Technical Architecture of a Computational Cluster

Greg Kurtzer
ITSD, LBNL
Scientific Cluster Support
GMKurtzer@lbl.gov

Introduction::Cluster

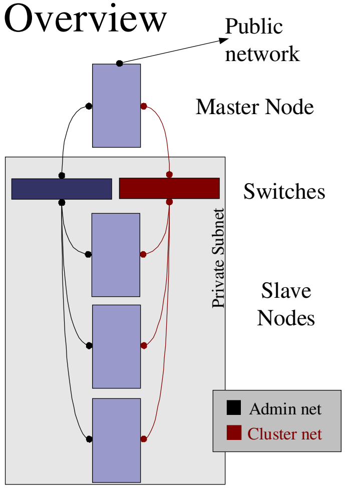
- The Cluster Defined
  - Literally a collection of systems coupled together for a common use
  - Many kinds of clusters
  - The Beowulf is the typical implementation for parallel computation
  - CPU farm is similar in architecture to the Beowulf, but for serial jobs
  - Usually implements some form of scheduler

Scientific Cluster Support

- We were funded to support Scientific Linux Clusters
- We developed an administration model that allows us to scale (so many clusters, to little time)
- This presentation is biased towards our standards and recommendations

Architecture::Overview

- Master/Slave node relationship
- Slaves networked behind master on fast switch
  - May have multiple private networks
  - Allows nodes to be trusting, without the security risks
- Slaves configured to trust the users on the master
- Typically built with COTS (Commodity off the shelf) hardware



Architecture::Master

- Cluster login gateway
- Slave node manager/monitor
- Open to the public network
- Multiple network interfaces
- Private subnet for slaves
- Developers tools
- Queuing submit host

Architecture::Slave

- May support logins (usually not recommended)
- Should be as dumb as possible
  - Less node based logic, the less that can go wrong
  - Lowers administration time
- OS is built and optimized to run jobs
- Usually high speed interconnect back to master

Architecture::Hardware::CPU

- AMD
  - Cheaper then Intel
  - Very fast floating point
  - Runs hot
- Intel
  - Expensive (especially 64bit)
  - Hyperthreading
  - Very good with Intel compilers
    - Makes use of CPU optimizations
    - May not compile Fortran77

Architecture::Hardware::Interconnects

- 100BaseT
  - Standard Fast Ethernet
  - Cheap (Very cheap)!
  - Uses:
    - Administration network of cluster
    - Lightly loaded CPU farms
    - Some embarrassingly parallel jobs
    - Very low bandwidth/chit-chat message passing

Architecture::Hardware::Interconnects

- 1000BaseT
  - Standard Faster Ethernet
  - Cheap (unless it is a large cluster)
  - Uses:
    - CPU farms
    - Embarrassingly parallel jobs
    - Low chit-chat message passing

Architecture::Hardware::Interconnects

- Myrinet
  - Proprietary Interconnect
  - Works very well with Linux
  - Uses:
    - CPU farms with large file transfers
    - Embarrassingly parallel jobs
    - Message passing

Architecture::Hardware::Interconnects

- Infiniband
  - Not in wide use yet
  - Open standard – Not proprietary!
  - Creates a unified I/O fabric between nodes
  - New interconnect design

Architecture::Other

- Typically >= 2Gb of RAM
- Floppy is recommended (boot/flash medium)
- CDROM/Disk drive is optional
- VGA is recommended
- 100BaseT is required (for administration/booting)
  - GigE (sometimes) and Myrinet do not network boot
- KVM is recommended

Software::Overview

- Message Passing
  - MPI (Message Passing Interface)
  - PVM (Parallel Virtual Machine)
- Scheduler (we use the GridEngine)
- Monitoring
- Cluster Distribution

Software::Message Passing

- MPI
  - MPI is a SPEC or Standard
  - Several Implementations
  - Only free solutions are widely used on Linux
    - MPICH
    - LAM-MPI
  - All Implementations should be portable

Software::Message Passing::MPICH

- Lots of third party modified versions (good and bad)
  - Very good Myrinet (GM) support
  - GridEngine (tightly integrated support)
  - Various companies distribute/require hacked versions
- Uses rsh/ssh to start jobs

Software::Message Passing::LAM

- Runs as a user invoked daemon and handles all nodes LAM processes for the user (lamboot)
- Only uses rsh/ssh to start daemons
- Consistently faster in our tests
- Very stable and reliable
- Our MPI of choice

Software::Message Passing::PVM

- One of the early message passing implementations
- Only one implementation available
- Less support in the community
- Seems to be going out of trend

Software::Scheduler::SGE

- We choose the Sun GridEngine
  - Truly free (under OSI approved license)
  - Had more features then the others
  - Already used by scientific groups we were working with
- Used for batch scheduling

Software::Scheduler::SGE::Overview

- Each queue is associated with a node (ie. node001.q)
- Nodes can have multiple queues
- Each queue contains a number of slots (for jobs)
- Nodes have limits that can prevent scheduling
  - Smarter then most schedulers
  - Able to utilize the most from your hardware without over utilizing the system
- Queues can be subordinate to others (high/low priority)

Software::Scheduler::SGE::qsub

- Scheduler accepts jobs in the form of scripts
  - 'qsub' command submits a job
  - 'qsub' takes a script from <STDIN> or as a filename argument
  - Must be a script (text) because it is stored in memory
    - Script can be any interpreted scripting language as long as the interpreter (#!/[.+$]) is present on the node
  - Script can be literally 2 lines
    - Interpreter definition
    - Command defined with the full paths to program and arguments

Software::Scheduler::SGE::qsub::script

```
#!/bin/sh
#$ -N SETI
NODE=`hostname`
export NODE
cp -r /home/$USER/seti /home/$USER/seti-$NODE
cd /home/$USER/seti-$NODE
./seti
cd ~
rm -rf /home/$USER/seti-$NODE
exit
```

Software::Scheduler::SGE::cpu farm

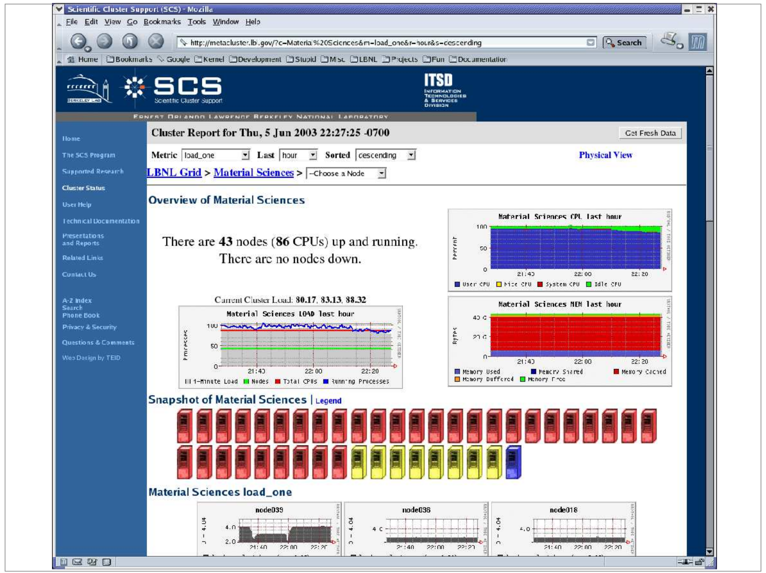
- User submits a script to the scheduler
- Scheduler, schedules the job to a node
- Node runs the job
- Scheduler returns the info back to the user
- Very straight forward

Software::Scheduler::SGE::LAM-MPI

- User submits a script the scheduler and specified a particular parallel environment to run under
- Scheduler determines what nodes (or slots) can be scheduled
- The first slot in the round robin sort is the Master
- SGE runs a script in the master slot that sets up LAM (calls lamboot with the proper machine file)
- Nodes defined in the machine file are assumed to be busy while the mater process is busy

Software::Monitoring::Ganglia

- Developed at UC Berkeley
- Uses a shared multicast channel for node communication
- All nodes keep a redundant copy of the stats
- Scales well (+2000 nodes)
- Low network overhead (transmits when needed)
- Nodes communicate with XML
- Our clusters can be seen using a web interface



Scientific Cluster Support (SCS) - Mozilla

http://metacluster.lbl.gov/?c=Material%20Sciences&m=load_one&r=hour&s=descending

Cluster Report for Thu, 5 Jun 2003 22:27:25 -0700

Metric load_one Last hour Sorted descending

LBNL Grid > Material Sciences > --Choose a Node

Overview of Material Sciences

There are **43** nodes (**86** CPUs) up and running.
There are no nodes down.

Current Cluster Load: **80.17**, **83.13**, **88.32**

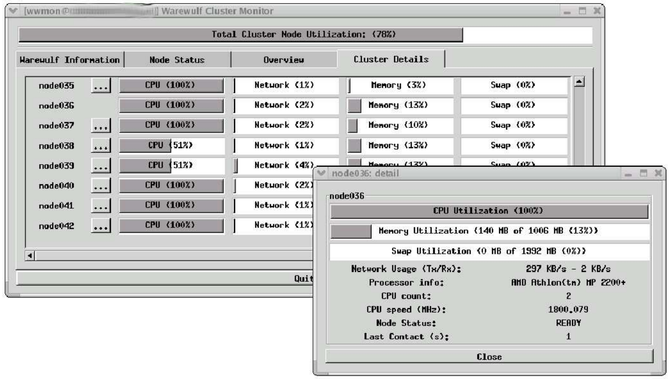
Snapshot of Material Sciences

Material Sciences load_one

Software::Monitoring::Warewulf

- Cluster distribution that includes some monitoring tools
- Realtime monitoring (within 1 second)
- Command line and GUI frontends
- Good for debugging bottlenecks
- Scales (theoretically) well (+~ 1000 nodes)
- Integrated well with the core cluster distribution

Software::Monitoring::wwmon



[wwmon@[redacted]] Warewulf Cluster Monitor

Total Cluster Node Utilization: (78%)

Warewulf Information | Node Status | Overview | Cluster Details

```
node035  ...  CPU (100%)  Network (1%)  Memory (3%)   Swap (0%)
node036       CPU (100%)  Network (2%)  Memory (13%)  Swap (0%)
node037  ...  CPU (100%)  Network (2%)  Memory (10%)  Swap (0%)
node038  ...  CPU (51%)   Network (1%)  Memory (13%)  Swap (0%)
node039  ...  CPU (51%)   Network (4%)  Memory (13%)  Swap (0%)
node040  ...  CPU (100%)  Network (2%)
node041  ...  CPU (100%)  Network (1%)
node042  ...  CPU (100%)  Network (1%)
```

Quit

node036: detail

node036

CPU Utilization (100%)

Memory Utilization (140 MB of 1006 MB (13%))

Swap Utilization (0 MB of 1992 MB (0%))

```
Network Usage (Tx/Rx):   297 KB/s - 2 KB/s
Processor info:          AMD Athlon(tm) MP 2200+
CPU count:               2
CPU speed (MHz):         1800.079
Node Status:             READY
Last Contact (s):        1
```

Close

Software::Distribution

- Warewulf Cluster Toolkit
  - One virtual slave node image on master
  - Nodes utilize etherboot booting from a floppy or eprom on the NIC
  - Includes configuration and monitoring tools
  - Intuitive to use for a knowledgeable admin
  - Scales so well there is no software administration difference if you have 1 or +100 nodes!

Software::Distribution

- Customizable
  - Extremely flexible virtual node filesystem image
  - Uses standard RedHat (or RPM based) distribution on master
  - Chroot'able node image with RPMdb intact
  - Pre-built addon's integrate easily into Warewulf
  - No weird kernel dependencies (except Linux-2.4)

Software::Distribution

- Uses in the community
  - Beowulf
  - HTTPD load balanced/fail over cluster
  - Parallel IDS
  - Render farm
  - Temporary lab clusters (no node filesystem writes)

Software::Distribution

- Other known installations
  - UC Berkeley (at least 3)
  - Kennedy Space and Science Center
  - Texas University (at least 2)
  - Arizona State
  - Cognigen Corp (Biotech/Genomics)
  - UC Davis
  - University of Geneva
  - Florida University
  - Almost 7000 downloads!

Photos

Conclusion::References

(in order of appearance)

- www.beowulf.org
- scs.lbl.gov
- www.myri.com
- www.infinibandta.org - infiniband.sourceforge.net
- www-unix.mcs.anl.gov/mpi/mpich
- www.lam-mpi.org
- www.csm.ornl.gov/pvm/pvm_home.html
- gridengine.sunsource.net
- www.etherboot.org
- ganglia.sourceforge.net
- www.warewulf-cluster.org