Southern Illinois University Carbondale

High–Performance Computing (**HPC**)

Prepared By:

Abdussamad Muntahi

Muhammad Rahman

Introduction to High–Performance Computing (**HPC**)

Introduction

- High-speed computing. Originally pertaining only to supercomputers for scientific research
- **Tools** and **systems** used to implement and create high performance computing systems
- Used for scientific research or computational science
- Main area of discipline is developing **parallel processing algorithms** and **software** so that programs can be divided into small parts and can be executed simultaneously by separate processors
- HPC systems have shifted from supercomputing to computing **clusters**

What is Cluster?

- Cluster is a group of machines interconnected in a way that they work together as a single system
- Used for better speed and capacity
- Types of Cluster
  - High-availability (HA) clusters
  - Load-balancing clusters
  - Grid computing
- Terminology
  - **Node** – individual machine in a cluster
  - **Head** node – connected to both the private network of the cluster and a public network and are used to access a given cluster. Responsible for providing user an environment to work and distributing task among other nodes
  - **Computer** nodes – connected to only the private network of the cluster and are generally used for running jobs assigned to them by the head node(s)

Benefits of Cluster

- **Reduced Cost**
  - The price of off-the-shelf consumer desktops has plummeted in recent years, and this drop in price has corresponded with a vast increase in their processing power and performance. The average desktop PC today is many times more powerful than the first mainframe computers.
- **Processing Power**
  - The parallel processing power of a high-performance cluster can, in many cases, prove more cost effective than a mainframe with similar power. This reduced price-per-unit of power enables enterprises to get a greater ROI (Return On Investment) from their IT budget.
- **Scalability**
  - Perhaps the greatest advantage of computer clusters is the scalability they offer. While mainframe computers have a fixed processing capacity, computer clusters can be easily expanded as requirements change by adding additional nodes to the network.

Benefits of Cluster

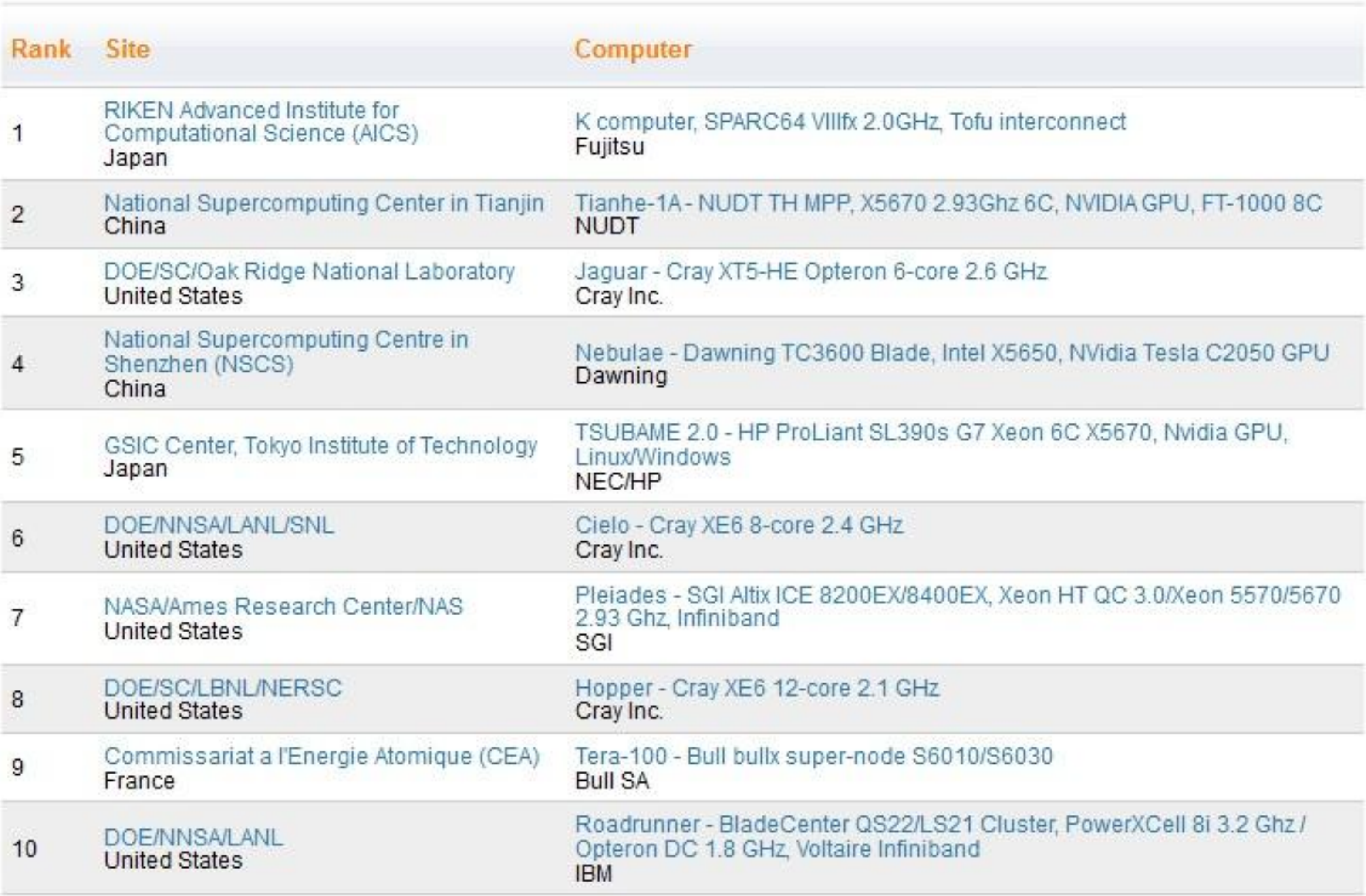
- **Improved Network Technology**
  - Driving the development of computer clusters has been a vast improvement in the technology related to networking, along with a reduction in the price of such technology.
  - In clusters, computers are typically connected via a single virtual local area network (VLAN), and the network treats each computer as a separate node. Information can be passed throughout these networks with very little lag, ensuring that data doesn't bottleneck between nodes.
- **Availability**
  - When a mainframe computer fails, the entire system fails. However, if a node in a computer cluster fails, its operations can be simply transferred to another node within the cluster, ensuring that there is no interruption in service.

Application of HPC

- Used to solve complex modeling problems in a spectrum of disciplines
- Topics include:
  - Artificial intelligence
  - Climate modeling
  - Cryptographic analysis
  - Geophysics
  - Molecular biology
  - Molecular dynamics
  - Nuclear physics
  - Physical oceanography
  - Plasma physics
  - Quantum physics
  - Quantum chemistry
  - Solid state physics
  - Structural dynamics.
- HPC is currently applied to **business** uses as well
  - data warehouses
  - line-of-business (LOB) applications
  - transaction processing

Top 10 Supercomputers for HPC

June 2011



Rank	Site	Computer
1	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu
2	National Supercomputing Center in Tianjin China	Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C NUDT
3	DOE/SC/Oak Ridge National Laboratory United States	Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz Cray Inc.
4	National Supercomputing Centre in Shenzhen (NSCS) China	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU Dawning
5	GSIC Center, Tokyo Institute of Technology Japan	TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows NEC/HP
6	DOE/NNSA/LANL/SNL United States	Cielo - Cray XE6 8-core 2.4 GHz Cray Inc.
7	NASA/Ames Research Center/NAS United States	Pleiades - SGI Altix ICE 8200EX/8400EX, Xeon HT QC 3.0/Xeon 5570/5670 2.93 Ghz, Infiniband SGI
8	DOE/SC/LBNL/NERSC United States	Hopper - Cray XE6 12-core 2.1 GHz Cray Inc.
9	Commissariat a l'Energie Atomique (CEA) France	Tera-100 - Bull bullx super-node S6010/S6030 Bull SA
10	DOE/NNSA/LANL United States	Roadrunner - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband IBM

Typical Cluster

Fastest Supercomputer in USA:

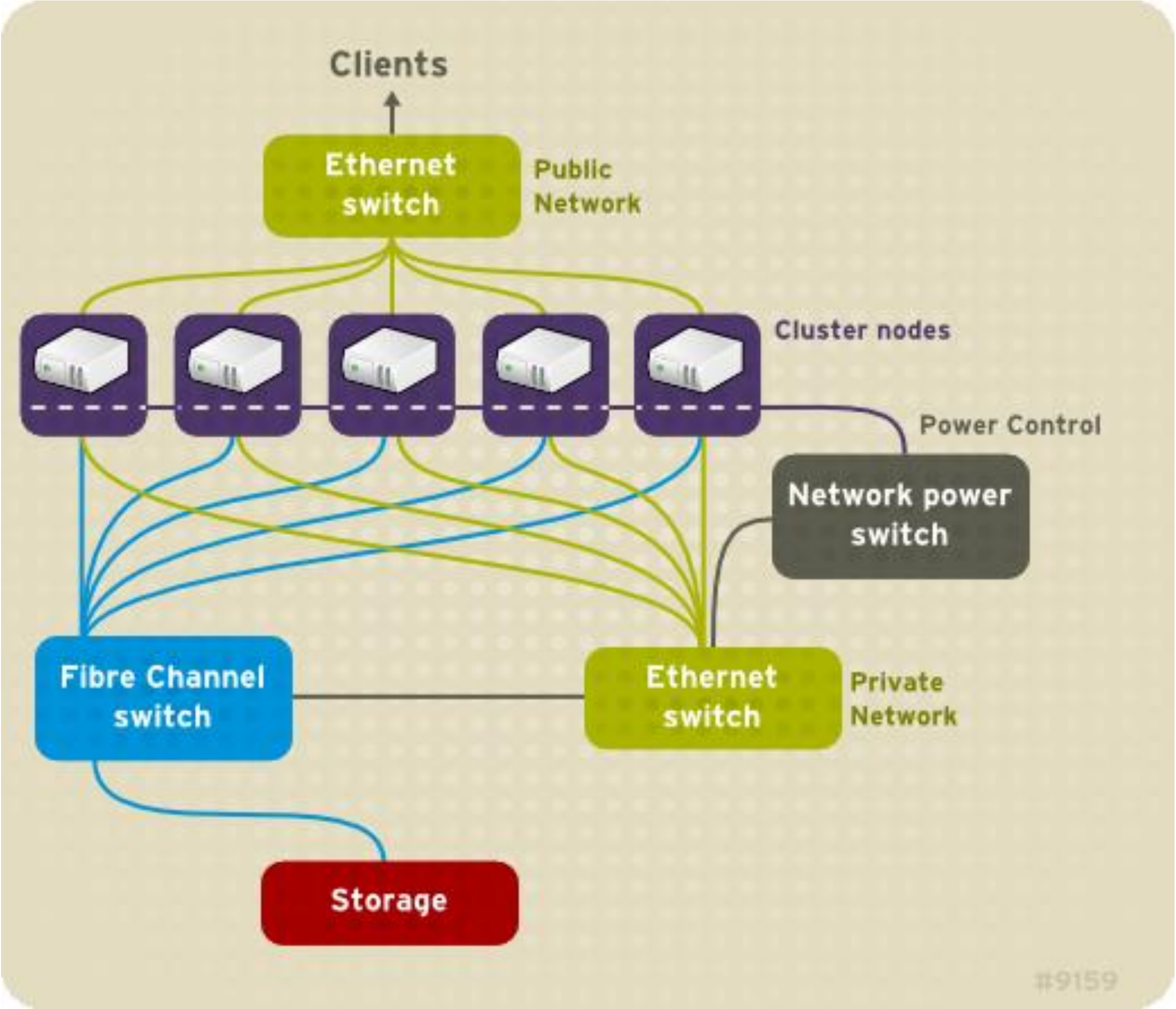
Jaguar @ Oak Ridge National Lab

http://computing.ornl.gov

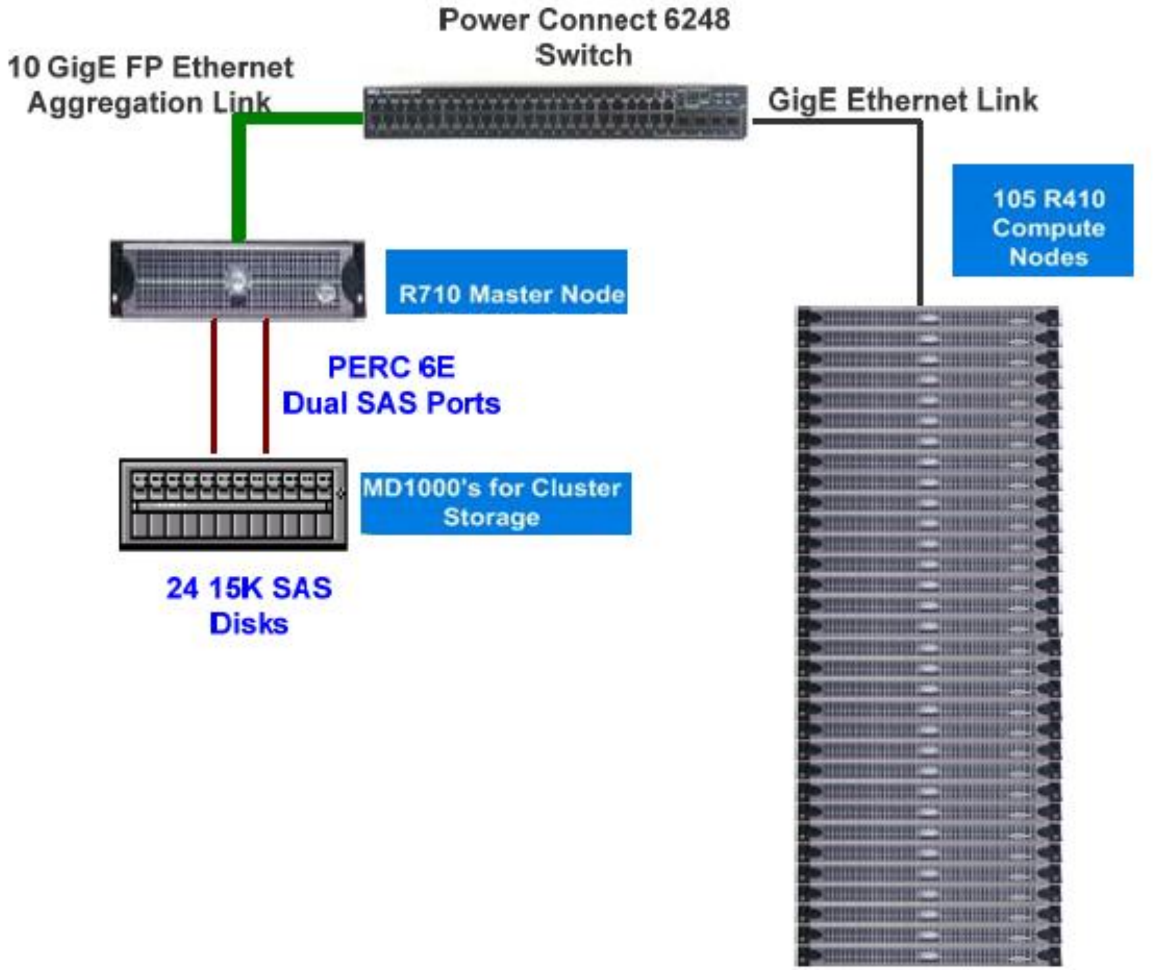
Hardware and Our Cluster

SIHPCI (maxwell)

Cluster Architecture: Typical



Cluster Architecture: Maxwell



The Cluster: maxwell

Tech Specs:

- No. of nodes: 106
- Each node is a Intel dual CPU Quad Core 2.3 GHz Processor
- Total No. of cores: 848
- RAM per node: 8 GB
- Storage Memory: 90 TB

Hardware: Master/Head Node

- Head node is responsible for providing user an environment to work and distributing task among other nodes
- Minimum Specification
  - CPU of i586 or above
  - A network interface card that supports a TCP/IP stack
  - At least 4GB total free space – 2GB under and 2GB under /var
  - A Floppy Drive
  - A CD-Rom Drive

Front End

Back End

Hardware: Master/Head Node

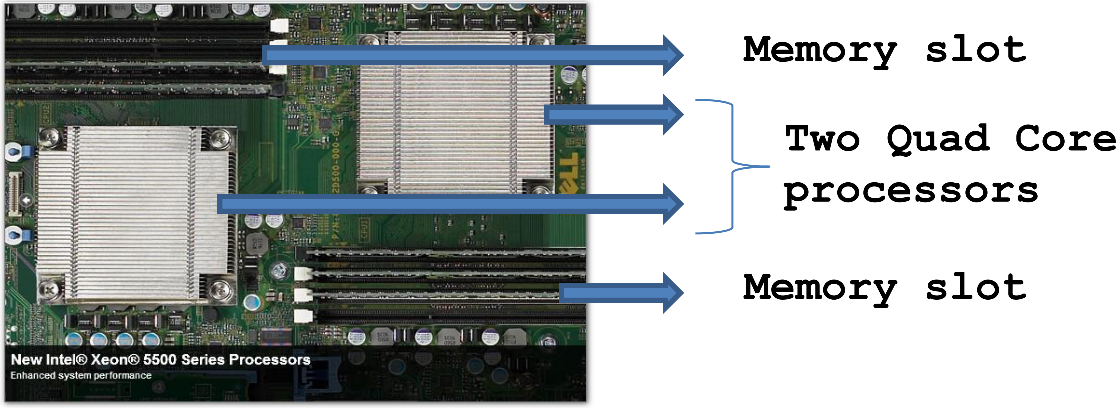
- Maxwell Specification

Front End

Server format	Rack
CPU family	Intel Xeon
CPU nominal frequency	2.26GHz
Processor Model	Xeon E5520
Processors supplied	2 Quad core
Memory RAM capacity	24GB Memory (6x4GB),
Memory type	DDR3
Memory frequency	1066MHz Quad Ranked RDIMMs
Storage HDD	146GB 15K RPM Serial-Attach SCSI
RAID module	PERC 6/i SAS RAID Controller 2x4 Connectors
Gigabit LAN	ports 2
Power supply rating	480W
Idle power consumption	150W
Peak power consumption	270W
OS	Red Hat Enterprise Linux 53AP x32 And x64

Hardware: Computing Node (Client)

- Dedicated for Computation
- Minimum Specification
  - CPU of i586 or above
  - A disk on each client node, at least 2GB in size
  - A network interface card that supports a TCP/IP stack
  - All clients must have the same architecture (e.g., ia32 vs. ia64)
  - Monitors and keyboards may be helpful, but are not required
  - Floppy or PXE enable BIOS
  - A CD-Rom Drive


New Intel® Xeon® 5500 Series Processors
Enhanced system performance

Memory slot

Two Quad Core processors

Memory slot

Hardware: Computing Node (Client)

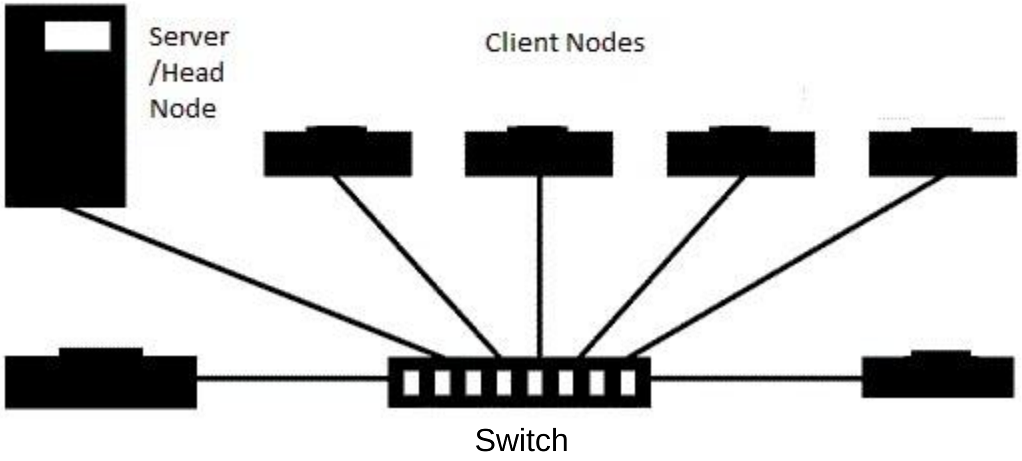
- Maxwell Specification

CPU family	Intel Xeon
CPU nominal frequency	2.13GHz
Processors supplied	2 quad core
Memory RAM capacity	8GB Memory (4x2GB)
Memory type	DDR3
Memory frequency	1333MHz Dual Ranked UDIMMs
storage HDD	160GB 7.2K RPM SATA
Gigabit LAN	ports 2
Power supply rating	480W
Idle power consumption	115W
Peak power consumption	188W
OS	Red Hat Linux 5 HPC

Front End

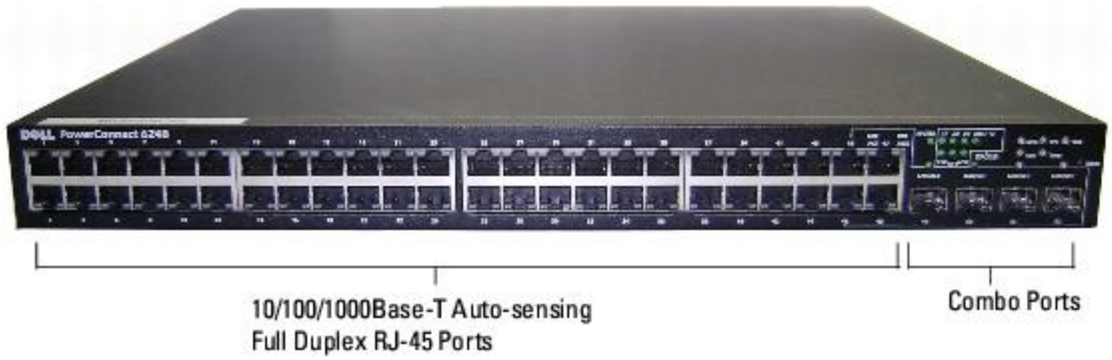
Hardware: Switch

- Minimum Specification
  - The switch is necessary for communication between the nodes
  - Each node (including the head node) should have its own port on the switch. In other words, if there are one head node and 8 client nodes, you need at a minimum a 9-port switch



Hardware: Switch

- Maxwell Specification
  - Model: Power Connect 6248
  - Port: 48 10/100/1000BASE-T auto-sensing Gigabit Ethernet switching ports
  - 48 GbE(giga bit ethernet) Port Managed Switch, two 10GbE and Stacking Capable



Power Connect 6248

Switch Stack

Hardware: Power Distribution Unit

- APC Switched Rack Power Distribution Units (PDUs) place rack equipment power control in the hands of the IT Manager. Remote outlet level controls allow power on/off functionality for power recycling to reboot locked-up equipment and to avoid unauthorized use of individual outlets. Power sequencing delays allow users to define the order in which to power up or down attached equipment. Avoid circuit overload during power recovery and extend uptime of critical equipment by prioritizing the load shedding.

  - PDU plug type L6-30P
  - PDU Model APC AP7541
  - PDU Max Amperage Load 30

Hardware: External Storage Array

- Minimum Specification:

Model Power Vault	MD1000 Hard Drive
Max Supported Capacity	1.1 TB
Host Channels	2
Data Transfer Rate	300 MBPs
Supported Devices Hard drive ,	Disk array (RAID)
Spindle Speed	15000 RPM

Front Side

- Maxwell Specification

Total storage array	6
In Each Storage Array	15 HDD
Each HDD has	1*1 TB
Total Storage Capacity	6*15*1.1 TB

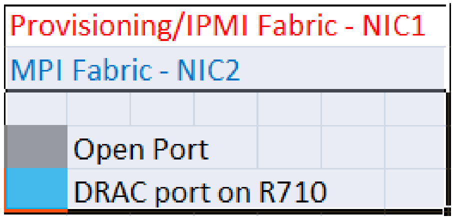
Back Side

Hardware: KVM Switch

- KVM (Keyboard Video Mouse) Switch is a device used to connect a keyboard, mouse and monitor to two or more computers. KVM switches save money, time, space, equipment and power. These switches are also widely deployed to control pools of servers in data centers. Some KVM switches support user terminals at both ends that allow local and remote access to all the computers or servers.

Hardware: Networking

- Clusters are interconnected with both GigE (Dell PowerConnect 6248 and 48 GbE PortManaged
- Switch, 2xDell PowerConnect 3424 24 Port FE with 2 GbE Copper Ports and 2 GbE Fiber SFP
- Ports and Infiniband (Dell 24-Port Internally Managed 9024 DDR InfiniBand Edge Switch)
- Switches and cards

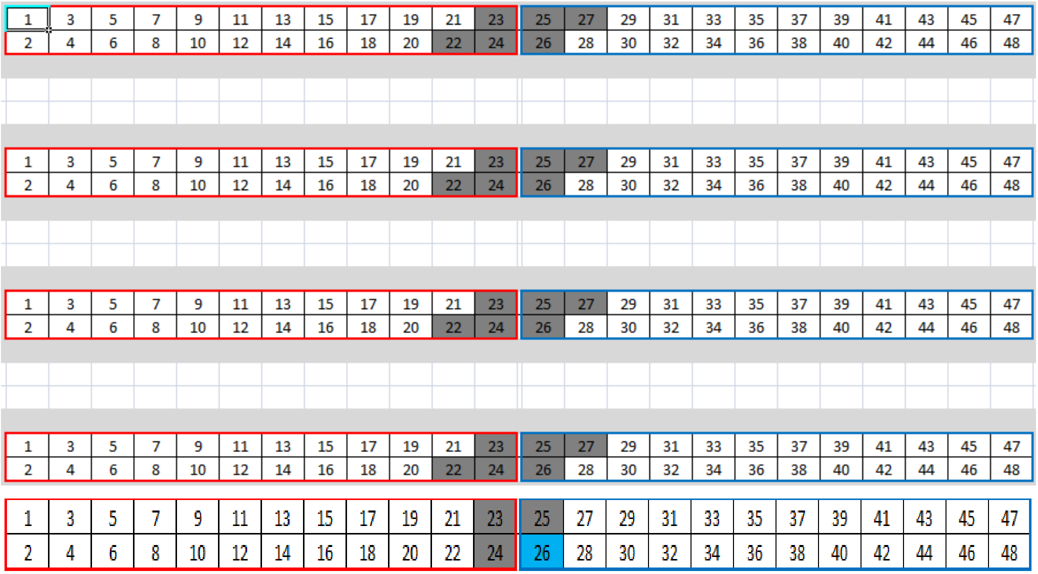


Provisioning/IPMI Fabric - NIC1

MPI Fabric - NIC2

▨ Open Port

🟦 DRAC port on R710



1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48

1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48

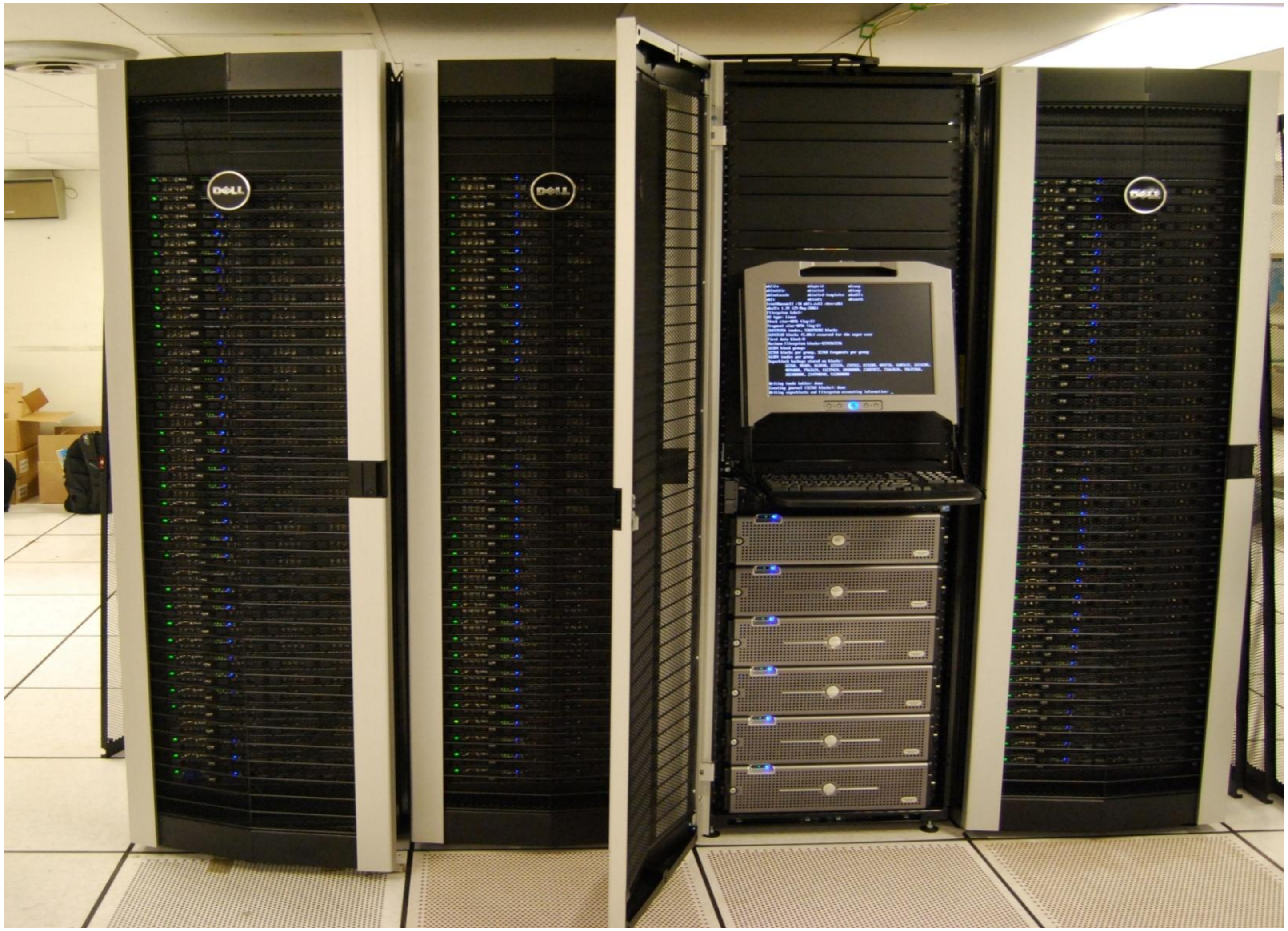
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48

1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48

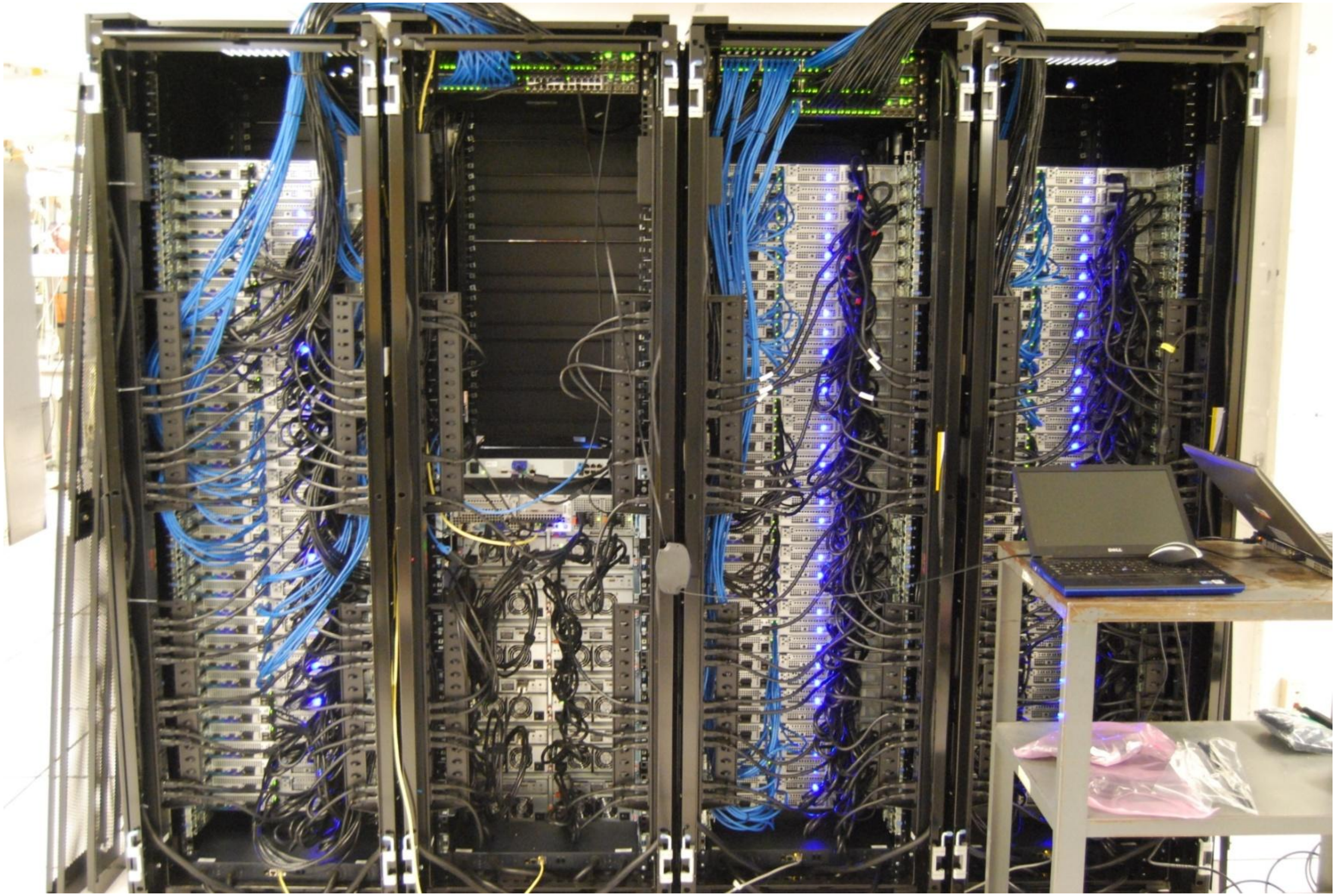
1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47
2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48

Switch Ports Cable connectivity

Maxwell (Front)



Maxwell (Back)



Software for High–Performance Computing (**HPC**)

Software for HPC

- For effective use of cluster for HPC the following tools are at our disposal
  - Remote hardware management
    - Remote power on/off
    - Monitoring CPU (for temperature etc.)
  - Cluster management
    - Monitoring programs, system administration etc.
  - Job scheduling
  - Libraries/languages for parallel programming
    - Massage Passing Interface (**MPI**)

Cluster Management

- Cluster management software offers
  - Easy-to-use interface for managing clusters
  - Automates the process of queuing jobs
  - Matching the requirements of a job and the resources available to the cluster
  - Migrating jobs across the cluster
- Maxwell uses *Red Hat Enterprise Linux*

Cluster Management

- *Red Hat Enterprise Linux*
  - – Specially for the scientific computing purpose to deploy clusters of systems that work together
  - – Excellent hardware detection and monitoring capabilities
  - – Centralized authentication and logging services
  - – Fast IO (Input/Output)

Parallel Computing

- Form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently i.e. "in parallel"
- Different forms of parallel computing
  - Bit-level parallelism
  - Instruction level parallelism
  - Data parallelism
  - Task parallelism
- Parallel Computer classification
  - Multiple processing elements (multi-core and multi-processor) within a single machine
  - Using multiple computers to work on the same task - clusters, MPPs (Massive Parallel Processing), and grids

Parallel Programming

- Parallel computer programs are more difficult to write than sequential programs
- Potential problems
  - Race condition (output depending on sequence or timing of other events)
  - Communication and synchronization between the different subtasks
- HPC Parallel Programming Models associated with different computing technology
  - Single Instruction Multiple Data (SIMD) on Single Processors
  - Multi-Process and Multi-Threading on SMP (symmetric multiprocessing) Computers
  - Message Passing Interface (MPI) on Clusters

Parallel Programming

- Message Passing Interface (**MPI**)
  - An application programming interface (API) specification that allows processes to communicate with one another by sending and receiving messages
  - Now a *de facto* standard for parallel programs running on distributed memory systems in computer clusters and supercomputers
  - A massage passing API with language-independent protocol and semantic specifications
  - Support both point-to-point and collective communication
  - Goals are high performance, scalability, and portability
  - Consists of a specific set of routines (i.e. APIs) directly callable from *C*, *C++*, *Fortran* and any language able to interface with such libraries, including *C#*, *Java* or *Python*

Tutorial on *Maxwell*

Maxwell: A Brief Introduction

Tech Specs:

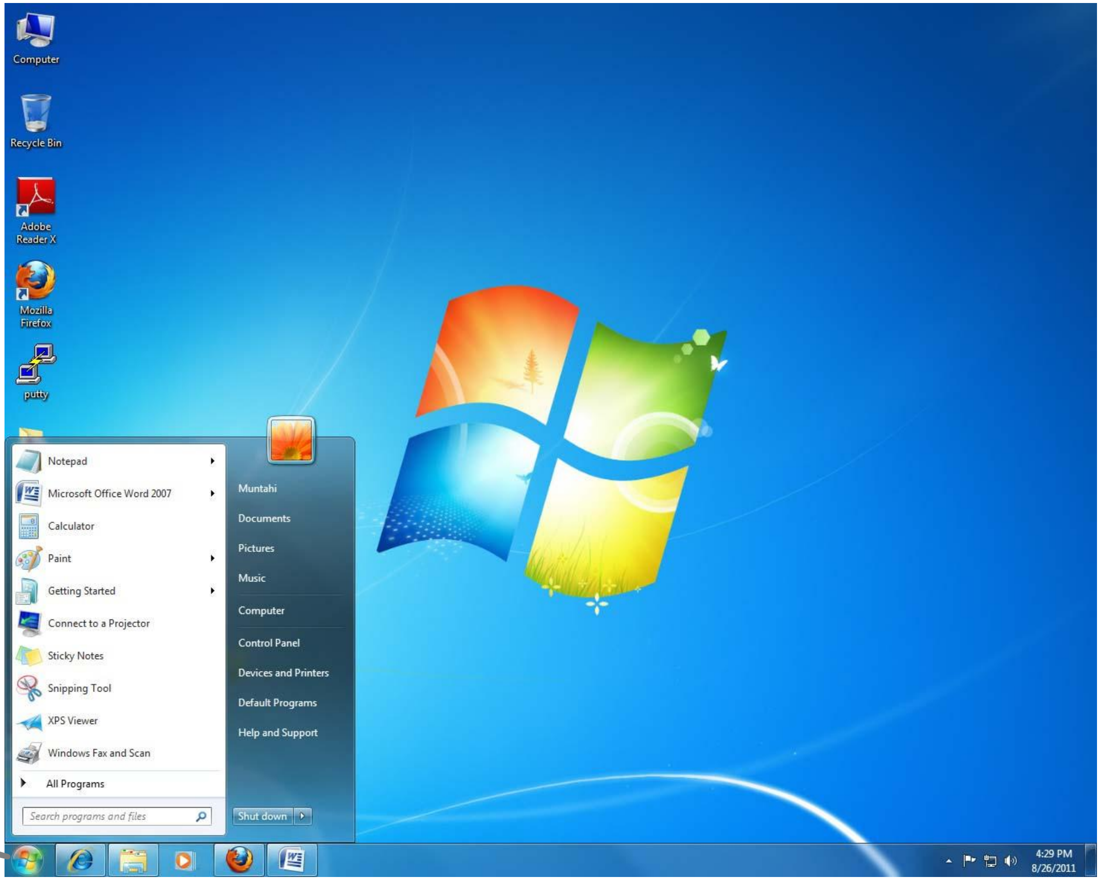
- No. of nodes: 106
- Each node is a Intel dual CPU Quad Core 2.3 GHz Processor
- Total No. of cores: 848
- RAM per node: 8 GB
- Storage Memory: 90 TB

How to create an account?

- Send an email to
  - Nancy Beasley nancyj0@siu.edu or
  - Dr. Shaikh Ahmed ahmed@siu.edu
- Provide the following information
  - Name
  - Affiliation
  - ***IP address*** of the computer(s) on SIUC network from which you would access Maxwell
- Will receive an email with Log In information

How to create an account?

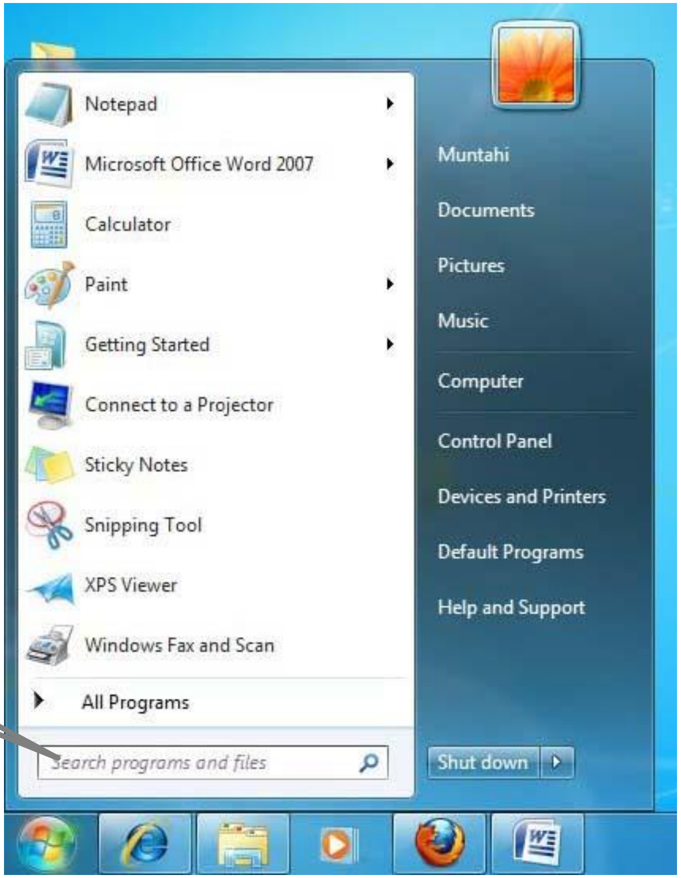
- IP address look-up



Go to Start

How to create an account?

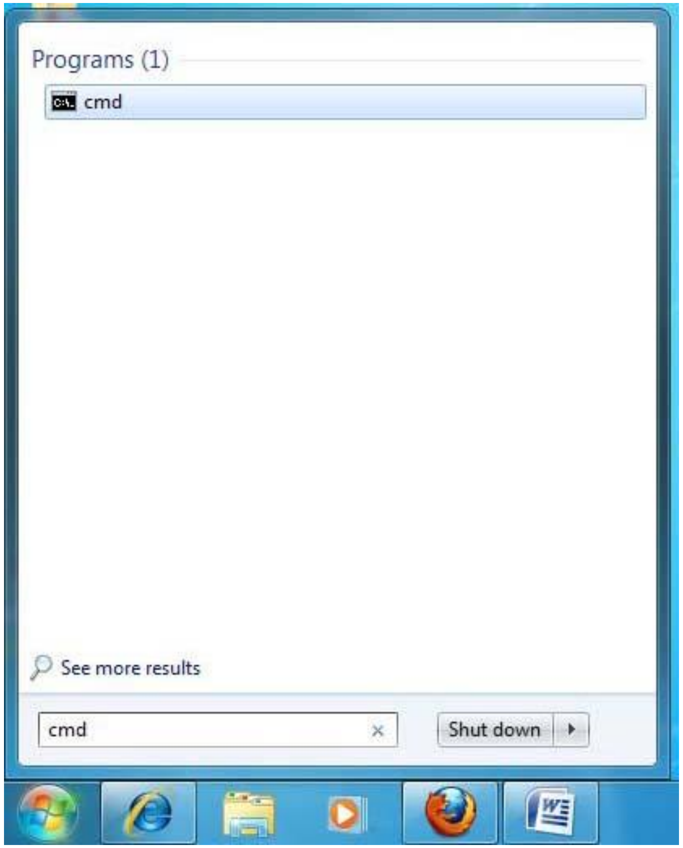
- IP address look-up



Type in 'cmd' to go to command prompt

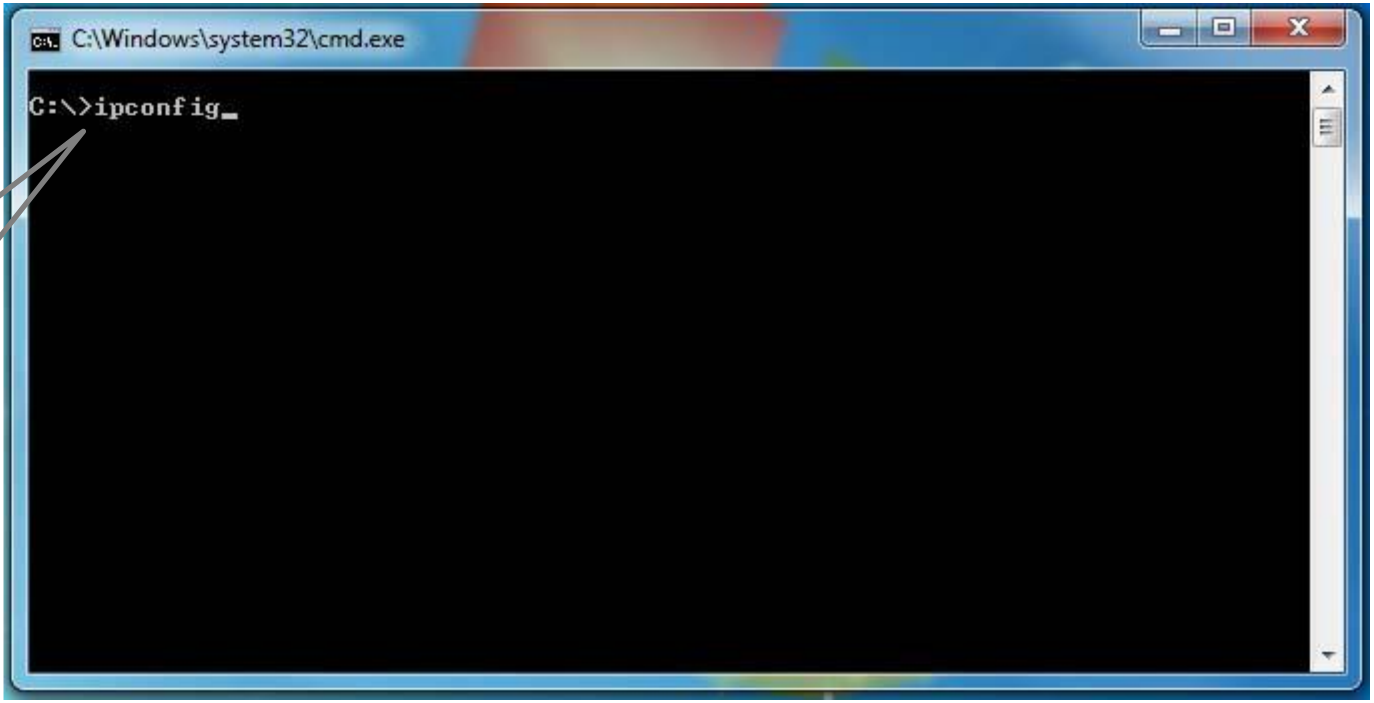
How to create an account?

- IP address look-up



How to create an account?

- IP address look-up

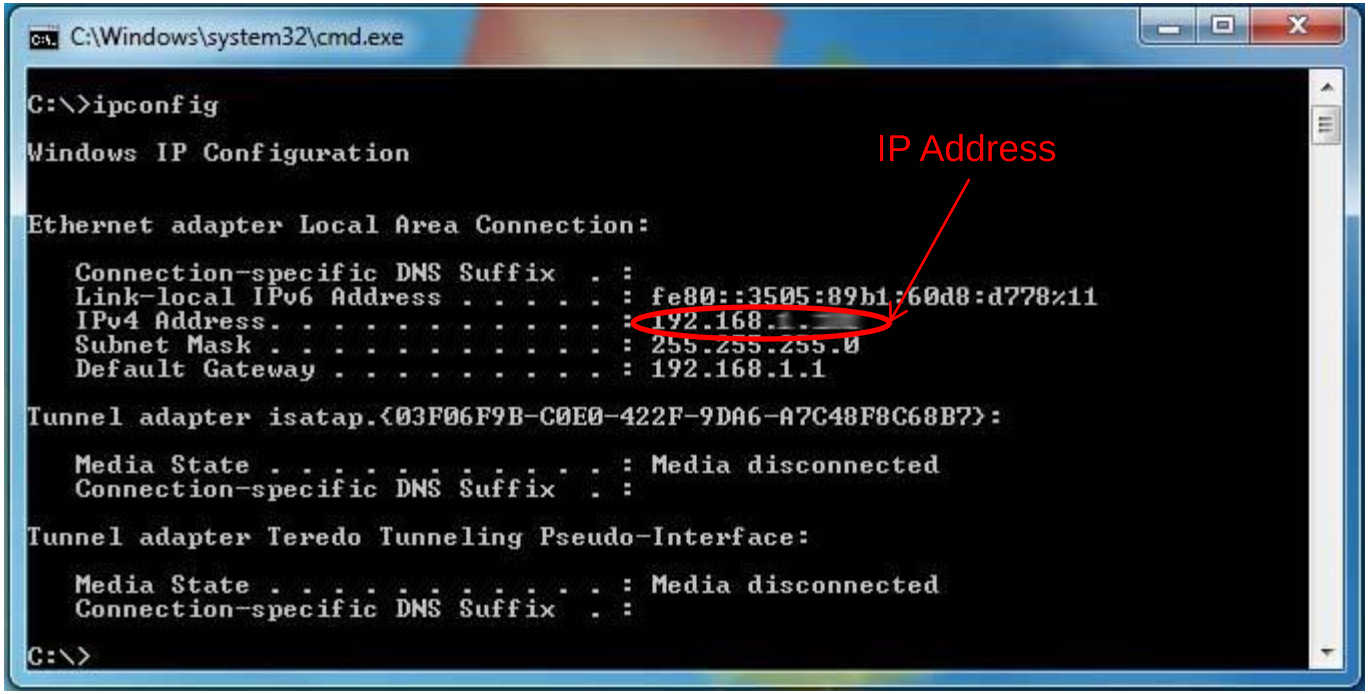


```
C:\>ipconfig_
```

Type in ‘ipconfig’ to get the IP Address

How to create an account?

- IP address look-up



```
C:\>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::3505:89b1:60d8:d778%11
   IPv4 Address. . . . . . . . . . . : 192.168.[redacted]
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter isatap.{03F06F9B-C0E0-422F-9DA6-A7C48F8C68B7}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Teredo Tunneling Pseudo-Interface:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\>
```

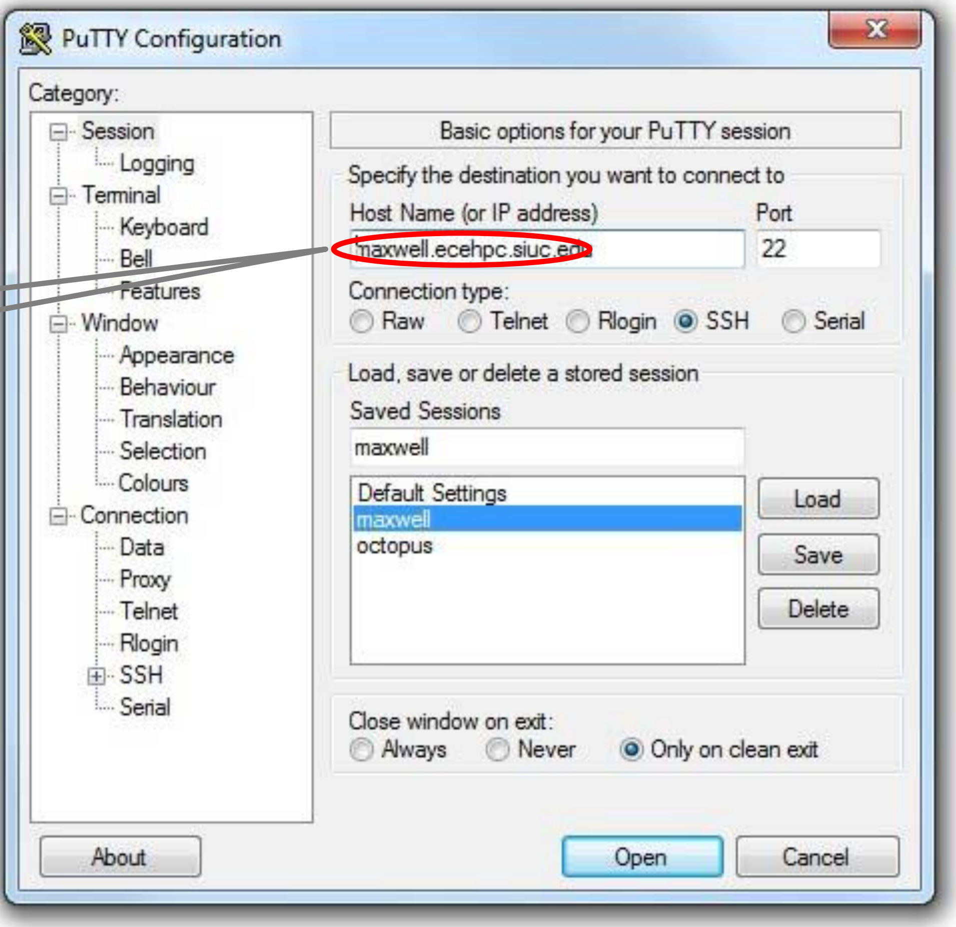
Login Procedure

- Download ‘Putty’
  - Web addresses
    - http://www.putty.org/
    - http://download.cnet.com/PuTTY/3000-7240_4-10808581.html
  - Run ‘Putty’
    - Use Host Name or IP address of Maxwell
      - Host Name: maxwell.ecehpc.siuc.edu
    - Enable X11

Login Procedure

- Run ‘Putty’
  - Start ‘Session’

Type in Host Name: “maxwell.ecehpc.siuc.edu”



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): maxwell.ecehpc.siuc.edu

Port: 22

Connection type:
○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: maxwell

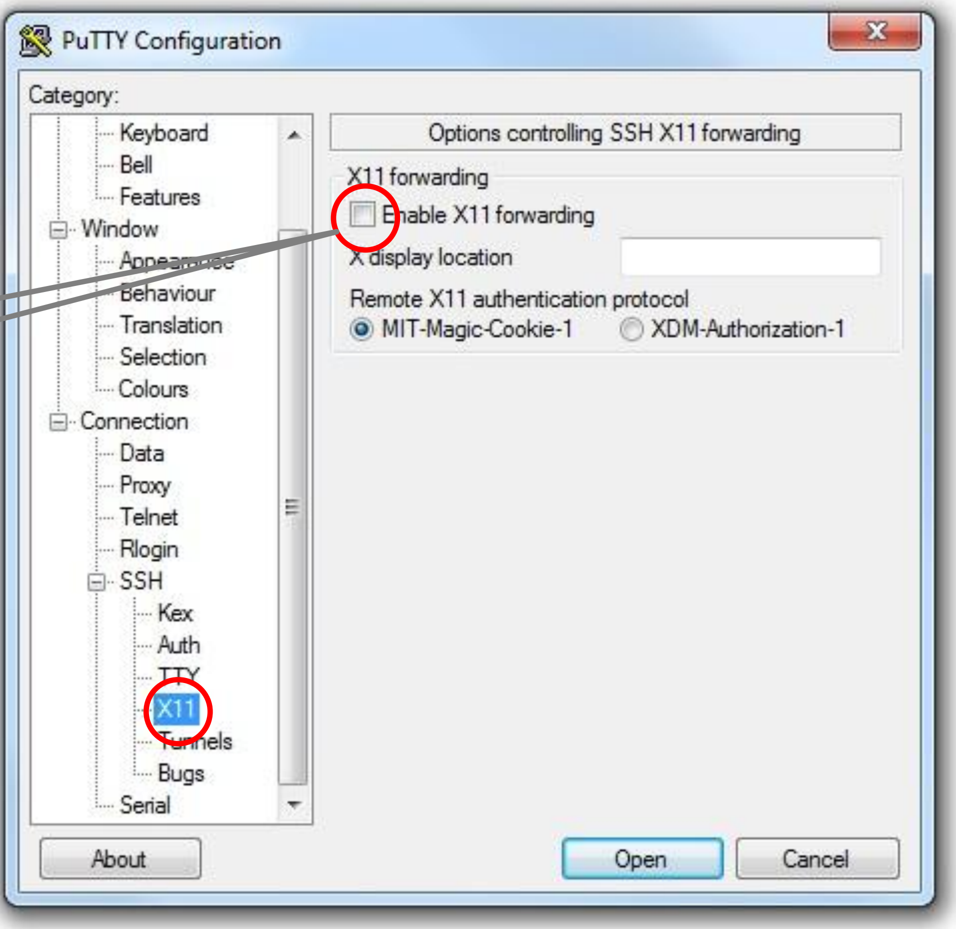
Default Settings
maxwell
octopus

Close window on exit:
○ Always ○ Never ● Only on clean exit

Login Procedure

- Run ‘Putty’
  - Start ‘Session’
  - Enable X11
    - Connection > SSH > X11

Check ‘Enable X11 Forwarding’



PuTTY Configuration

Category:

Options controlling SSH X11 forwarding

X11 forwarding

□ Enable X11 forwarding

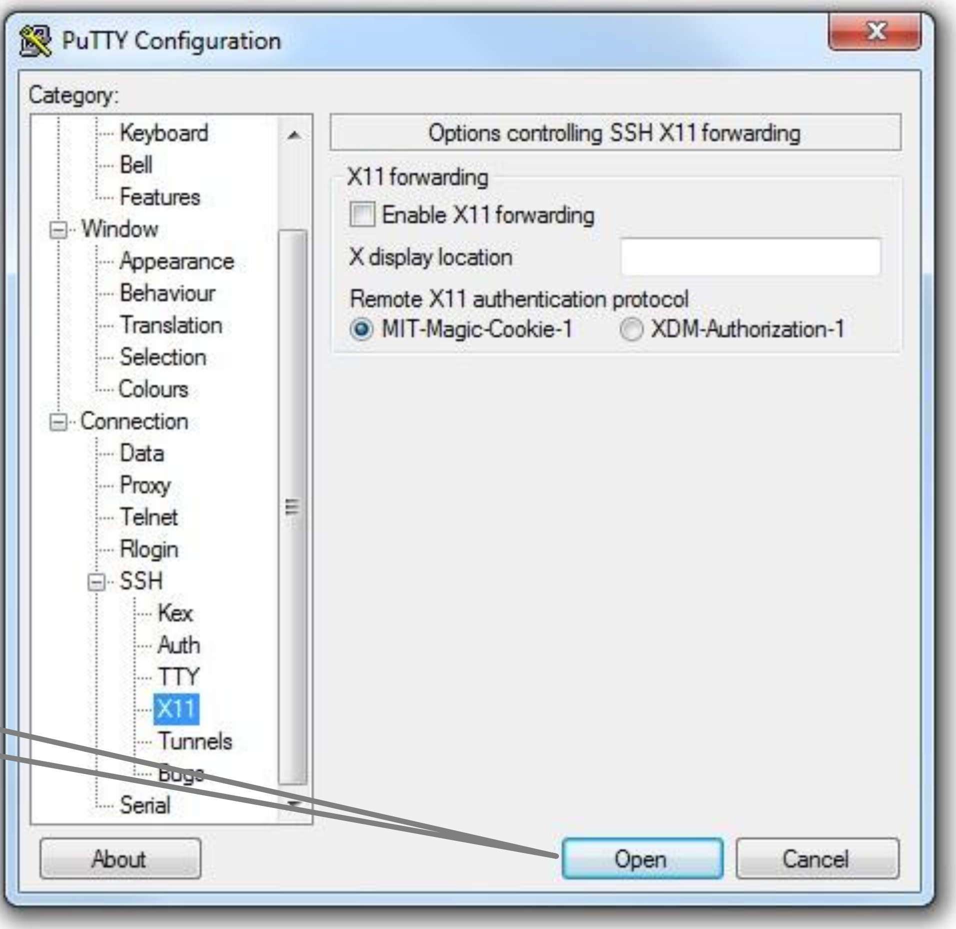
X display location

Remote X11 authentication protocol

● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

Login Procedure

- Run ‘Putty’
  - Start ‘Session’
  - Enable X11
    - Connection > SSH > X11
  - Open



Options controlling SSH X11 forwarding

X11 forwarding

□ Enable X11 forwarding

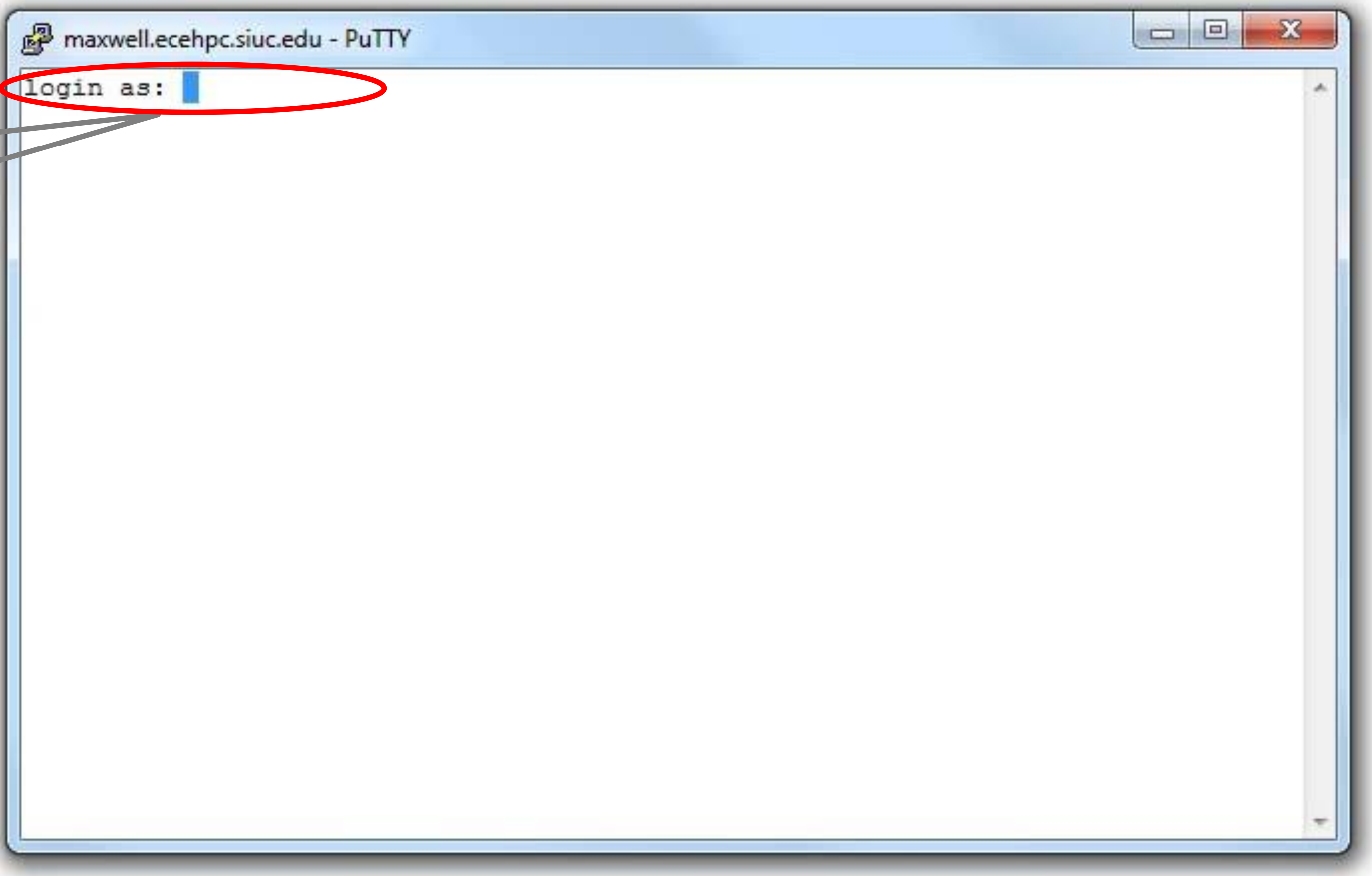
X display location

Remote X11 authentication protocol

● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

Press Open to start the session

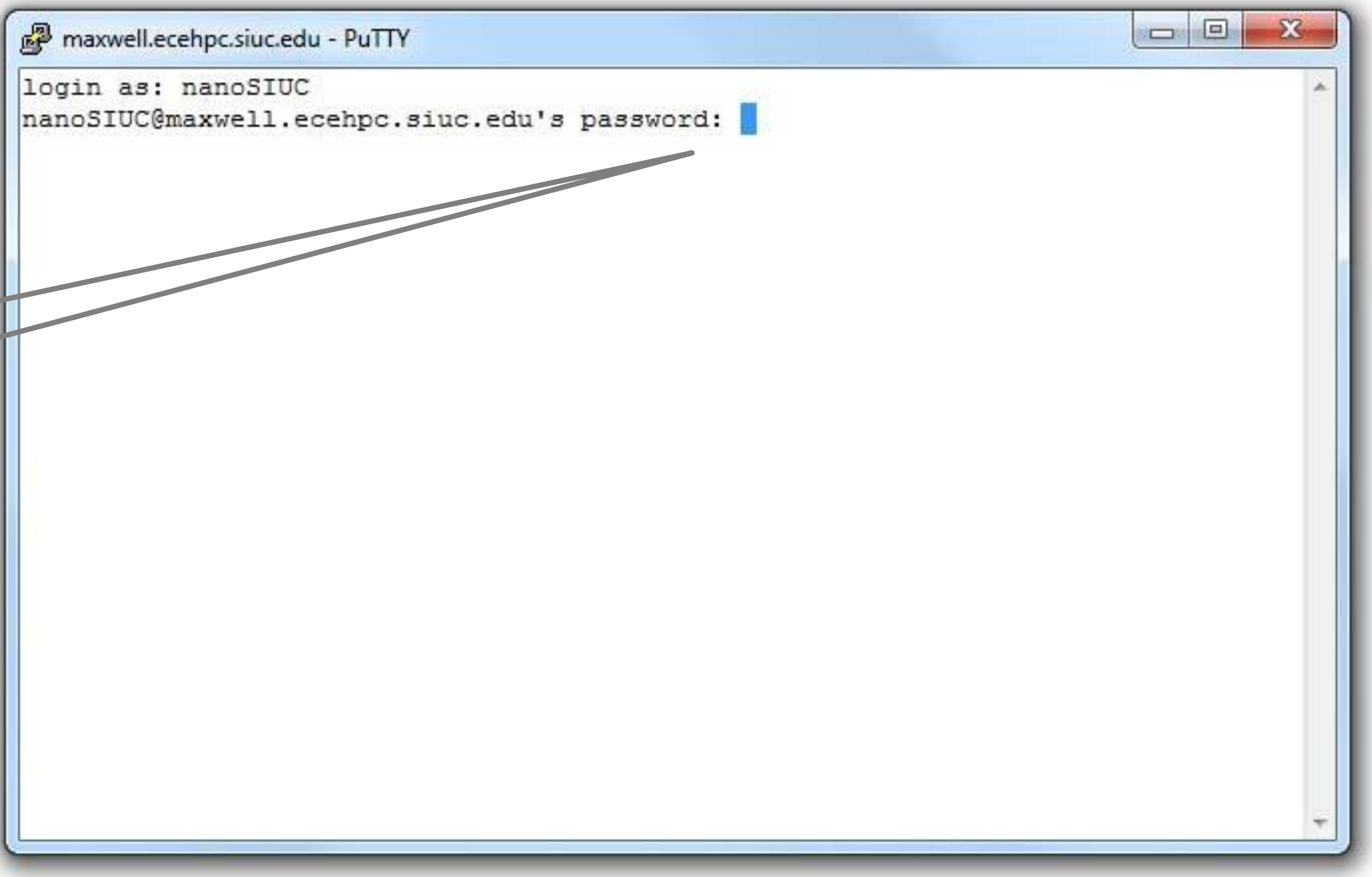
Login Procedure



```
login as: 
```

Type in Your Login ID
Example ‘nanoSIUC’

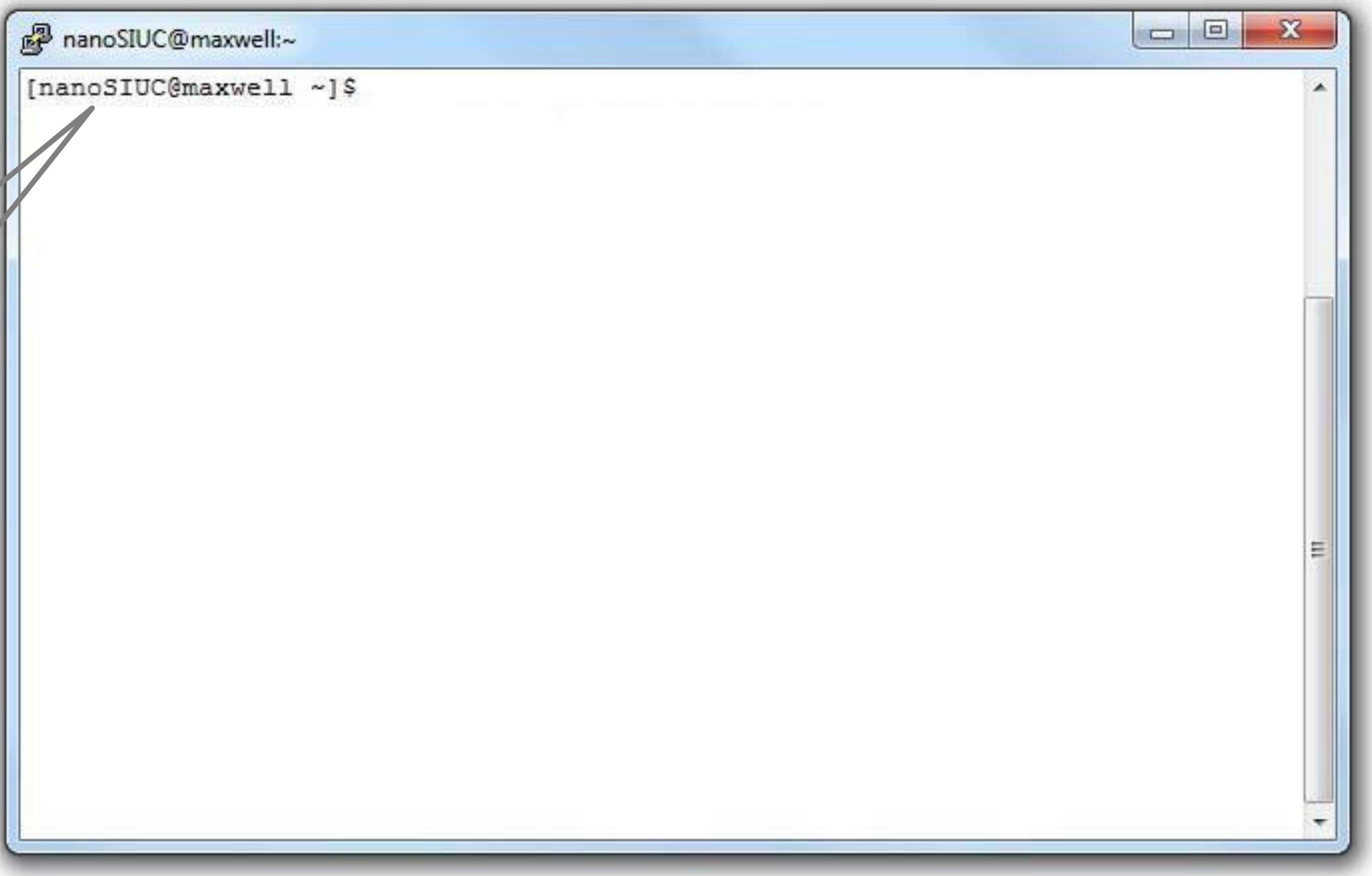
Login Procedure



```
login as: nanoSIUC
nanoSIUC@maxwell.ecehpc.siuc.edu's password:
```

Type in Password

Login Procedure



```
[nanoSIUC@maxwell ~]$
```

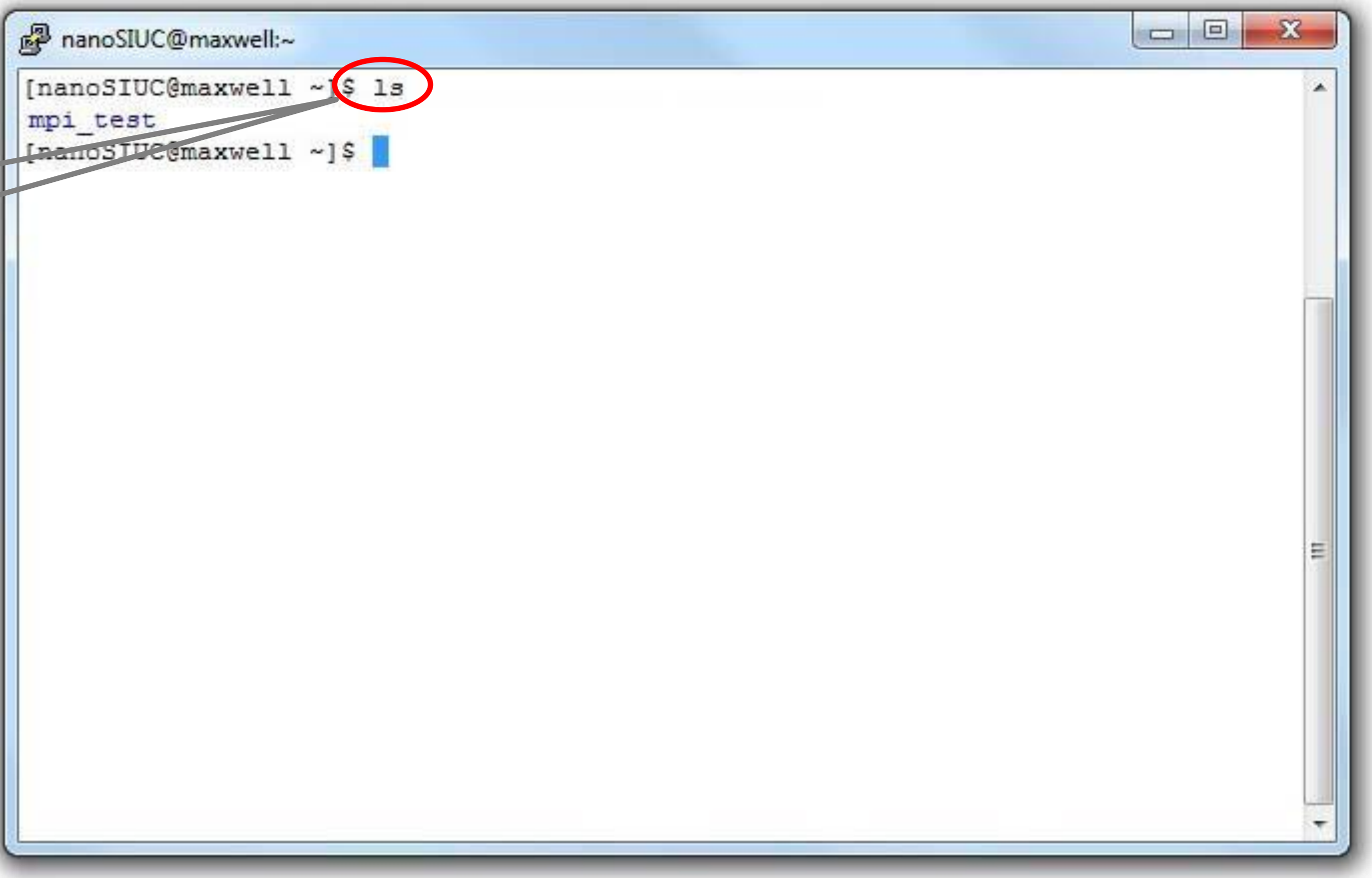
Prompt with your Login ID. Interface for User to use Maxwell

MPI Test

- Copy “mpi_test” directory to your “home” directory
  - Type in the following command

    -
      ```
      cp –r /home/nanoSIUC/mpi_test .
      ```

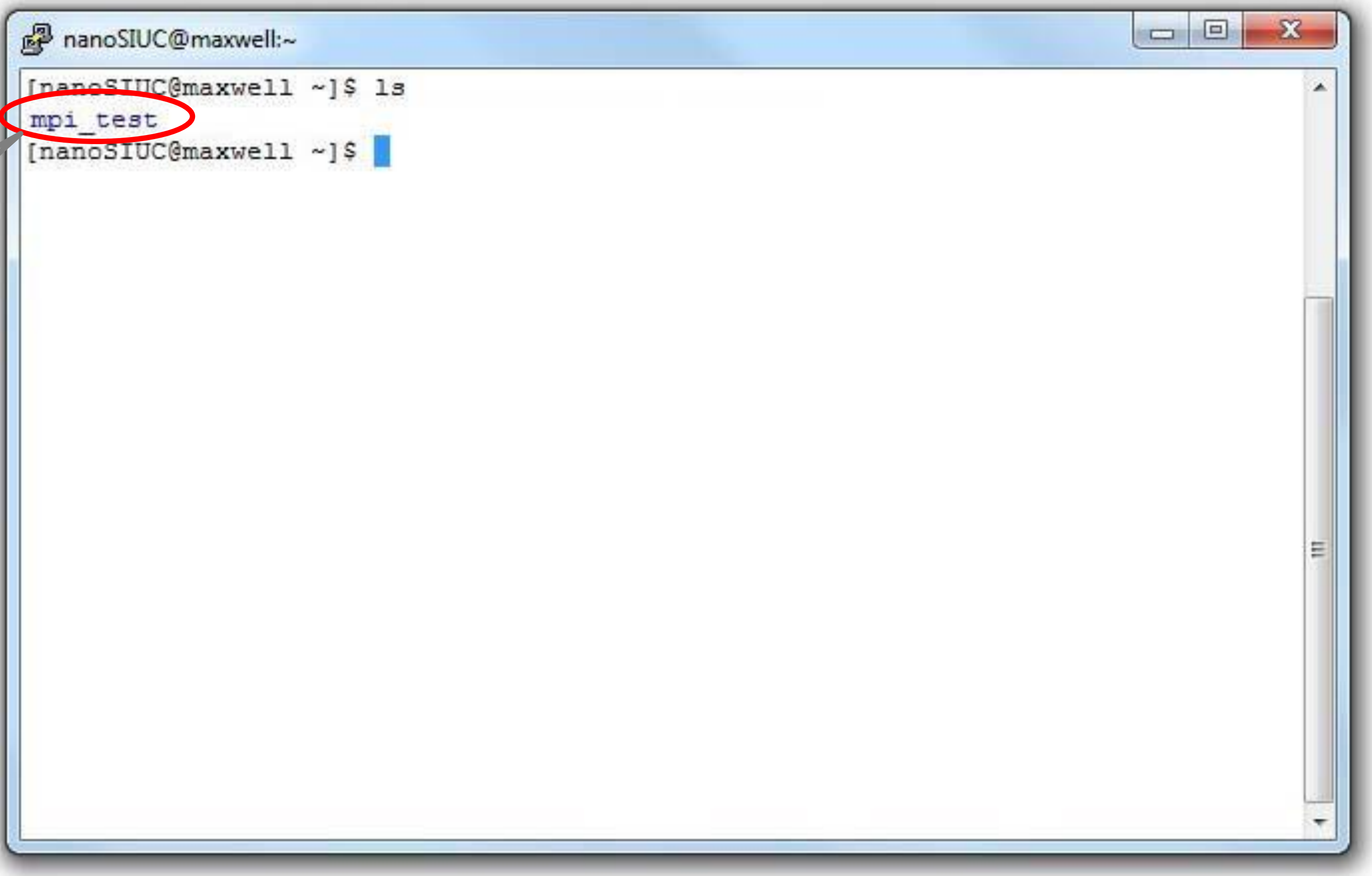
Run **MPI**



```
[nanoSIUC@maxwell ~]$ ls
mpi_test
[nanoSIUC@maxwell ~]$
```

Type in ‘ls’ to view the content of your account

Run **MPI**

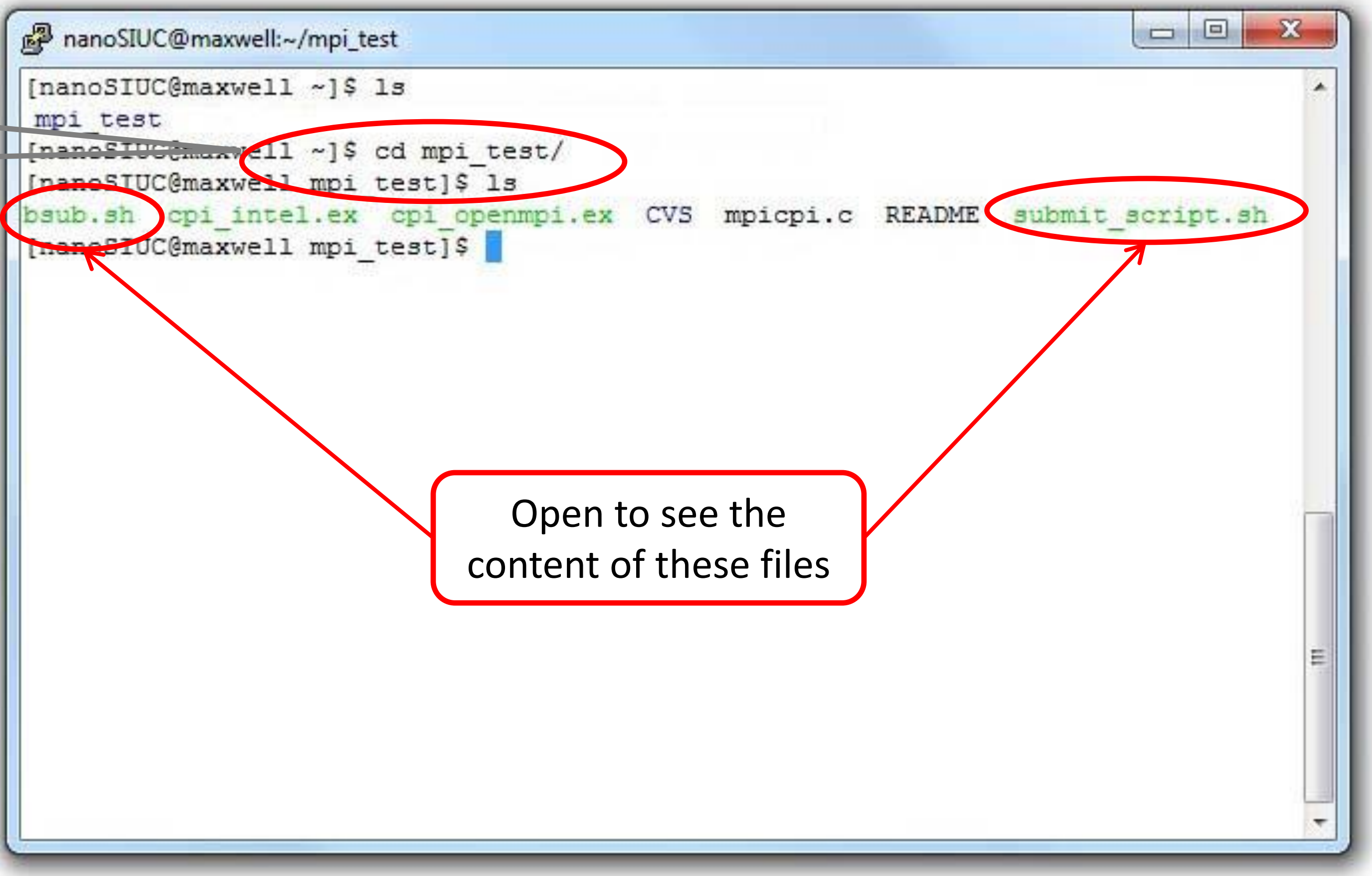


```
[nanoSIUC@maxwell ~]$ ls
mpi_test
[nanoSIUC@maxwell ~]$
```

Folder containing files to test MPI

Run **MPI**

Change Directory and view the files in ‘mpi_test’

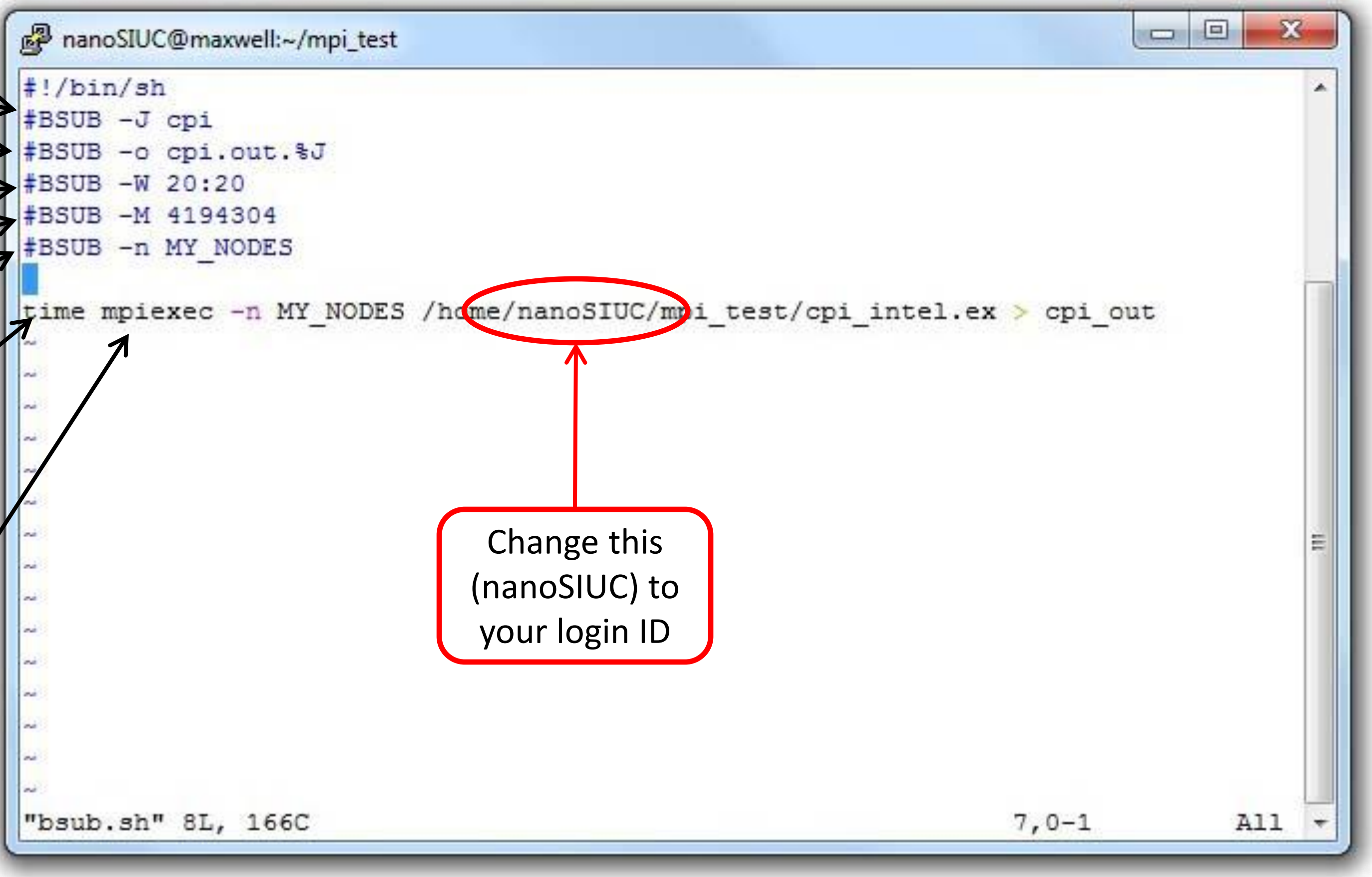


```
[nanoSIUC@maxwell ~]$ ls
mpi_test
[nanoSIUC@maxwell ~]$ cd mpi_test/
[nanoSIUC@maxwell mpi_test]$ ls
bsub.sh  cpi_intel.ex  cpi_openmpi.ex  CVS  mpicpi.c  README  submit_script.sh
[nanoSIUC@maxwell mpi_test]$ 
```

Open to see the content of these files

Run **MPI**

- File: ‘bsub.sh’



```
#!/bin/sh
#BSUB -J cpi
#BSUB -o cpi.out.%J
#BSUB -W 20:20
#BSUB -M 4194304
#BSUB -n MY_NODES

time mpiexec -n MY_NODES /home/nanoSIUC/mpi_test/cpi_intel.ex > cpi_out
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"bsub.sh" 8L, 166C                                          7,0-1         All
```

job_name

output_file

run_limit

memory_limit

processor_use

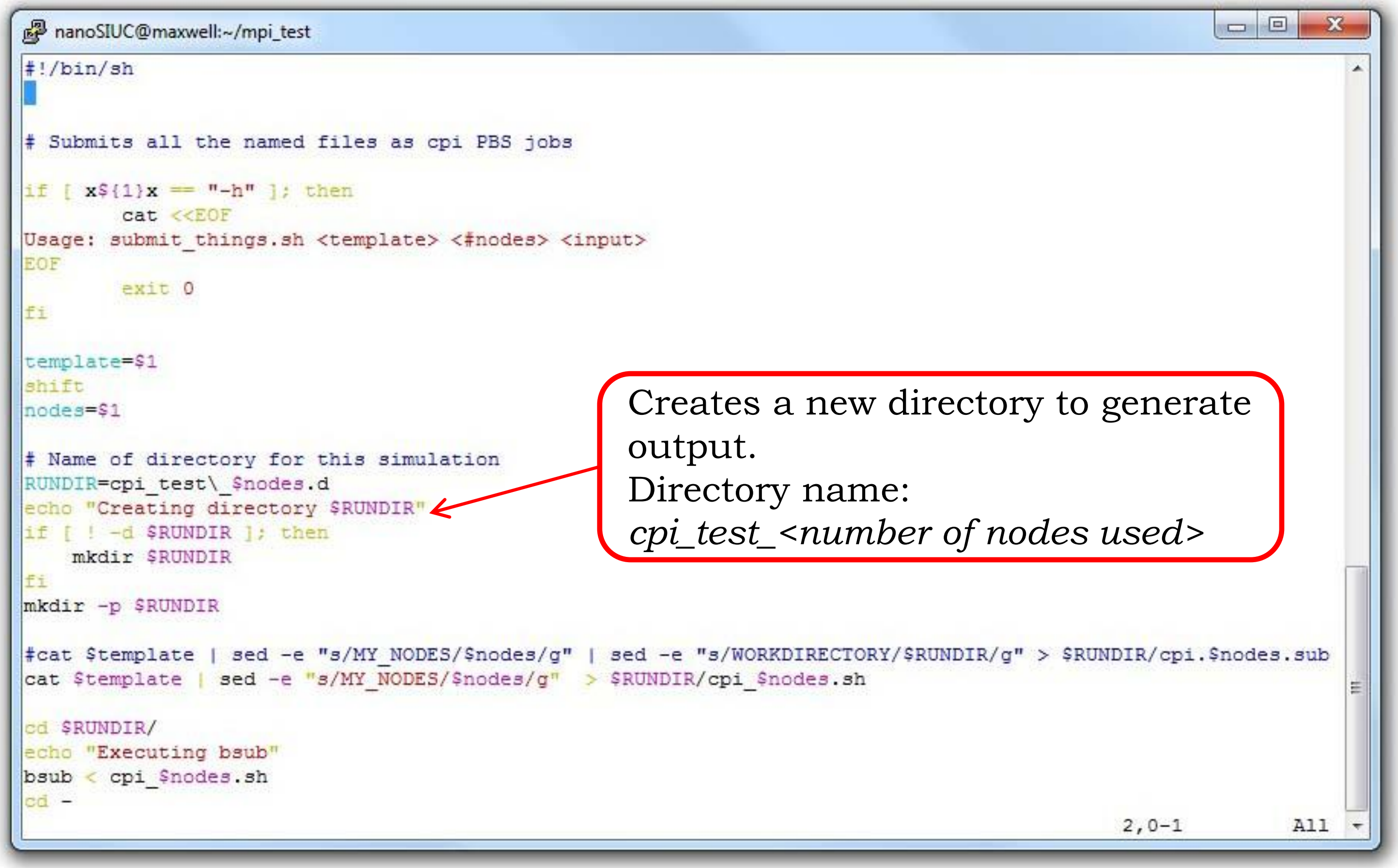
run_time

to run a parallel program by MPI

Change this (nanoSIUC) to your login ID

Run **MPI**

- File: ‘submit_script.sh’



```
#!/bin/sh


# Submits all the named files as cpi PBS jobs

if [ x${1}x == "-h" ]; then
        cat <<EOF
Usage: submit_things.sh <template> <#nodes> <input>
EOF
        exit 0
fi

template=$1
shift
nodes=$1

# Name of directory for this simulation
RUNDIR=cpi_test\_$nodes.d
echo "Creating directory $RUNDIR"
if [ ! -d $RUNDIR ]; then
    mkdir $RUNDIR
fi
mkdir -p $RUNDIR

#cat $template | sed -e "s/MY_NODES/$nodes/g" | sed -e "s/WORKDIRECTORY/$RUNDIR/g" > $RUNDIR/cpi.$nodes.sub
cat $template | sed -e "s/MY_NODES/$nodes/g"  > $RUNDIR/cpi_$nodes.sh

cd $RUNDIR/
echo "Executing bsub"
bsub < cpi_$nodes.sh
cd -
                                                                                2,0-1          All
```

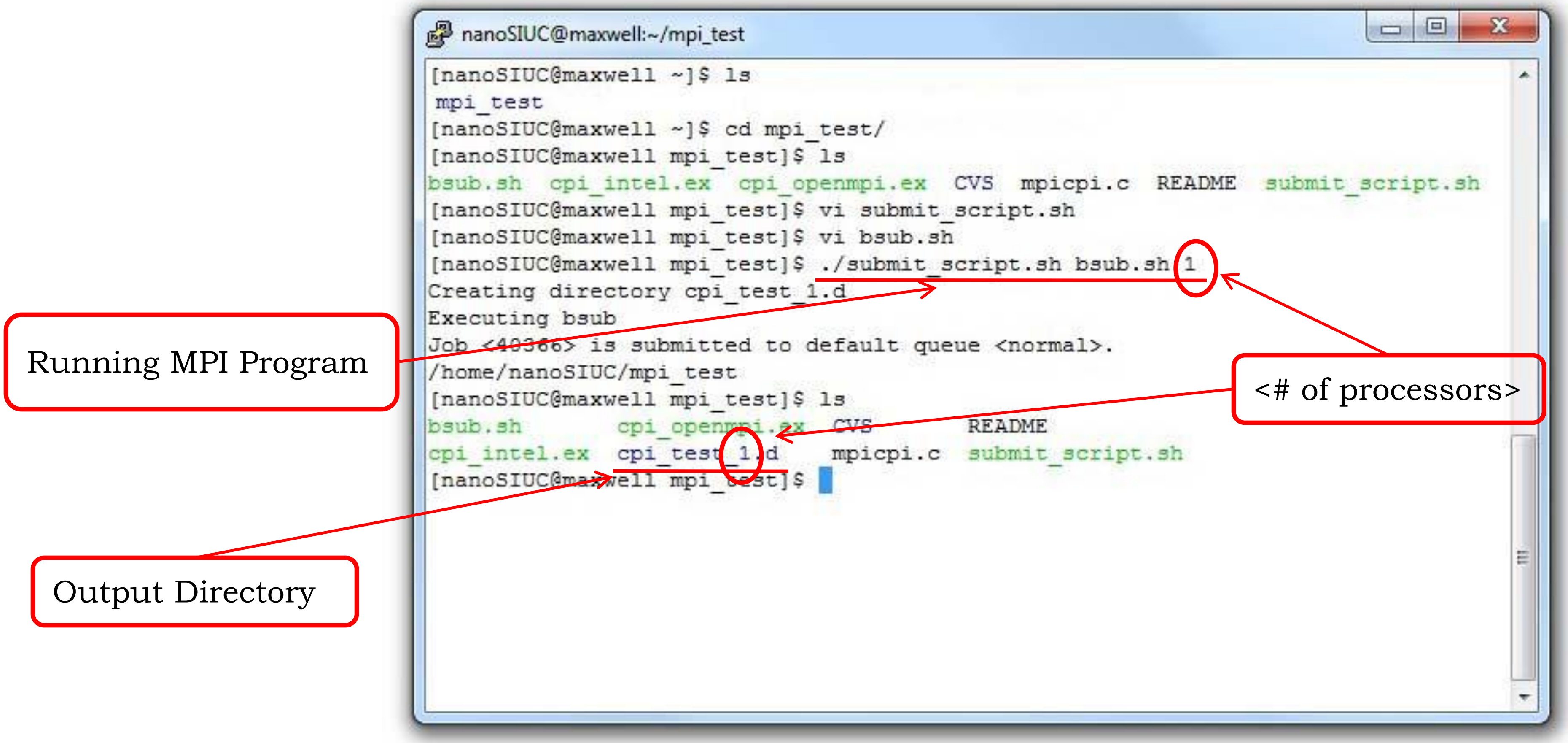
Creates a new directory to generate output.
Directory name:
cpi_test_<number of nodes used>

Run **MPI**

- Script to Run MPI
  - ./submit_script.sh bsub.sh <# of Processors> <input file>
  - <# of Processors> is an integer
  - <input file> is optional.
    - If in different directory, use the path of the input file as well

Run **MPI**

- Script to Run MPI



```
[nanoSIUC@maxwell ~]$ ls
 mpi_test
[nanoSIUC@maxwell ~]$ cd mpi_test/
[nanoSIUC@maxwell mpi_test]$ ls
bsub.sh  cpi_intel.ex  cpi_openmpi.ex  CVS  mpicpi.c  README  submit_script.sh
[nanoSIUC@maxwell mpi_test]$ vi submit_script.sh
[nanoSIUC@maxwell mpi_test]$ vi bsub.sh
[nanoSIUC@maxwell mpi_test]$ ./submit_script.sh bsub.sh 1
Creating directory cpi_test_1.d
Executing bsub
Job <40366> is submitted to default queue <normal>.
/home/nanoSIUC/mpi_test
[nanoSIUC@maxwell mpi_test]$ ls
bsub.sh       cpi_openmpi.ex  CVS       README
cpi_intel.ex  cpi_test_1.d    mpicpi.c  submit_script.sh
[nanoSIUC@maxwell mpi_test]$
```

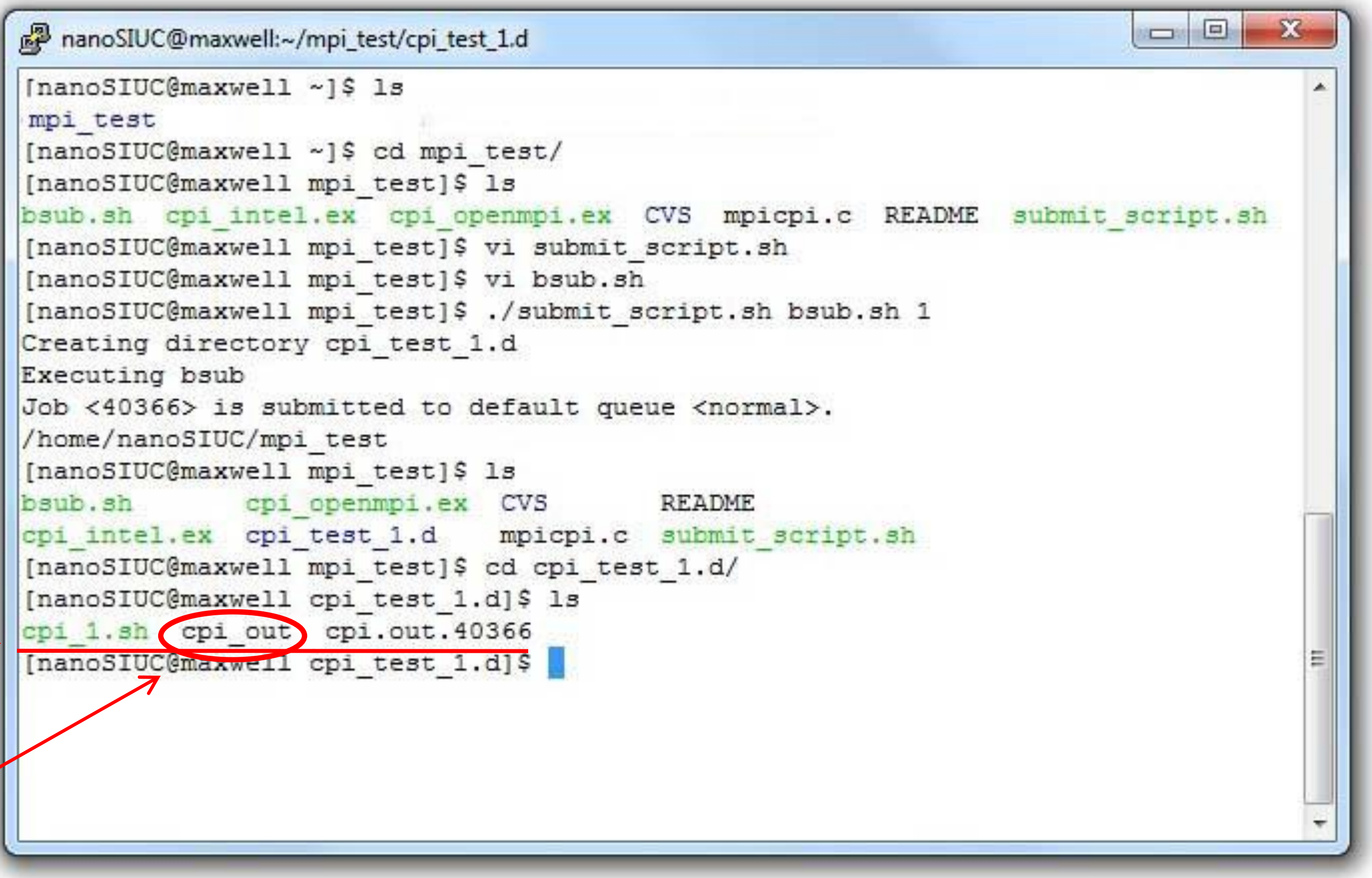
Running MPI Program

<# of processors>

Output Directory

Run **MPI**

- Viewing Output



nanoSIUC@maxwell:~/mpi_test/cpi_test_1.d

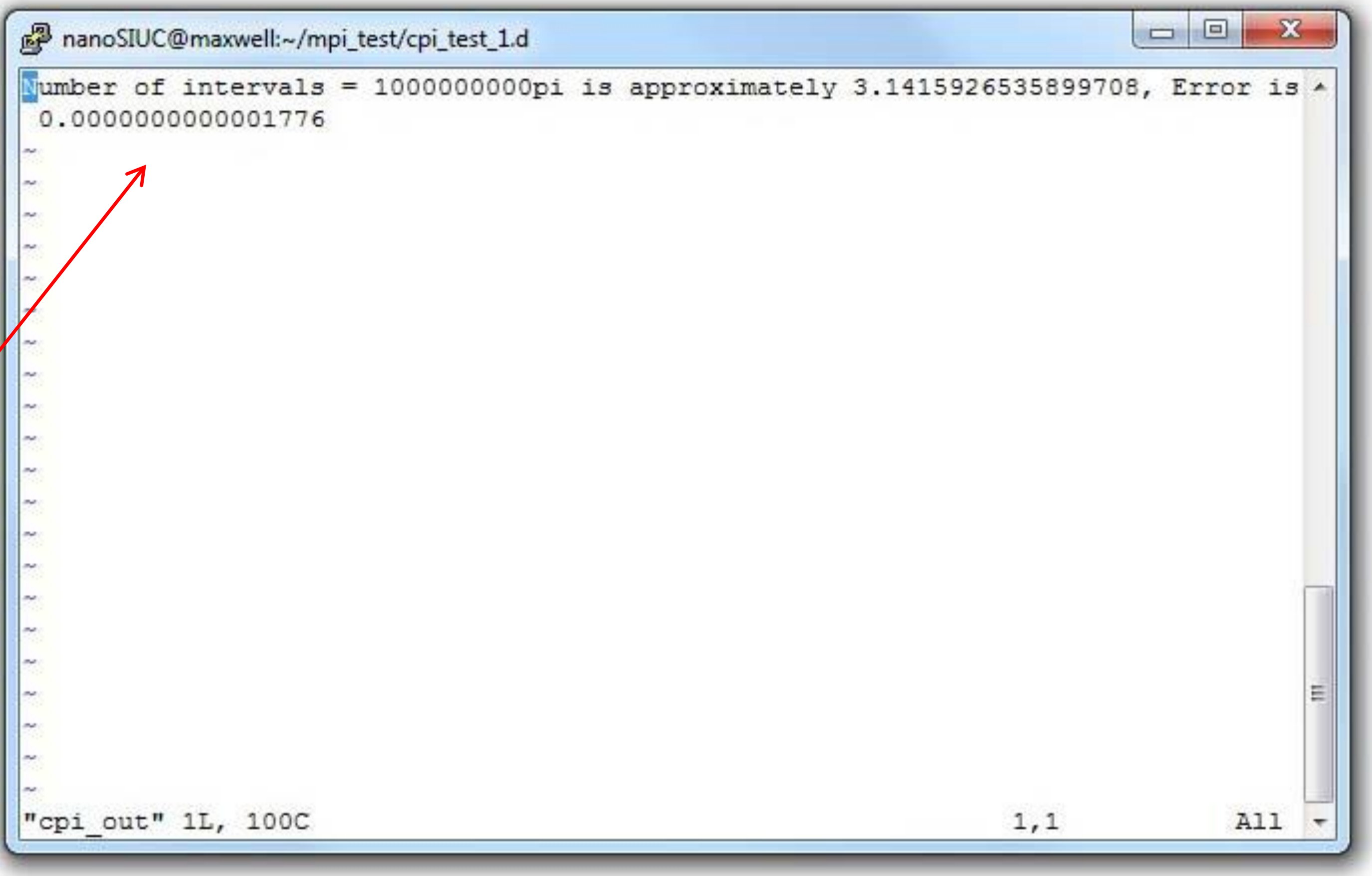
```
[nanoSIUC@maxwell ~]$ ls
mpi_test
[nanoSIUC@maxwell ~]$ cd mpi_test/
[nanoSIUC@maxwell mpi_test]$ ls
bsub.sh  cpi_intel.ex  cpi_openmpi.ex  CVS  mpicpi.c  README  submit_script.sh
[nanoSIUC@maxwell mpi_test]$ vi submit_script.sh
[nanoSIUC@maxwell mpi_test]$ vi bsub.sh
[nanoSIUC@maxwell mpi_test]$ ./submit_script.sh bsub.sh 1
Creating directory cpi_test_1.d
Executing bsub
Job <40366> is submitted to default queue <normal>.
/home/nanoSIUC/mpi_test
[nanoSIUC@maxwell mpi_test]$ ls
bsub.sh       cpi_openmpi.ex  CVS       README
cpi_intel.ex  cpi_test_1.d    mpicpi.c  submit_script.sh
[nanoSIUC@maxwell mpi_test]$ cd cpi_test_1.d/
[nanoSIUC@maxwell cpi_test_1.d]$ ls
cpi_1.sh  cpi_out  cpi.out.40366
[nanoSIUC@maxwell cpi_test_1.d]$
```

Content of Output Directory

Output File

Run **MPI**

- Viewing Output



```
Number of intervals = 1000000000pi is approximately 3.1415926535899708, Error is
 0.0000000000001776
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"cpi_out" 1L, 100C                                                  1,1          All
```

Content of Output File

The End