Training for New Mexico State University

advanced clustering
technologies, inc.

2

Agenda

- Cluster hardware overview
- Connecting to the system
- Advanced Clustering provided software
- Torque Scheduler

3

Cluster overview

- Head node (qty 1)
- Storage node (qty 1)
- GPU nodes (qty 5)
- FPGA nodes (qty 5)
- Infiniband network
- Gigabit network
- Management/IPMI network

4

Head Node

- Hardware Specs
  - Dual Eight-Core E5-2650v2 “Ivy Bridge” 2.6GHz processors
  - 128GB of RAM (8x 16GB DIMMs)
  - 2 x 1TB HDD (RAID 1 mirror)
- Hostname / networking
  - bigdat.nmsu.edu / 128.123.210.57
  - private: 10.1.1.254
  - IPoIB: 10.3.1.254
  - ipmi: 10.2.1.253
- Roles
  - DHCP/TFTP servers
  - GridEngine qmaster
  - Nagios management / monitoring system
  - Ganglia monitoring system

5

Storage Node

- Hardware Specs
  - Dual Twelve-Core E5-2695v2 “Ivy Bridge” 2.4GHz processors
  - 256GB of RAM (16x 16GB DIMMs)
  - 30 x 4TB HDD (RAID 6 mirror, w/ Hostpare) approx 100TB usable
    - 2x SSD drives as cache for RAID array to improve performance
- Hostname / networking
  - storage
  - Private: 10.1.1.252
  - IPoIB: 10.3.1.252
  - ipmi: 10.2.1.252
- Roles
  - NFS server exports /home

6

GPU Nodes (qty 5)

- Hardware Specs
  - Dual Twelve-Core E5-2695v2 “Ivy Bridge” 2.4GHz processors
  - gpu01-gpu03 : 256GB of RAM (16x 16GB DIMMs)
  - gpu04-gpu05: 256GB of RAM (8x 32GB DIMMs)
  - 3x 3TB in RAID5 /data-hdd
  - 3x 512GB SSDs in RAID5 /data-ssd
  - 1x Tesla K40 GPU (12GB of GDDR5)
- Hostname / networking
  - gpu01 - gpu05
  - private: 10.1.1.1 - 10.1.1.5
  - IPoIB: 10.3.1.1 - 10.3.1.5
  - ipmi: 10.2.1.1 - 10.2.1.5

7

FPGA Nodes (qty 5)

- Hardware Specs
  - Dual Twelve-Core E5-2695v2 “Ivy Bridge” 2.4GHz processors
  - fpga01-fpga03 : 256GB of RAM (16x 16GB DIMMs)
  - fpga04-fpga05: 256GB of RAM (8x 32GB DIMMs)
  - 3x 3TB in RAID5 /data-hdd
  - 3x 512GB SSDs in RAID5 /data-ssd
  - Space to add future FPGAs
- Hostname / networking
  - fpga01 - fpga05
  - private: 10.1.1.21 - 10.1.1.25
  - IPoIB: 10.3.1.21 - 10.3.1.25
  - ipmi: 10.2.1.21 - 10.2.1.25

8

Shared storage

- Multiple filesystems are shared across all nodes in the cluster
  - /home = home directories (from storage)
  - /opt = 3rd party software and utilities (from head)
  - /act = Advanced Clustering provided tools (from head)
- /home/[USERNAME] is for all your data
  - LVM currently sized to 10TB of usable space, ~90TB free to allocate

9

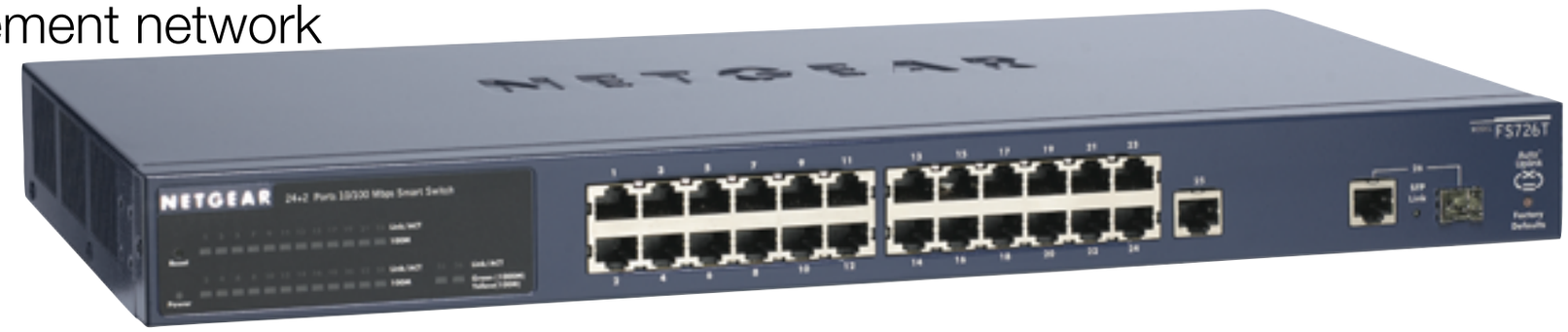
Gigabit Ethernet network

- 1x 48 port Layer 2 managed switch
- Network switch and cluster network completely isolated from main university network
- Black color ethernet cables used for gigabit network

10

Management / IPMI network

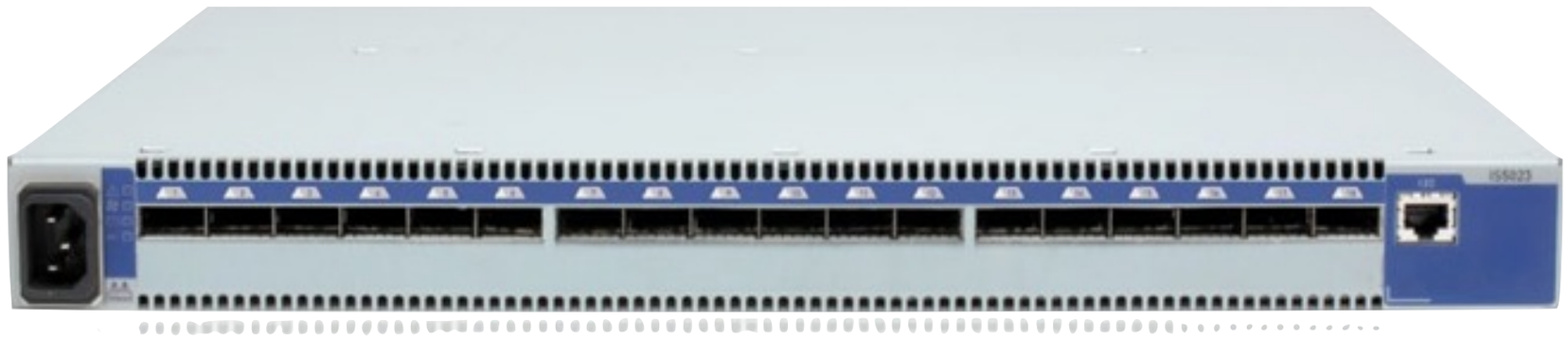
- 1x 48 port 10/100Mb ethernet switch
- Each server has a dedicated 100Mb IPMI network interface that is independent of the host operating system
- Red color ethernet cables used for management network

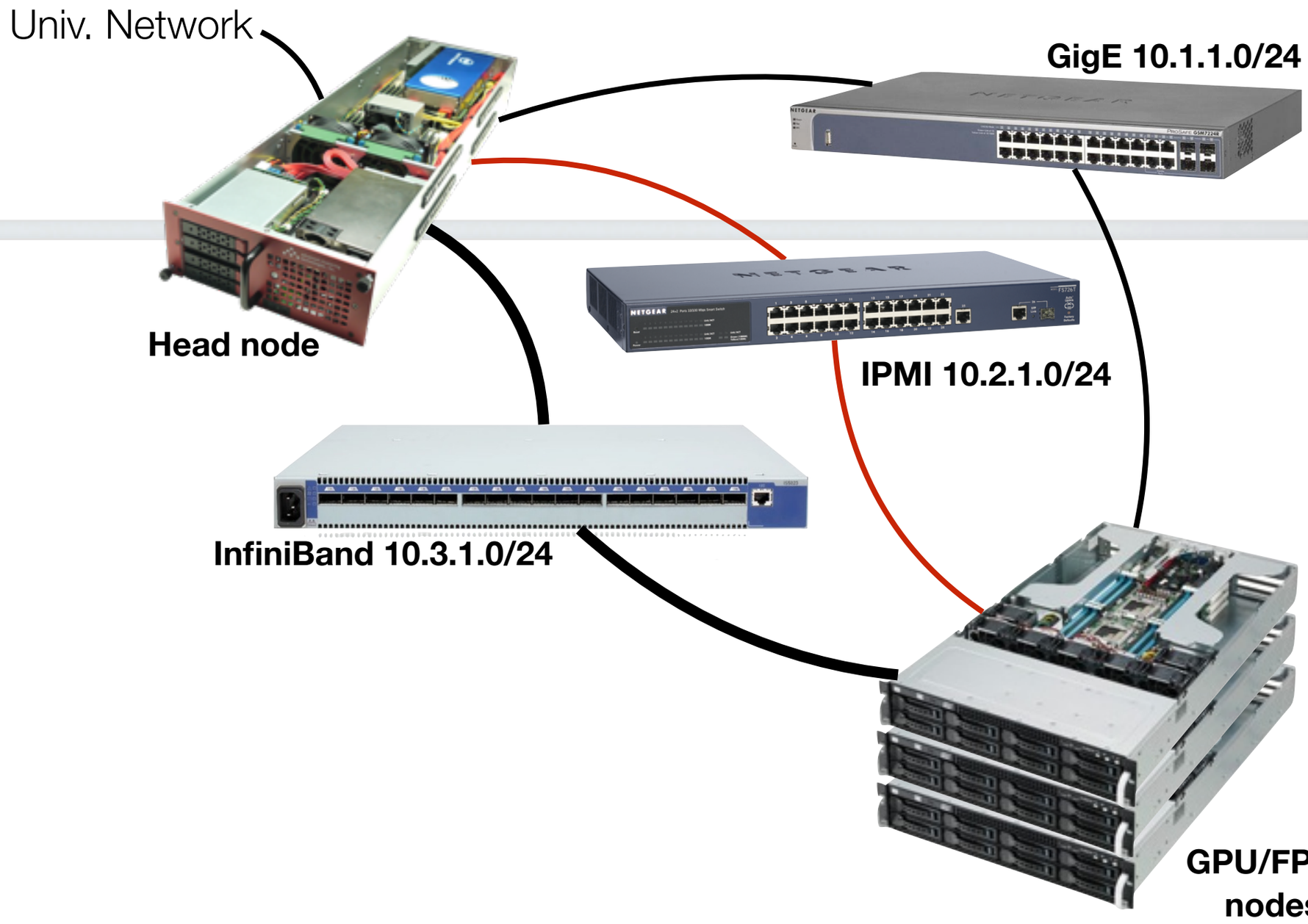


11

InfiniBand network

- 36 port QDR InfiniBand switch
  - 40Gb/s data rate
  - ~ 1.5us latency
- 12 ports utilized (10 nodes, 1 head, 1 storage)
- Switch is “dumb” no management, subnet manager runs on head node




Univ. Network
GigE 10.1.1.0/24
Head node
IPMI 10.2.1.0/24
InfiniBand 10.3.1.0/24
GPU/FPGA
nodes

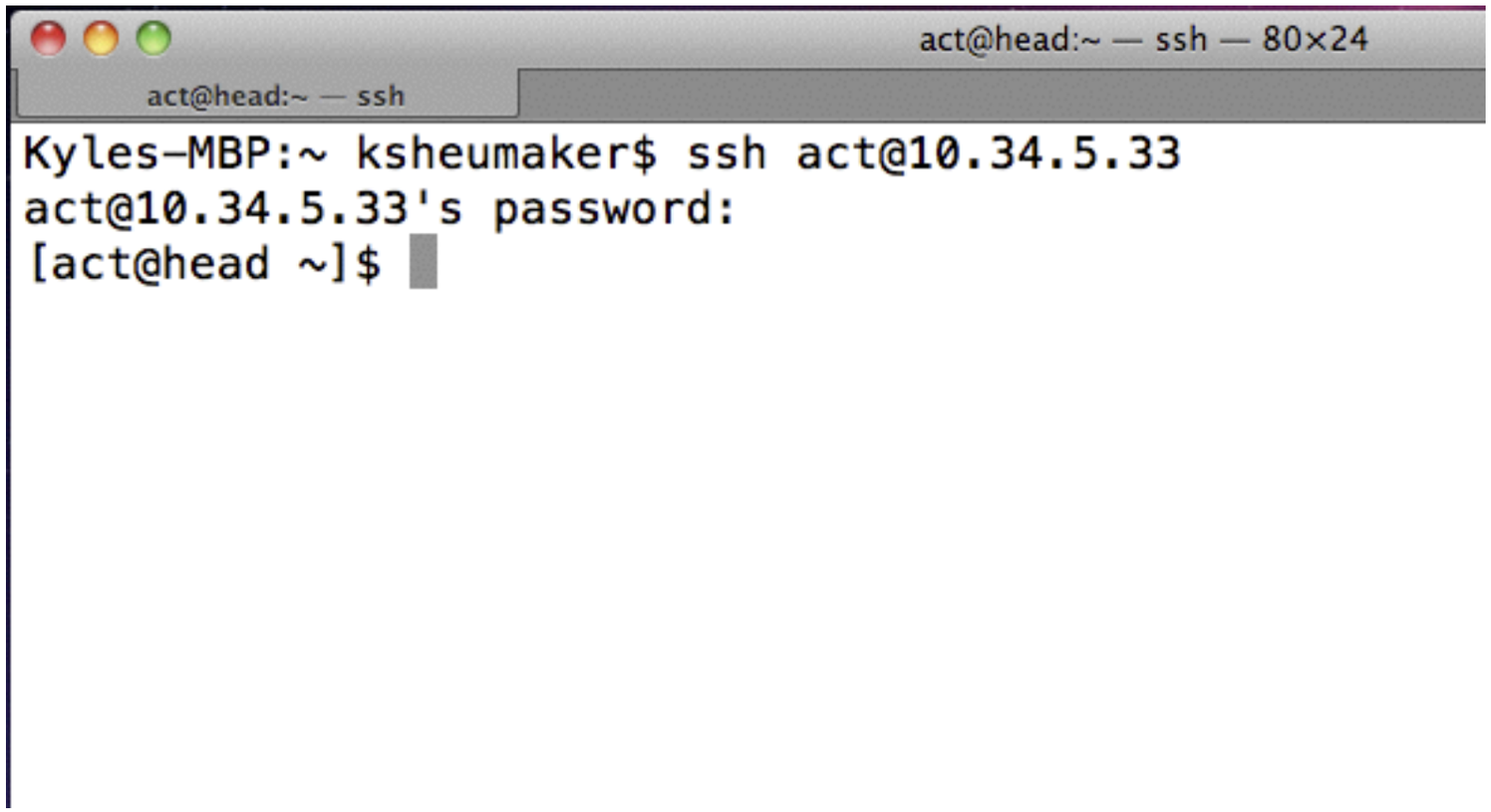
13

Connecting to the system

- Use SSH to login to system
  - client builtin to Mac OSX and Linux
  - Multiple options available for windows
    - Putty - http://www.chiark.greenend.org.uk/~sgtatham/putty/
- Use SFTP to copy files
  - Multiple options available
    - FileZilla - http://filezilla-project.org/
- Hostname/IP address: bigdat.nmsu.edu

Connecting to the system (cont.)

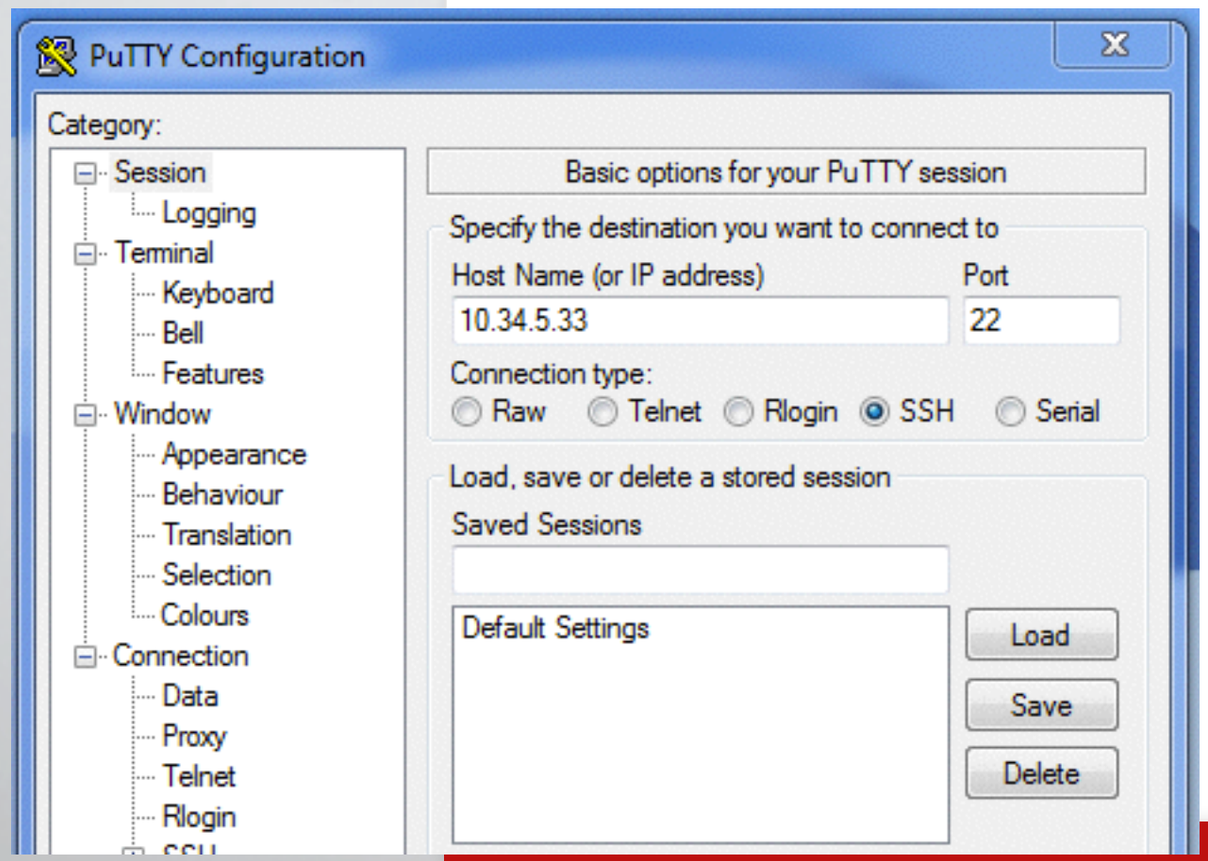
- Example using MacOSX / Linux

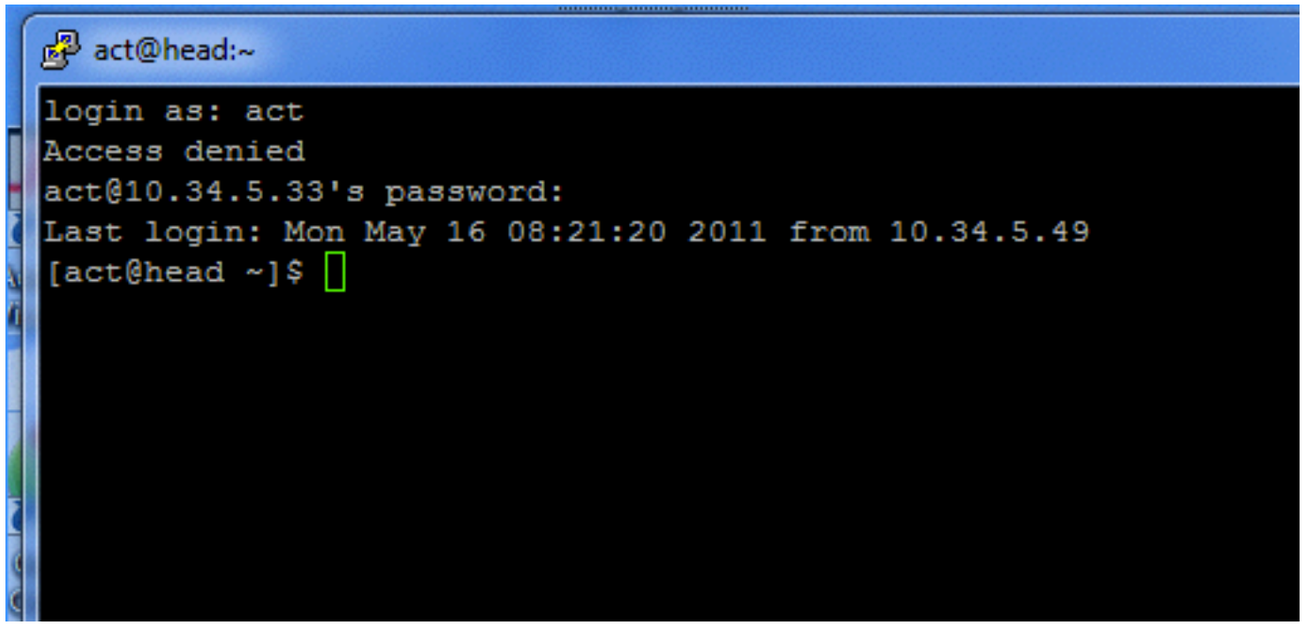


15

Connecting to the system (cont.)

- Example using Putty under windows

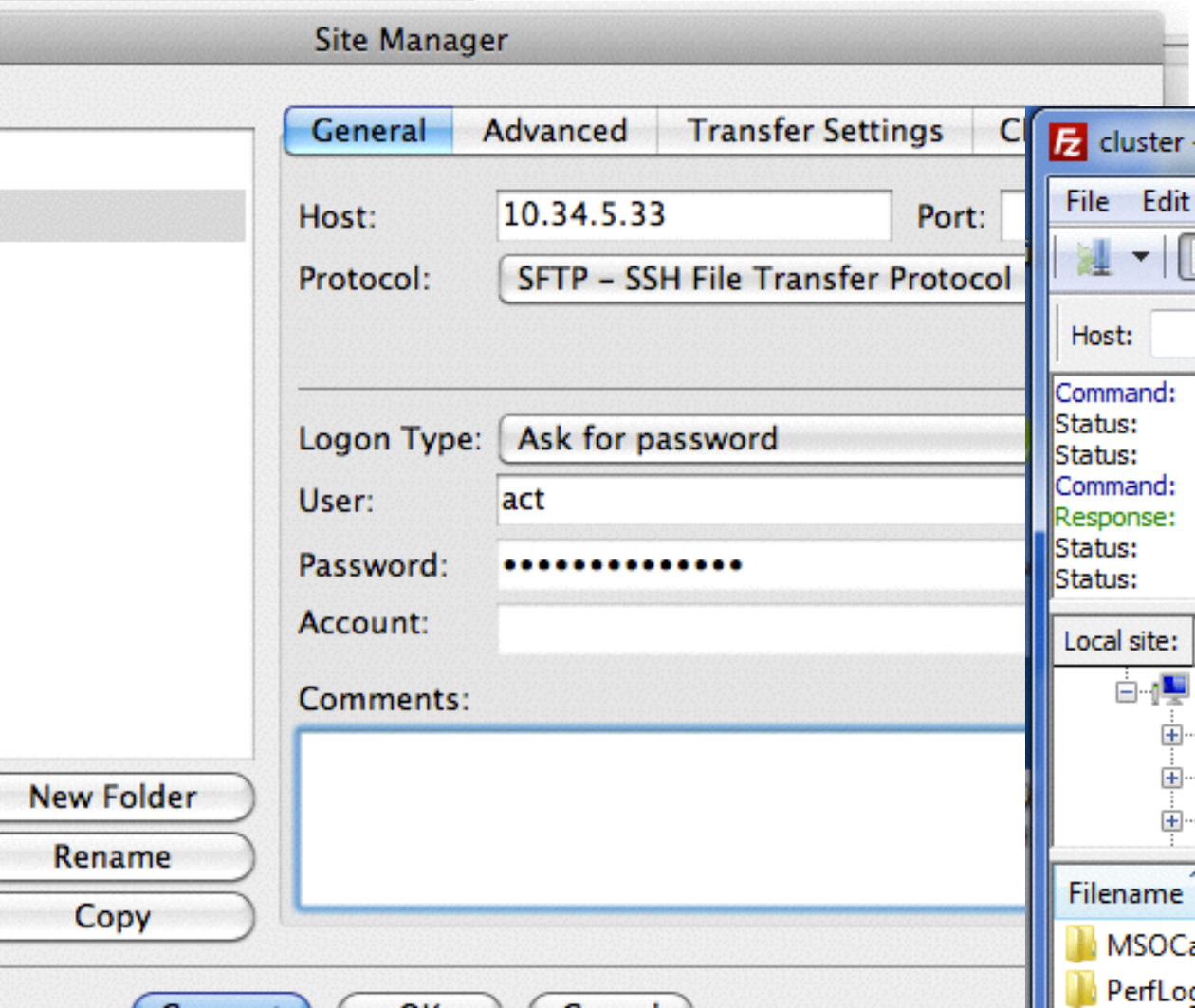




16

Connecting to the system (cont.)

- Copying files via FileZilla



17

Advanced Clustering Software

- Modules
- ACT Utils
- Cloner
- Nagios
- Ganglia
- Torque / PBS
- eQUEUE

18

Modules command

- Modules is an easy way to setup the user environment for different pieces of software (path, variables, etc).
- Setup your .bashrc or .cshrc
  - source /act/etc/profile.d/actbin.[sh|csh]
  - source /act/Modules/3.2.6/init/[bash|csh]
  - module load null

19

Modules continued

- To see what modules you have available:
  - module avail
- To load the environment for a particular module:
  - module load modulename
- To unload the environment:
  - module unload modulename
  - module purge (removes all modules from environment)
- Modules are stored /act/modulefiles/ - can customize for your own software (modules use tcl language)

20

ACT Utils

- ACT Utils is a series of commands to assist in managing your cluster, the suite contains the following commands:
  - act_authsync - sync user/password/group information across nodes
  - act_cp - copy files across nodes
  - act_exec - execute any Linux command across nodes
  - act_netboot - change network boot functionality for nodes
  - act_powerctl - power on, off, or reboot nodes via IPMI or PDU
  - act_sensors - retrieve temperatures, voltages, and fan speeds
  - act_console - connect to the hosts's serial console via IPMI [escape sequence: enter enter ~ ~ .]

21

ACT Utils common arguments

- All utilities have a common set of command line arguments that can be used to specify which nodes to interact with
  - --all all nodes defined in the configuration file
  - --exclude a comma separated list of nodes to exclude from the command
  - --nodes a comma separated list of node hostnames (i.e. node01,node04)
  - --groups a comma separated list of group names (i.e. nodes, storage, etc)
  - --range a “range” of nodes (i.e. node01-node04)
- Configuration (including groups and nodes) defined in /act/etc/act_utils.conf

22

Groups defined on your cluster

- gpu - gpu01-gpu03
- gpuhimem - gpu04-gpu04
- fpga - fpga01-fpga03
- fpgahimem - fpga04-fpga05
- nodes - all gpu nodes & all fpga nodes

23

ACT Utils examples

- Find the current load on all the compute nodes
  - act_exec -g nodes uptime
- Copy the /etc/resolv.conf file to all the login nodes
  - act_cp -g nodes /etc/resolv.conf /etc/resolv.conf
- Shutdown every compute node except node04
  - act_exec --group=nodes --exclude=node04 /sbin/poweroff
- tell nodes node01 and node03 to boot into cloner on next boot
  - act_netboot --nodes=node01,node03 --set=cloner

24

Shutting the system down

- To shut the system down for maintenance (run from head node):
  - act_exec -g nodes /sbin/poweroff
- Make sure you shut down the head node last by just issuing the poweroff command
  - /sbin/poweroff

25

Cloner

- Cloner is used to easily replicate and distribute the operating system and configuration to nodes in a cluster
- Two main components:
  - Cloner image collection command
  - A small installer environment that is loaded via TFTP/PXE (default), CD-ROM, or USB key

Cloner image collection

- Login to the node you'd like to take an image of, and run the cloner command (this must execute on the machine you want to take the image, it does not pull the image, it pushes it to a server)
  - /act/cloner/bin/cloner --image=IMAGENAME --server=SERVER
- Arguments:
  - IMAGENAME = a unique identifier label for the type of image (i.e. node, login, etc)
  - SERVER = the hostname running the cloner rsync server process

27

Cloner data store

- Cloner data and images in /act/cloner/data
  - /act/cloner/data/hosts - individual files named with the system's hardware MAC address. These files files are used for auto-installation of nodes. File format include two lines:
    - IMAGE="imagename"
    - NODE="nodename"

Cloner data store

- /act/cloner/data/images - a sub directory is automatically created for each image with the name specified with the --image argument when creating
  - sub directory called “data” that contains the actual cloner image (an rsync’d copy of the system cloned).
  - sub directory called “nodes” and a series of subdirectories with the name of each node (i.e. data/images/storage/nodes/node01, data/images/storage/nodes/node02)
    - the node directory is an overlay that gets applied after the full system image
    - can be used for customizations that are specific to each node (i.e. hostname, network configuration, etc).

29

Creating node specific data

- ACT Utils includes a command to assist in creating all the node specific configuration for cloner (act_cfgfile)
- Example: create all the node specific information
  - act_cfgfile --cloner --prefix=/
  - Writes /act/cloner/data/images/IMAGENAME/nodes/NODENAME/etc/sysconfig/networking, ifcfg-eth0, etc for each node

30

Installing a cloner image

- Boot a node from the PXE server to start re-installation
- Use act_netboot to set the network boot option to be cloner
  - act_netboot --set=cloner3 --node=node01
- On next boot system will unattended install the new image
- After node re-installation will automatically reset image back to localboot

31

Ganglia

- Ganglia a cluster monitoring tool, is installed on the head node and available at:
  - http://bigdat.nmsu.edu/ganglia
- More information: http://ganglia.sourceforge.net/

32

Nagios

- Nagios, is a system monitoring tool which is used to monitor all nodes and send out alerts when there are problems
  - http://bigdat.nmsu.edu/nagios
  - username: nagiosadmin
  - password: cluster
- Configuration located in: /etc/nagios on the head node
- More information: http://www.nagios.org/

33

PBS/Torque introduction

- Basic setup
- Submitting serial jobs
- Submitting parallel jobs
- Job status
- Interactive jobs
- Managing the queuing system

34

Basic setup

- 3 main pieces
  - pbs_server - main server components responds to user commands, etc
  - pbs_sched - decide where to run jobs
  - pbs_mom - a daemon that runs on every node that will execute jobs
- Each node has 24 slots which can be used for any number of jobs
- There is currently only 1 queue, named “batch” it’s set as a FIFO (first-in first-out)

Torque node resources

- **gpu** - gpu01-gpu03
- **gpu-himem** - gpu04-gpu05
- **gpu-all** - all the GPU nodes gpu01-gpu05
- **fpga** - fpga01-fpga03
- **fpga-himem** - fpga04-fpga05
- **fpga-all** - all the FPGA nodes (fpga01-fpga05)
- **himem** - gpu04,gpu05 and fpga04,fpga05
- **hod** - all nodes to run Hadoop on Demand (currently all nodes)
- request as part of the -l argument
  - qsub -l nodes=2:ppn=24:gpu (2 nodes, 24 cores each node, with GPU)
  - qsub -l nodes=1:ppn=1:fpga (1 nodes, 1 core, with an FPGA)

36

Submitting batch jobs

- Basic syntax: qsub jobscript
- jobscripts are simple shell scripts in either SH or CSH which at a minimum contain the name of your program. Here is the minimum jobscript:

```
#!/bin/bash
/path/to/executable
```

37

Common qsub arguments

- -q queuename
  - name of the queue to run the job in
- -N jobname
  - a descriptive name of the job
- -o filename
  - path to the filename to write the contents of STDOUT
- -e filename
  - path of the filename to write the contents of STDERR

38

Common qsub arguments

- -j oe
  - Join the contents of STDERR and STDOUT into one file
- -m [a|b|e]
  - Send out e-mail at different states (a = job aborted, b = job begins, e = job ends)
- -M emailaddr
  - email address to send messages to
- -l resourcename=value,[resourcename=value]
  - a list of resources needed to run this job

39

Resource options

- walltime
  - maximum amount of real time the job can be running (if exceeded it will be terminated)
- mem
  - maximum amount of memory to be consumed by the job
- nodes
  - number of nodes requested to run this job

40

Submitting serial jobs

- A moderately complex job script can suggest command line parameters to PBS (prefixed with #PBS) that you may have left off of qsub as well as perform environment setup before running your program:

```
#!/bin/bash
#PBS -N testjob
#PBS -j oe
#PBS -q batch

echo Running on `hostname`.
echo It is now `date`.
sleep 60
echo It is now `date`.
```

41

Submitting parallel jobs

- Very similar to batch jobs except a new argument “-l nodes=X:ppn=X”
  - nodes: number of physical servers to run on
  - ppn: processors per node to run on, i.e. 8 to run on all 8 cores
- Examples:
  - run on 2 nodes using 8 cores per node, for a total of 16 cores: -l nodes=2:ppn=8
  - run on 4 nodes using 1 core per node, and 2 nodes using 2 cores per node: -l nodes=4:ppn=1+2:ppn=2

42

Job status

- You can check your own job submission status by looking at the output of "qstat".
- To examine the details of a jobs, use "qstat -f jobid"
- Common job states
  - R = running
  - Q = queued
  - E = error

43

Interactive jobs

- Using qsub -I you can submit an interactive job.
- When a job is scheduled, it lands you in a shell on the remote machine
- You can pass any argument that you'd normally pass to qsub (i.e. qsub -N name -I nodes=1:ppn=5).
- When you exit, the resources are immediately freed for others to use.

44

Managing the queuing system

- qdel - delete a job that has been submitted
- qalter - alter a job after submission
- qhold - hold a job in the queue and do not execute
- qrls - release a hold on a job
- pbsnodes - see nodes configured in the system
- pbsnodes -o nodename - take a node offline from the queuing system
- pbsnodes -c nodename - clear the offline state of a node
- qmgr - create queues and manage system properties

45

More information

- Administrator manual:
  - http://www.clusterresources.com/products/torque/docs/

46

eQUEUE

- Web-based job submission portal
- Remove visualization and GUI applications, without need for an X server
- Accounting log and analysis tool
- Links
  - View: http://bigdat.nmsu.edu/equeue
  - Manual: https://bigdat.nmsu.edu/equeue/docs/admin.html

The End

advanced clustering
technologies, inc.