User Session 1
Introduction to Clusters

Rocks-A-Palooza III
Starting at 10:00am
Outline of the Day

Session 1
- Introduction to Clusters
- High level definition of Rocks
- Some other projects for perspective
- “Tuner Tale”

Session 2
- More complete definition of Rocks
- Software Components
- Description based installation
Session 3
  - Definition of Rolls
  - Cluster build demonstration

Session 4
  - Open Lab
  - Remote access to cluster at UCSD
User Track: Goals

- Training for users and technical managers in Rocks
- Build on the Rocks community and introduce people face-to-face
- Entry into the Rocks-A-Palooza Tracks
  - Year 1: User Track
  - Year 2: Developer Track
  - Year 3: Working Groups
Ground Rules

◆ We are going to go slow
  ➢ Starting with “what is a cluster”
  ➢ Ending with building a Rocks cluster

◆ This is for new users
  ➢ Slides are recycled from RAP I, RAP II
  ➢ If you are bored go to the developer track

◆ Interrupt me at ANY time
  ➢ This is for you and should be interactive
  ➢ I’d also rather interact than present slides
Before We Start

- Who are you?
  - Name
  - Title (optional)
  - Institution
- Why are you where?
- Are you running Rocks now?
Let’s Start
Sampling of HPC Hardware

- Cray Y-MP 8 Vectors 2.5 GF
- CM-5
- Cray Y-MP 1024 CPUs
- IBM SP1 700GF
- Hitachi CP-PACs 700GF
- KSR
- ASCI Red 1 TF
- ASCI Blue 3 TF
- ASCI White 12 TF
- IBM SP3 1 TF
- NCSA Platinum 1 TF (x86)
- TeraGrid 13 TF (IA64)
- Earth Simulator 40 TF

© 2007 UC Regents
Some Significant Software
Relationships
NOW
Network of Workstations

- Pioneered the vision for clusters of commodity processors.
  - David Culler (UC Berkeley) started early 90’s
  - SunOS on SPARC Microprocessor
  - High Performance, Low Latency Interconnect
    - First generation of Myrinet
    - Active Messages
  - Glunix (Global Unix) execution environment

- Brought key issues to the forefront of commodity-based computing
  - Global OS
  - Parallel file systems
  - Fault tolerance
  - High-performance messaging
  - System Management
Beowulf
www.beowulf.org

Definition
- Collection of commodity computers (PCs)
- Using a commodity network (Ethernet)
- Running open-source operating system (Linux)

Interconnect
- Gigabit Ethernet (commodity)
  - High Latency
  - Cheap
- Myrinet, Infiniband, … (non-commodity)
  - Low Latency
  - OS-bypass
  - Expensive
- Programming model is Message Passing

NOW pioneered the vision for clusters of commodity processors.
Beowulf popularized the notion and made it very affordable
Come to mean any Linux cluster
Outcomes of NOW / Beowulf

- Clusters of PCs Popularized
- Allowed more people to work on parallel computing
- Almost all software components published as open-source
- Brought key ingredients of MPPs into the commodity space
  - Message passing environments
  - Batch processing systems
- Extremely hard to build and run
High Performance Computing Cluster

Frontend Node

Private Ethernet Network

Public Ethernet

Application Network (Optional)

Node
Node
Node
Node
Node
Node
Node
Node
Node
Node
Node
Node

Power Distribution
(Net addressable units as option)

© 2007 UC Regents
Minimum Components

- Local Hard Drive
- Power
- Ethernet

- i386 (Pentium/Athlon)
- x86_64 (Opteron/EM64T)
- ia64 (Itanium) server
Optional Components

- High-performance network
  - Myrinet
  - Infiniband

- Network-addressable power distribution unit

- Keyboard/video/mouse network not required
  - Non-commodity
  - How do you manage your management network?
Growth of Clusters

Architectures / Systems

© 2007 UC Regents
Growth of Linux

Operating System / Systems

© 2007 UC Regents
Growth of Commodity CPUs
x86_64, EM64T, IA-64, IA-32
Growth of Commodity Networks
Infiniband, Gigabit, Myrinet
Top500: Linpack Performance
Observations

- Clusters Dominate
  - Slowly growing since late 90’s
  - Now at 72% of deployed Top500 machines

- Growth of Aggregate Top500 performance remains constant
  - Even though clusters can be less efficient than other architectures
  - If cost is low enough efficiency is not the most important metric
key point

If you are fast you can be stupid
Other Clusters

- **Highly Available (HA)**
  - Generally small, less than 8 nodes
  - Redundant components
  - Multiple communication paths
  - This is not Rocks

- **Visualization Clusters**
  - Each node drives a display
  - OpenGL machines
  - This is not core Rocks
  - But, there is a Viz Roll
The Dark Side of Clusters

- Clusters are phenomenal price/performance computational engines…
  - Can be hard to manage without experience
  - High-performance I/O is still unsolved
  - Finding out where something has failed increases at least linearly as cluster size increases
- Not cost-effective if every cluster “burns” a person just for care and feeding
- Programming environment could be vastly improved
- Technology is changing very rapidly. Scaling up is becoming commonplace (128-256 nodes)
The Top 2 Most Critical Problems

- The largest problem in clusters is *software skew*
  - When software configuration on some nodes is different than on others
  - Small differences (minor version numbers on libraries) can cripple a parallel program

- The second most important problem is adequate job control of the parallel process
  - Signal propagation
  - Cleanup
Rocks (open source clustering distribution)
www.rocksclusters.org

- Technology transfer of commodity clustering to application scientists (non-technical people)
  - “make clusters easy”
  - Scientists can build their own supercomputers and migrate up to national centers, or international grids, as needed
  - Supports more than just MPI machines

- Rocks is a cluster on set of CDs (or a DVD)
  - Red Enterprise Hat Linux (open source, de facto standard, and free)
  - Clustering software (PBS, SGE, Ganglia, GT4, …)
  - Highly programmatic software configuration management

- Core software technology for many UCSD projects
  - BIRN, CTBP, EOL, GEON, NBCR, OptIPuter, CAMERA, …

- First Software release Nov, 2000
  - Began as an MPI cluster solution
  - Now builds grid resources
  - Moving towards virtualization (XEN) and other OSes (Solaris)

- Supports x86, Opteron/EM64T, and Itanium
Philosophy

- Caring and feeding for a system is not fun
- System Administrators cost more than clusters
  - 1 TFLOP cluster is less than $100,000 (US)
  - Close to actual cost of a fulltime administrator
- The system administrator is the weakest link in the cluster
  - Bad ones like to tinker (make small changes)
  - Good ones still make mistakes
Philosophy continued

- All nodes are 100% automatically configured
  - Zero “hand” configuration
  - This includes site-specific configuration
- Run on heterogeneous standard high volume components
  - Use components that offer the best price/performance
  - Software installation and configuration must support different hardware
  - Homogeneous clusters do not exist
  - Disk imaging requires homogeneous cluster
Philosophy
continued

◆ Optimize for installation
  ➢ Get the system up quickly
  ➢ In a consistent state
  ➢ Build supercomputers in hours not months

◆ Manage through re-installation
  ➢ Can re-install 128 nodes in under 20 minutes
  ➢ No support for on-the-fly system patching

◆ Do not spend time trying to issue system consistency
  ➢ Just re-install
  ➢ Can be batch driven

◆ Uptime in HPC is a myth
  ➢ Supercomputing sites have monthly downtime
  ➢ HPC is not HA
Q: Contributions to user docs
A: https://wiki.rocksclusters.org
Other Cluster Toolkits

related work
OpenMosix

- Overview
  - Single system image - all nodes look like one large multiprocessor
  - Jobs migrate from machine to machine (based on machine load)
  - No changes required for apps to use system
- Interconnects supported
  - All IP-based networks
- Custom Linux Kernel
  - Download a new kernel
  - Or patch and compile
  - Install kernel on all nodes
- Supports
  - Diskfull
  - Diskless
Warewulf

◆ Overview
  ✅ Install frontend first
    • Recommend using RPM-based distribution
  ✅ Imaged based installation
    • “Virtual node filesystem”
  ✅ Attacks problem of generic slave node management

◆ Standard cluster software not included
  ✅ Added separately
  ✅ Use ‘chroot’ commands to add in extra software

◆ Supports
  ✅ Diskfull
  ✅ Diskless
Scyld Beowulf

- Single System Image
  - Global process ID
  - Not a global file system
- Heavy OS modifications to support BProc
  - Patches kernel
  - Patches libraries (libc)
- Job start on the frontend and are pushed to compute nodes
  - Hooks remain on the frontend
  - Does this scale to 1000 nodes?
- Easy to install
  - Full distribution
  - Often compared to Rocks
SCore

- Research group started in 1992, and based in Tokyo.
- Score software
  - Semi-automated node integration using RedHat
  - Job launcher similar to UCB’s REXEC
  - MPC++, multi-threaded C++ using templates
  - PM, wire protocol for Myrinet
- Development has started on SCore Roll
Scalable Cluster Environment (SCE)

- Developed at Kasetsart University in Thailand
- SCE is a software suite that includes:
  - Tools to install, manage, and monitor compute nodes
    - Diskless (SSI)
    - Diskfull (RedHat)
  - A batch scheduler to address the difficulties in deploying and maintaining clusters
  - Monitoring tools (SCMSWeb)
- User installs frontend with RedHat and adds SCE packages.
- Rocks and SCE are working together
  - Rocks is good at low level cluster software
  - SCE is good at high level cluster software
  - SCE Roll is now available for Rocks
  - ThaiGrid is SCE + Rocks
Open Cluster Group
(OSCAR)

- OSCAR is a collection of clustering best practices (software packages)
  - PBS/Maui
  - OpenSSH
  - LAM/MPI
- Image based installation
  - Install frontend machine manually
  - Add OSCAR packages to frontend
  - Construct a “golden image” for compute nodes
  - Install with system imager
  - “Multi-OS” – Mainly RPM-based distributions (aka Red Hat)
- Started as a consortium of industry and government labs
  - NCSA, ORNL, Intel, IBM, Dell, others
  - Dell now does Rocks.
  - NCSA and IBM are no longer contributors.
System Imager

- Originally VA/Linux (used to sell clusters) (now “bald guy software)
- System imaging installation tools
  - Manages the files on a compute node
  - Better than managing the disk blocks
- Use
  - Install a system manually
  - Appoint the node as the golden master
  - Clone the “golden master” onto other nodes
- Problems
  - Doesn’t support heterogeneous
  - Not method for managing the software on the “golden master”
  - Need “Magic Hands” of cluster-expert admin for every new hardware build
Cfengine

- Policy-based configuration management tool for UNIX or NT hosts
  - Flat ASCII (looks like a Makefile)
  - Supports macros and conditionals
- Popular to manage desktops
  - Patching services
  - Verifying the files on the OS
  - Auditing user changes to the OS
- Nodes pull their Cfengine file and run every night
  - System changes on the fly
  - One bad change kills everyone (in the middle of the night)
- Can help you make changes to a running cluster
Kickstart

- RedHat
  - Automates installation
  - Used to install desktops
  - Foundation of Rocks

- Description based installation
  - Flat ASCII file
  - No conditionals or macros
  - Set of packages and shell scripts that run to install a node
LCFG

- Edinburgh University
  - Anderson and Scobie
- Description based installation
  - Flat ASCII file
  - Conditionals, macros, and statements
    - Full blown (proprietary) language to describe a node
- Compose description file out of components
  - Using file inclusion
  - Not a graph as in Rocks
- Do not use kickstart
  - Must replicate the work of RedHat
- Very interesting group
  - Design goals very close to Rocks
  - Implementation is also similar
Rocks Basic Approach

- Install a frontend
  1. Insert Rocks Base CD
  2. Insert Roll CDs (optional components)
  3. Answer a few screens of configuration data
  4. Drink coffee/tea/beer (takes about 30 minutes to install)

- Install compute nodes:
  1. Login to frontend
  2. Execute insert-ethers
  3. Boot compute node with Rocks Base CD (or PXE)
  4. Insert-ethers discovers nodes
  5. Goto step 3

- Add user accounts
- Start computing

Optional Rolls
- Condor
- Grid (GT4)
- Java
- SCE (developed in Thailand)
- Sun Grid Engine
- PBS (developed in Norway)
- Area51 (security monitoring tools)
- Many Others …
Minimum Requirements

- **Frontend**
  - 2 Ethernet Ports
  - CDROM
  - 18 GB Disk Drive
  - 512 MB RAM

- **Compute Nodes**
  - 1 Ethernet Port
  - 18 GB Disk Drive
  - 512 MB RAM

- Complete OS Installation on all Nodes
- No support for Diskless (yet)
- Not a Single System Image
- All Hardware must be supported by RHEL
key point

The frontend machine of the cluster requires two Ethernet ports.
HPCwire Reader’s Choice Awards for 2004/2005

- Rocks won in Several categories:
  - Most Important Software Innovation (Reader’s Choice)
  - Most Important Software Innovation (Editor’s Choice)
  - Most Innovative - Software (Reader’s Choice)
Commercial Interest

Makes Beowulf Clusters child’s play!

Scalable Rocks Web Console

- Portable cluster management
- Enhanced performance
- Reduced management time and costs

With Platform Rocks, you can:
- Rapidly deploy new Linux-based computing infrastructure
- Achieve a lower total cost of ownership (TCO)
- Reduce the hassle and business risks associated with deploying and managing Linux clusters

<table>
<thead>
<tr>
<th>Processor</th>
<th>Type of NIC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PC/XTD (Level 3P)</td>
</tr>
<tr>
<td></td>
<td>PCiX/P (Level 3P)</td>
</tr>
<tr>
<td></td>
<td>PCiX/P (Level 33P)</td>
</tr>
<tr>
<td>MXnet Roll for i386</td>
<td>MX-2G 1.1.1 roll for i386</td>
</tr>
<tr>
<td>MXnet Roll for i64</td>
<td>MX-2G 1.1.1 roll for i64</td>
</tr>
<tr>
<td>MXnet Roll for x86_64</td>
<td>MX-2G 1.1.1 roll for x86_64</td>
</tr>
</tbody>
</table>

Note: Each MXnet roll contains MX-2G 1.1.1, MX-2G-32X 1.2.6.0, OpenMP 1.0, and EPL. Installation instructions are available on the Rocks homepage.

© 2007 UC Regents
# Registration Page

(conditional)

## Rocks Cluster Register

Back to [www.rockclusters.org](http://www.rockclusters.org)

Add your cluster to the Register.
Click on a column header to sort by that field.
Click on an id for details and to edit your cluster.

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Org</th>
<th>CPUType</th>
<th>CPUs</th>
<th>CPUClock (GHz)</th>
<th>FLOPS (GfLOPS)</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>876</td>
<td>Total CPUs, Ave CPUClock, Total FLOPS:</td>
<td></td>
<td>51610</td>
<td>2.20</td>
<td>249024.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>969</td>
<td>Jews</td>
<td>MHPCC</td>
<td>EM64T</td>
<td>1206</td>
<td>3.00</td>
<td>7776</td>
<td>Maui</td>
</tr>
<tr>
<td>497</td>
<td>Tungsten 2</td>
<td>NCSA</td>
<td>EM64T</td>
<td>1040</td>
<td>3.60</td>
<td>7488</td>
<td>Urbana, IL</td>
</tr>
<tr>
<td>51</td>
<td>GridKa</td>
<td>Forschungszentrum Karlsruhe</td>
<td>Pentium 4</td>
<td>1558</td>
<td>2.37</td>
<td>7384.92</td>
<td>Karlsruhe, Germany</td>
</tr>
<tr>
<td>571</td>
<td>EMGS</td>
<td>EMGS</td>
<td>Pentium 4</td>
<td>1060</td>
<td>3.40</td>
<td>7208</td>
<td>Trondheim, Norway</td>
</tr>
<tr>
<td>652</td>
<td>Athena-69</td>
<td>ACME</td>
<td>EM64T</td>
<td>969</td>
<td>3.40</td>
<td>6589.2</td>
<td>Brazil</td>
</tr>
<tr>
<td>130</td>
<td>Lonestar</td>
<td>TACC</td>
<td>Pentium 4</td>
<td>1024</td>
<td>3.06</td>
<td>6266.88</td>
<td>Austin, Texas</td>
</tr>
<tr>
<td>665</td>
<td>Tatanka</td>
<td>University Of Calgary Biological Computing</td>
<td>EM64T</td>
<td>624</td>
<td>3.40</td>
<td>4243.2</td>
<td>Calgary, Alberta Canada</td>
</tr>
<tr>
<td>299</td>
<td>USCMS Fermilab Tier1</td>
<td>Fermi National Accelerator Lab</td>
<td>Pentium 4</td>
<td>704</td>
<td>2.80</td>
<td>3042.4</td>
<td>Batavia, IL</td>
</tr>
<tr>
<td>65</td>
<td>Iceberg</td>
<td>Bio-X @ Stanford University</td>
<td>Pentium 4</td>
<td>604</td>
<td>2.80</td>
<td>3382.4</td>
<td>Stanford, CA</td>
</tr>
<tr>
<td>599</td>
<td>Sepeli (MGrid)</td>
<td>CSC - Scientific Computing Ltd.</td>
<td>Opteron</td>
<td>768</td>
<td>2.20</td>
<td>3379.2</td>
<td>Espoo, Finland</td>
</tr>
</tbody>
</table>
User Base

- > 1300 Users on the Discussion List
- 5 Continents
- **University**, Commercial, Hobbyist
High Performance Computing Community is eager to adopt open-source clustering solutions
Optimization?

Re-inventing the wheel does not advance science
A Tale of a Cluster Tuner
(288 AthlonMP Hand Built Machine)
07.2002: The Idea
08.2002 - 11.2002: Construction
12.2002: Build Complete & Celebration

- Machine only 50% operational
- But, they are getting results
- Machine is fully operational 3 months later
Summary

- 07.2002
  - Design system
- 08.2002 - 11.2002
  - Build system
- 03.2003
  - System in Production
- 7 months (maybe 8)
  - Concept to Cluster
  - Still just a Beowulf
  - Moore-cycle is 18 months
    - Half life for performance
    - Half life for cost
  - Useful life is 36-48 months
- What did they optimize for?
Rockstar Cluster

- 129 Sun Fire V60x servers
  - 1 Frontend Node
  - 128 Compute Nodes
- Gigabit Ethernet
  - $13,000 (US)
  - 9 24-port switches
  - 8 4-gigabit trunk uplinks
- Built live at SC’03
  - In under two hours
  - Running applications
- Top500 Ranking
  - 11.2003: 201
  - 06.2004: 433
  - 49% of peak
Rockstar Topology

- 24-port switches
- Not a symmetric network
  - Best case - 4:1 bisection bandwidth
  - Worst case - 8:1
  - Average - 5.3:1
Super Computing 2003 Demo

- We wanted to build a Top500 machine live at SC’03
  - From the ground up (hardware and software)
  - In under two hours
- Show that anyone can build a super computer with:
  - Rocks (and other toolkits)
  - Money
  - No army of system administrators required
- HPC Wire Interview
  - **HPCwire**: What was the most impressive thing you’ve seen at SC2003?
  - **Larry Smarr**: I think, without question, the most impressive thing I’ve seen was Phil Papadopoulos’ demo with Sun Microsystems.
Building Rockstar
Standard Rocks Installation

- Day 1 - Idea
- Day 30 - Production

- Not just us, worldwide user base has done the same
Example:
NCSA (National Center for Supercomputing Applications)

- Tungsten2
  - 520 Node Cluster
  - Dell Hardware
  - Topspin Infiniband
- Deployed 11.2004
- Easily in top 100 of the 06.2005 top500 list
- “We went from PO to crunching code in 2 weeks. It only took another 1 week to shake out some math library conflicts, and we have been in production ever since.” -- Greg Keller, NCSA (Dell On-site Support Engineer)

2nd Largest registered Rocks cluster

source: topspin (via google)
© 2007 UC Regents
NCSA
National Center for Supercomputing Applications

- Tungsten2
  - 520 Node Cluster
  - Dell Hardware
  - Topspin Infiniband
- Deployed 11.2004
- Easily in top 100 of the 06.2005 top500 list
- “We went from PO to crunching code in 2 weeks. It only took another 1 week to shake out some math library conflicts, and we have been in production ever since.” -- Greg Keller, NCSA (Dell On-site Support Engineer)

Largest registered Rocks cluster

source: topspin (via google)
© 2007 UC Regents
Lost Time = Lost Computation

- Assumption
  - **Rocks**
    - 256 2.2 GHz Pentium IV
    - 1,126 GFlops
    - Available at same time as tuner build
    - 1 month to build
  - **Tuner**
    - 144 - 264 Athlon-MP 2200+
    - 512 - 950 Gflops
    - 5 - 7 months to build
- Baseline of 50% CPU efficiency for Rocks
- Tuner improvement beyond baseline
  - 10% (55% efficiency)
  - 20% (60% efficiency)
  - 50% (75% efficiency)
- Tuner must have 50% gain to catch baseline after 1.5 years
Assumptions
- Two salaried tuners
- “Full burden” (salary, grant overhead, office space, etc) is $180k / year.

Invest
- 5 months salary into baseline
- $150k (5 months)
- Just buy more nodes
  - $2500 / node

Month 7
- Baseline cluster grows
- 54 2.2 GHz servers
- Ignoring Moore’s Law!

Baseline wins
Other Tuners

◆ Kernel Tuning
  ✐ “My handcrafted kernel is X times faster.”

◆ Distribution Tuning
  ✐ “Distribution Y is X times faster.”
  ✐ RFP: “Vendor will be penalized for a Red Hat only solution”
    • Typical of grant purchases (Request For Proposals)

◆ White-box Tuning
  ✐ “White-box vendor Y has a node that is X times cheaper.”
Conclusion

- Need to factor in the human cost for optimization
- With commodity hardware prices it is difficult to justify optimized or tuned machines
- This is not just a lesson for commodity clustering
key point

Spend money on hardware not people