvis
  - user: iit
  - password: iit2014



Login to jarvis cluster

- *ssh username@jarvis.cs.iit.edu*
- Linux & MacOS

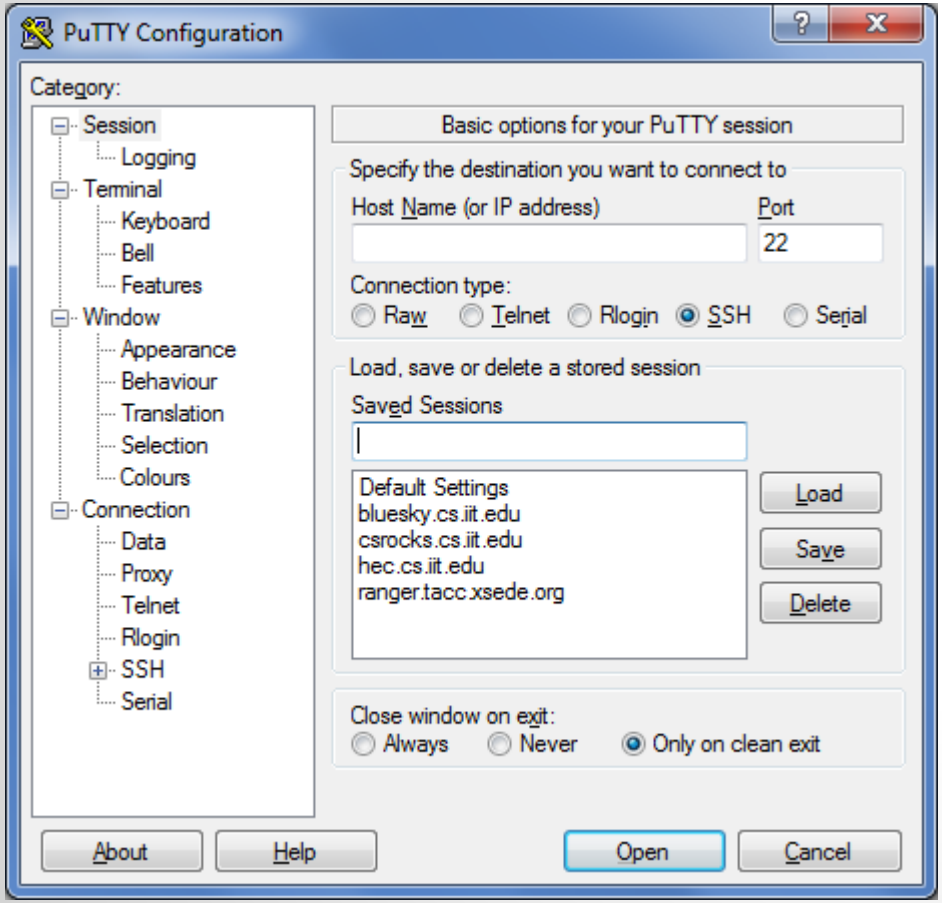
  Add the following to ~/.ssh/config:

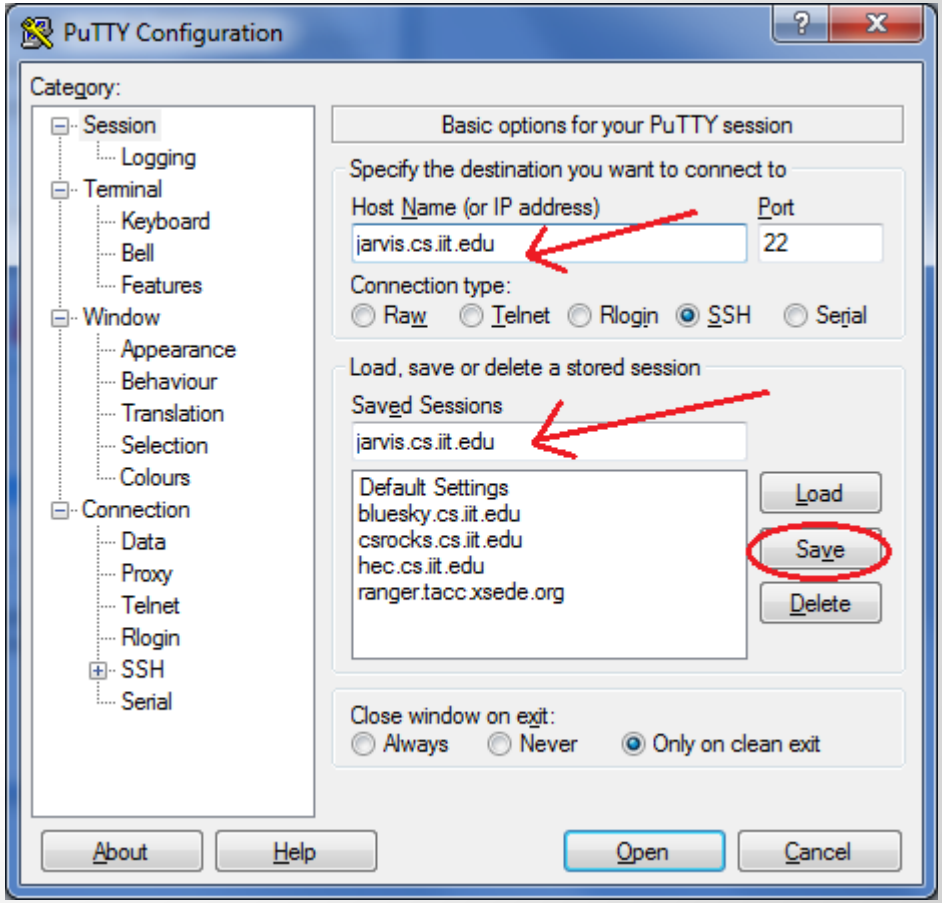
```
Host jarvis
    HostName jarvis.cs.iit.edu
    User username
    IdentityFile ~/.ssh/id_rsa
```

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

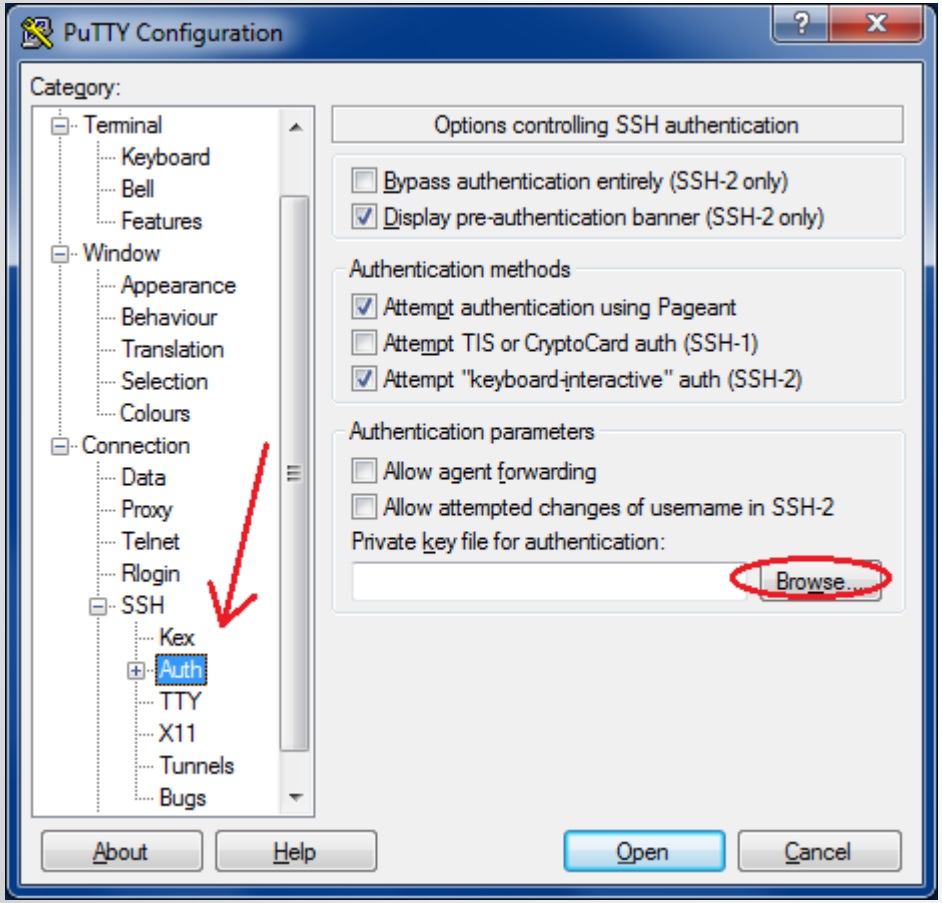
- Windows
  - PuTTY: http://www.putty.org/
  - WinSCP: http://winscp.net/eng/download.php

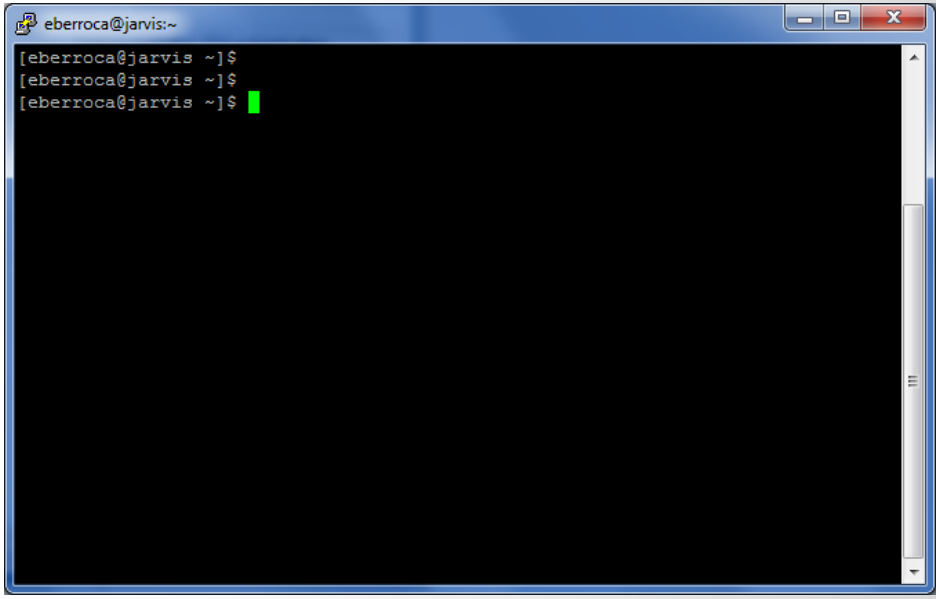
Login to jarvis (windows)



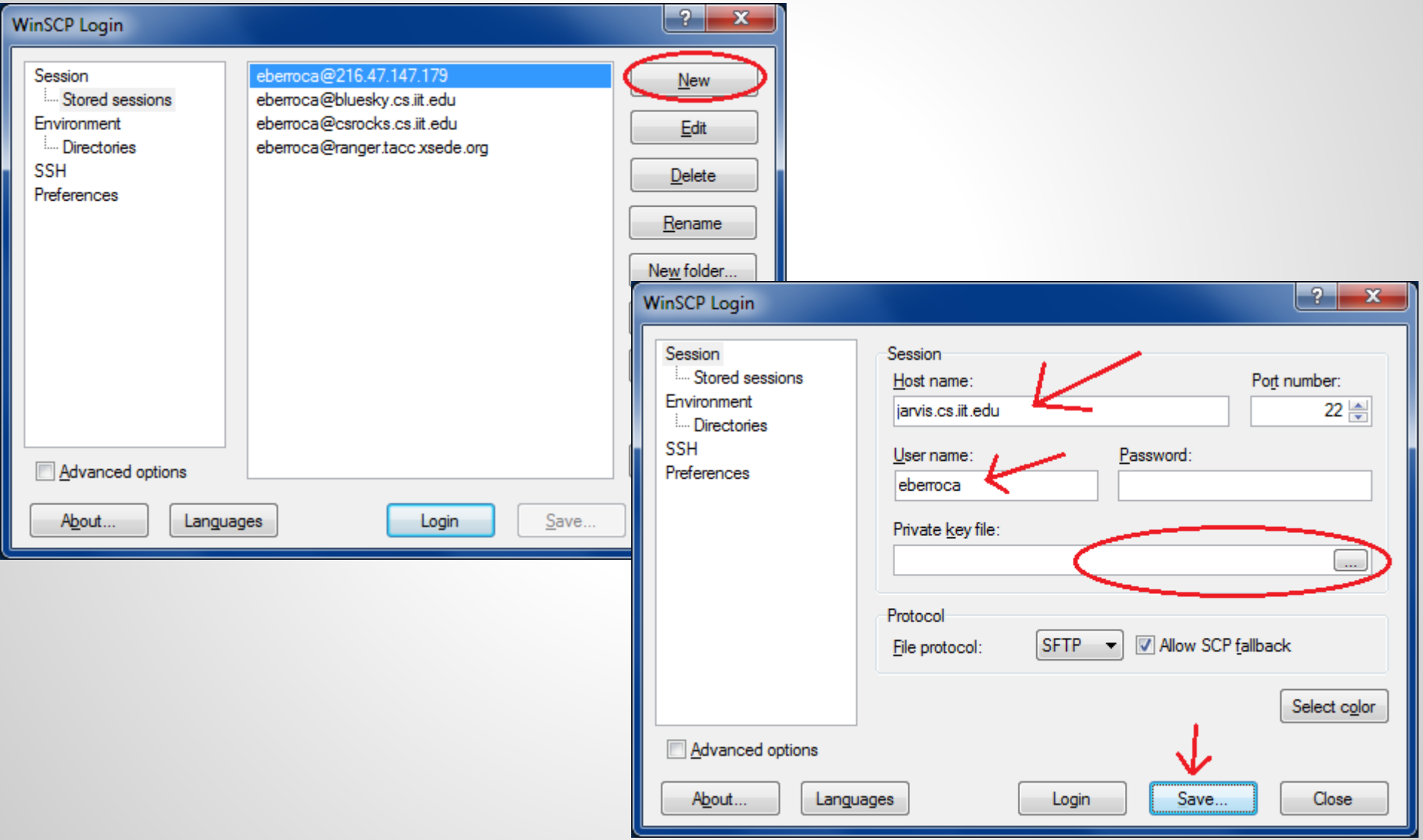


Login to jarvis (windows)

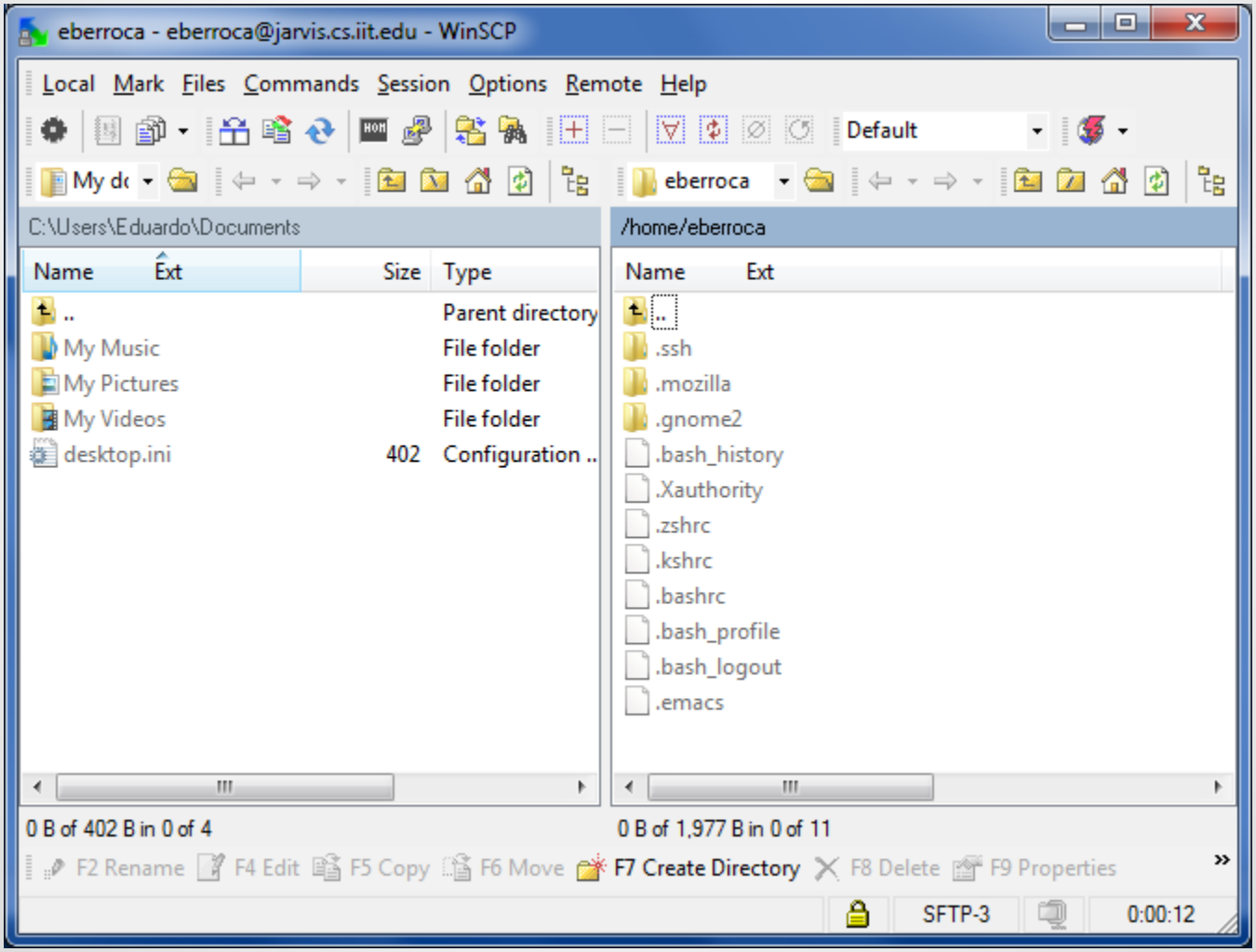




Copying files to jarvis (windows)



Copying files to jarvis (windows)



SGE (How to submit jobs)

- Login node for programming, compiling and small programs “hello world” 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/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

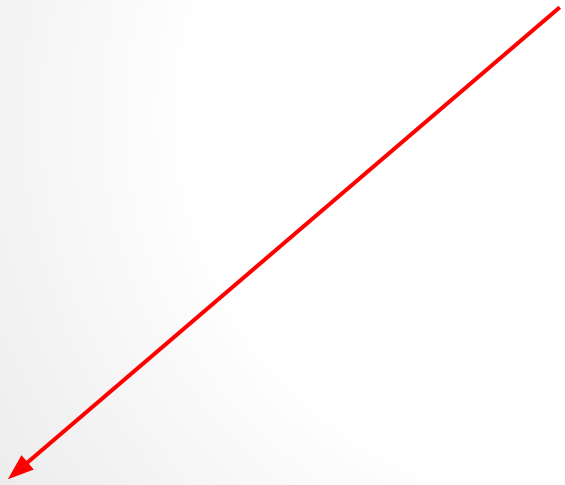
Submit job

```
$ qsub -pe mpich 2 testrun.sh
```

Parallel environment. This script will be run on two nodes using the MPI mpich implementation.

job submission script (testrun.sh)
nodes with a specific nvidia card

Running on nodes with nvidia GTX 650 cards only. The same way, you can specify 480.q, 640.q or k20.q



```
$ qsub -q 650.q -pe mpich 2 testrun.sh
```

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

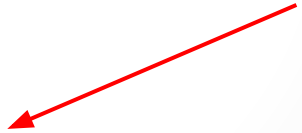
```
$ cat testrun.sh.o91
Hello world from process 0 of 8
Hello world from process 1 of 8
Hello world from process 2 of 8
Hello world from process 3 of 8
Hello world from process 6 of 8
Hello world from process 7 of 8
Hello world from process 5 of 8
Hello world from process 4 of 8
```

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 ~]$
```

You need to use the interactive queue.



```
[test_user@jarvis ~]$ qlogin -q interactive.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 have limited (although enough) slots. If you can’t login, wait for 5 or 10 minutes and try again.
- If you don’t use a login session anymore, please close the terminal (with *exit*).

SGE (How to submit interactive jobs)

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

```
[test_user@jarvis ~]$ qlogin -q interactive.q -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)
- ***Don’t use this option to interactively use more than one node. For that, open multiple terminals and do qlogin one node at a time***

SGE (How to submit interactive jobs)

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

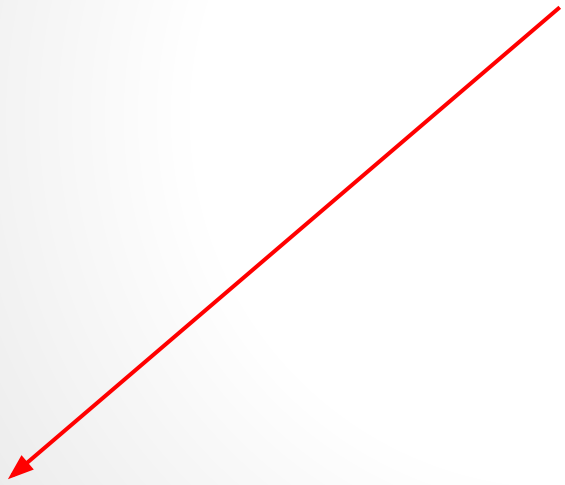
```
[test_user@gpu-compute-1-0 ~]$ qstat -f
```

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

```
---------------------------------------------------------------------------------
interactive.q@gpu-compute BIP   0/0/1        0.00      linux-x64
---------------------------------------------------------------------------------
[...]
interactive.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
---------------------------------------------------------------------------------
interactive.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
[...]
```

Interactive jobs
nodes with a specific nvidia card

Running on nodes with nvidia GTX 650 cards only. The same way, you can specify interactive.480.q, interactive.640.q or interactive.k20.q



```
$ qlogin -q interactive.650.q
```

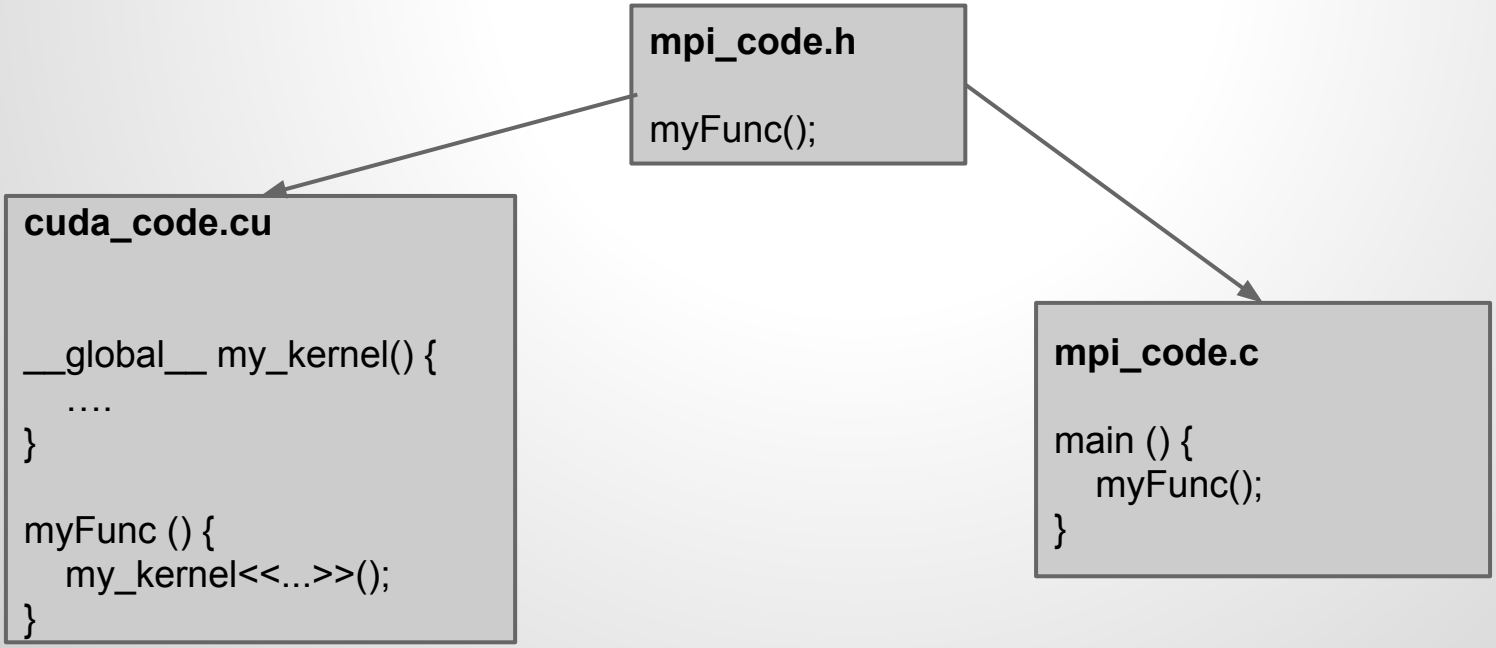
SGE (How to run MPI+CUDA)

- Compile CUDA code
- Include *.o objects, library paths, and include paths when compiling normal C MPI code
- Compile final MPI executable
- Submit to cluster as normal MPI code

SGE (How to run MPI+CUDA)

$ nvcc -c cuda_code.cu

$ mpicc -I/usr/local/cuda/include -L/usr/local/cuda/lib64 -lcudart -o *mpi_executable* mpi_code.c cuda_code.o



Linux Basics

- ls - list a directory
- cd - move directories
- emacs/vim - command line file editing
- mkdir - create a directory
- touch - touches a file
- Makefile - automating build process

Emacs basics

new file - emacs hello.c

save - ctrl + x, ctrl + s

close - ctrl + x, crtl + c

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

For all info check the wiki

https://sites.google.com/site/iitcuda/home