Linux Clusters Institute: Getting the Most from Your Linux Cluster

Software Basics

A Cluster versus gaggle of linux nodes

**Cluster**
- Nodes dedicated to compute
- Shared home/work space
- Node interconnect
- Consistent software stack
- A set of resources to perform a common set of tasks

**Gaggle of linux nodes**
- Users able to log into every node
- Filesystems don’t need to be common
- No need for dedicated connection between nodes
- Software stack on node independent of other nodes
A Cluster versus gaggle of linux nodes

A fundamental piece of software for a cluster would be some sort of cluster management toolkit. Many admins abhor such software, especially if it has a GUI. However, to efficiently set up and manage a cluster, some sort of cluster management software is needed.

Why do you need such a set of software?
• Cost of cluster versus salary of administrator
• Stability of cluster depends on skills of administrator
• Even a good cluster administrator is still fallible

What makes setting up a cluster difficult

There are many tasks that a linux cluster requires. Some of these include:
• Boot server
• Batch system host
• Login (or head) nodes (versus “compute” nodes)
• Possible License server(s)
• Possible Database server(s)
• Storage host(s)
• Monitoring host
• NTP server
• DHCP server
• Local OS repository
• DNS server

Many of the above functions can share a host but there are some that it isn’t a really good idea to have share. Cluster management software automates what the cluster admin would normally have to do by hand.
Cluster pieces

Maybe, at this point, it would be good to explain the parts and functions of a cluster.

- **Login nodes** – these nodes are where the cluster users login. Here, the normal users submit jobs, copy/save data, check the status of jobs and even run small tests.
- **Compute nodes** – this is where the users main computation takes place, usually through running batch jobs. Users are mostly restricted from logging directly into these nodes. Some sites allow the user that owns the job to login to the node(s) that are running the job.
- **Management nodes** – nodes on which cluster management and control functions are based. Cluster users are generally restricted from these nodes.

Cluster pieces continued

- **Service nodes** – nodes which host infrastructure services for the cluster, such as:
  - shared storage
    - NFS
    - Lustre
    - GPFS
    - Others
  - image/boot servers
  - monitoring hosts
  - log hosts
  - various system services
    - NTP
    - DHCP
    - OS/patch repository
    - DNS
    - Etc
  - network(s) – one or more networks which make the disparate pieces into a whole
Cluster networking

While this portion of the workshop, “software basics” has little to do with the cluster’s network, how the network is set up has a lot to do with how the software is set up.

How many networks do you have? How many networks should you have? The possible functions your network may have to facilitate are:

- internal communications - general node to node communications
- external – network open to the outside
- management – node and cluster management functions
- interconnect – high speed, low latency for multi-node jobs

Some or all of these can share a medium, but there are advantages and drawbacks to doing so and all factors should be considered when working with this part of the cluster design.

Common problems

- Nodes get out of sync
  - Running different versions of software or libraries on different nodes can cause some serious problems with running multi-node jobs
  - Having the password files out of sync also causes problems
  - Nodes whose date/time setting varies causes authentication issues
- Shared filesystem problems
- Network problems
- Node hardware problems
- Firmware issues
  - Node
  - Chassis (if you have them)
  - Network, both NIC and switches
- Problems introduced on nodes from previous jobs
Homogeneous versus heterogeneous nodes

Most of what you see here are assuming a homogeneous set of nodes which is how the majority of clusters tend to be set up. However, if your cluster has users from different disciplines, it may be necessary to set up a cluster with different sets of nodes with different capabilities.

Most cluster management tools are built with this in mind, but having heterogeneous nodes can make node synchronization a little more complicated.

Functions of cluster management software

What should a good cluster management package offer?

• A way to automate the building of a cluster
• Some way to easily maintain cluster system consistency
• The ability to automate cluster maintenance tasks
• Offer some way to monitor cluster health and performance
Some Cluster Management software suites

- xCAT (Extreme Cluster/Cloud Administration Toolkit) ([http://sourceforge.net/p/xcat/wiki/Main_Page/](http://sourceforge.net/p/xcat/wiki/Main_Page/))
- Warewulf ([http://warewulf.lbl.gov/trac](http://warewulf.lbl.gov/trac))
- Perceus ([http://perceus.org/](http://perceus.org/))
- Bright Cluster Manager ([http://www.brightcomputing.com/Bright-Cluster-Manager](http://www.brightcomputing.com/Bright-Cluster-Manager))

There are many others. Feel free to choose the one that suits your needs.

Basics of cluster setup

There are some decisions to make as to how the cluster should be setup. One of those is how the OS is loaded to the nodes.

Do you:
- Boot the node from a centrally managed image?
  - Load to hard drive versus load to memory
  - Stateful versus stateless versus stateless “lite”
    - *What are the advantages and disadvantages?*
- Load the OS on each node
  - *What are some advantages and disadvantages with this approach?*
Node synchronization

While it is true that booting with a central boot server can make it
easier to make sure the OS on each compute node (or, at least, each
type of compute node) has an identical setup/install, there are still files
which wind up being more dynamic. Some such files are
password/group/shadow and hosts files. Depending on your
installation, there will undoubtedly be others.

Some tools to consider:
• rsync
• CFEngine
• others

Software installation and management

All linux distros have some sort of package management tool. For
Redhat/CentOS/Scientific Linux based clusters, this is rpm and yum. Debian has
things like dpkg and apt.

In any case pre-packaged software tends to assume that it is going to be installed
in a specific place on the machine and that it will be the only version of that
software on that machine. There are exceptions, of course.

On a cluster, it may be necessary to look at software installation differently from a
standard linux desktop/server.
• Why might you want to install software in a non-standard location?
  • Install to global filesystem
  • Keep boot image as small as possible
• Why might you need to support several different versions of software?
  • Upgrading software needs testing
  • Users might have version requirements for their code
Software installation and management, cont.

Installing software to a central location might make sense, but there are problems. First off, software packages tend not to be relocatable. Secondly, maintaining different versions of the same software can be tricky.

Changing the installation location of software either requires that the software supports such a change or that you get the source and compile it to work from a different location.

Maintaining different versions of the same software to be available to users can be tricky. Various environment variables (PATH, MANPATH, LD_LIBRARY_PATH, etc..) can get difficult. There are, however, tools to make that easier. Two such tools are:

- modules (http://modules.sourceforge.net/)

How is the cluster used?

Generally, such clusters are a batch environment and users don’t tend to log directly into nodes that their computation is running on. Running jobs in a batch mode means that you can define how and weather users will share a node’s resources and affect the operation of one another.

Batch systems tend to be made of a resource manager and a scheduler. The resource manager knows the state of the various resources on the cluster and maintains a list of the jobs that are requesting resources. The scheduler, using the information from the resource manager determines if there are enough resources to run any specific job and will run the jobs in the list in the order that it was configured to.
Resource managers

While both pieces are generally available from the same vendor, it is also possible to do a limited amount of mix and match such that if a resource manager is dictated by the hardware, you may not be tied to the bundled scheduler. Also, if the particular resource manager isn’t quite compatible with what you wish to do, you can choose one that is. Some resource managers include:

- SGE (Sun Grid Engine)
- LSF (Load Sharing Facility)
  - openlava
- PBS (Portable Batch System)
  - OpenPBS
  - Torque
  - PBS Professional

Job scheduler

While the resource managers previously mentioned have their own job scheduler (simplistic or not), Adaptive Computing has created a job scheduler which is designed to be compatible with many different resource managers and which can take the place of the scheduler bundled with the resource manager.

maui is the free version of that software and moab is the commercial version.

This software will also be covered more in-depth later in this workshop
Authentication and authorization

Authentication is the process of making sure that the user is who they are claiming to be. Someone knowing the password associated with a user in the list of approved users (/etc/passwd) was sufficient.

Authorization is determining what the authenticated user is authorized to do on the cluster. The use of /etc/group and what groups a user belongs to can determine the access to various resources, including files, applications, and even parts of the resource manager and job scheduler.

Authentication/Authorization on a cluster

Having local files (passwd & group) control this tends not to be a good practice anymore, especially for a cluster. On a cluster, keeping the files in sync across nodes is problematic.

• Synchronizing/transferring/updating files can have a measurable effect on the performance of a job running on a node

• Sometimes the process breaks down and files go out of sync

Running authorization and authentication against a central database such as LDAP or NIS means that files don’t have to be transferred and that nodes don’t go out of sync.

You do, however, introduce dependencies on the database server and on the network path between the cluster and that server.
Logging on a cluster

There are a couple of different aspects of logging here to talk about. First, there are the various system and application logs and then there are system change logs. While there are different packages that handle system and application logging, we tend to prefer rsyslog because it tends to be:

- Robust
  - Implements a more reliable protocol to prevent message loss
  - Has settings for local spooling if forwarding connection cannot be made
  - Able handle large volumes of message traffic
- Extensible
  - Output to multiple sources
    - Complex forwarding mechanism
    - Can output to various databases
  - Plugin interface to allow local customizations

The importance of cluster change logs

When working with a cluster, it is important to make a policy to log all hardware and software changes on a cluster. Logging software changes (including configuration file) can help:

- pinpoint software and/or software configuration changes which may have caused or contributed to a some application failure
- admin (or admins) remember changes/fixes they may have made in the past and how to do them again
- coordinate the activities of multiple admins on a system and reduce any occurrence of one admin stepping on the work of another

Logging hardware changes can:

- detect environmental problems with the cluster
- detect problematic cluster failure trends
Linux Clusters Institute: Getting the Most from Your Linux Cluster
Cluster Security And Best Practices

Agenda
• Security Overview
• Security Domains
• Security Threats
• Security Approaches
• Best Practices for Securing your cluster
Security Overview

What is Cluster Security?
- “Freedom from risk or danger”
  - Acceptable level of risk
- “Control of my own resources”
- “Uninterrupted work on my own resources”
- Many other definitions and ideas

“The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards.”
- Gene Spafford

Even attempting to secure a cluster results in:
- Much more work for the admin
- Limits access and use for valid usages
- Requires specific requests to add new services
- Less efficient use of resources due to security overhead
- May require more hardware for monitoring and closed networks
Security Overview

Can you run an “open” system?

• Open system:
  • Let anyone register as a user
  • Let anyone install software and run jobs
• Today, open systems are called “honeypots”
  • Great for the FBI, NSA, etc
• Or cloud computing
  • Hey, someone has to pay for those CPUs and power
  • You’re probably not getting paid to do this

Why is it necessary?

• One security incident can deprive you of the use of the entire cluster
• Can take days and many FTE’s to identify the issue
• Even after addressing the issue, a full review or restore of all data/executables is suggested
• If your cluster is compromised, it’s a springboard for any users that have accounts on other systems
• Any previous work done on the system may be lost
• In short, yes. It’s less work, money and FTE’s in the long run
Security Overview

Pet Peeve

• A Denial of Service (DOS) is a security issue
• Doesn’t matter if the cause is internal or external
  • Including the admin!
• Don’t do things that cause you to lose access to your own cluster
• If you do, treat that with the same priority that you do for a security incident

Security Overview

Security is not a checklist of things to do but a sampling of all the things to consider – Dan LaPine
Security Domains

We can identify specific Domains to protect:
• Physical
• System
• Network
• Data
These are the areas we need to protect on our Cluster

“Securing a computer system has traditionally been a battle of wits: the penetrator tries to find the holes, and the designer tries to close them..”

- Gosser

What is Physical Security?
• Protecting physical assets
• Controlling physical access to them

We can consider Physical security to cover the machines and network equipment, as well as methods to protect access to them
• For Unix, having physical access to a server is the same as having root access to a server, eventually.
Security Domains

What is System Security?

- Maintaining control of the OS and underlying hardware
- Restricting access to authorized persons for authorized operations

We can consider System security to cover the software installed on a cluster, and the configuration/policies that control access to it.

- If you oversecure the system and Deny Service, it can be considered a security issue

What is Network Security?

- Maintaining control of the network
  - This is particularly important for a cluster
  - Restricting remote access to authorized persons

We can consider Network security to cover the software/configuration used to communicate on the network, on every computer using the network, and the policies that control access to it.

- For a cluster, a network outage equals no cluster
Security Domains

What is Data Security?

• Maintaining control of the Data
  • For clusters, this implies a shared filesystem
  • Restricting access to authorized persons

We can consider Data security to cover the datasets that have been collected and stored, the shared filesystems and the policies that control access to them

In order to achieve a useful measure of Computer security we must provide coverage for all the security domains

“Remember, security is only as strong as the weakest link.

- Bruce Schneier
Security Threats

In order to secure a computer system or cluster, you need to understand the threat.

• Who
• Why
• How
• What
• Threat awareness

“If you know the enemy and you know yourself, you need not fear the result of a hundred battles.”

- Sun Tzu

Who are the security threats?

• External
  • People who are not targeting your resources
    • Botnets, IP scrapers, etc
  • People who are targeting your resources
    • They are specifically looking for ways into your system
    • Motives vary
  • Security measures vary based on the type of attacker
Security Threats

Who are the security threats?

- Internal
  - Anyone with access and authority by design
  - Three classes
    - Administrators
    - Users
    - Compromised user accounts

Why do they do want access?

- Intentional
  - Access to resources == $$
    - E.G. use your resources to generate electronic currency
    - Use your resources to hide their identity
    - Investigation/“Fun”

- Unintentional
  - User gets extra authority
    - Vulnerability
    - Misconfiguration
Security Threats

How do they get access?
• Exploit a vulnerability
  • Published exploit lists
  • Brute force attacks
  • Trial and error
• Exploit a social contact
  • Information gained through a social contact
• Exploit a compromised account on other system
• Exploit a misconfiguration
  • Lack of information/planning when deploying a service

Security Threats

What does an attacker see?
• Login nodes
• Public IP addresses
• Services
• System information pages
• Network activity
• Shell history with connection attempts
• MOTD announcements
Security Threats

Your Awareness of the current Threat:
- General Security Awareness
- User reports
- Security announcements/alerts
- Analysis of collected data
- Comparison of data with Peers
- Network analysis/Firewall logs

If you know what the threat is, you can prioritize your efforts to guard against that threat, and increase the odds that your system stays secure.
Security Approaches

There are different approaches to security

1. Assume that the cluster can be completely protected if everything is done correctly
2. Assume that the cluster will be hacked, and focus on early detection, recovery, and fixing the issues
3. Assume that the cluster must not be hacked and follow Gene Spafford’s advice or similar from slide 1
   • These are not binary- but they are the basis for most security policies

Security Approaches

• You may not get to pick which approach should be used
  • Might be the campus, the IT department or your boss’s call
• Each approach has advantages and disadvantages
  • You need understand these and adjust your practices as needed
• The emphasis on which security functions to employ will vary based on the approach
Security Approaches

• I can note that both NCSA as an Institution and XSEDE as an organization make the assumption that clusters can and will be hacked to some extent, and put much effort into detection, recovery and limited the damage that such attacks cause.
• I can also note that in my time in the USAF, our systems were not allowed to be hacked and all the effort was directed at that aim.
• I would not recommend the first approach.

Best Practices for Cluster Security

In this section we’ll cover some of the general best practices for securing a cluster. These need to be modified based on your cluster, the threat and your security policy.

I’ll provide the general principles by function and give the specific practices following.
Best Practices

Here is a quick review of the general functions to secure a cluster.

- Risk Avoidance
- Deterrence
- Prevention
- Detection
- Recovery

Note that your security approach determines the priority of the above functions.

---

Best Practices

**Risk Avoidance Principles:**

- Don't provide services that aren't necessary for your goal
- Don't allow access to everyone, all the time
- Keep your systems as simple as possible

**If you don't provide it, an attacker can not exploit it.**
Best Practices

**Risk Avoidance Practices:**

- Install the minimum number of packages necessary to meet the goals of your cluster
  - You can always install them later if needed
  - For different versions of Linux this may be offered in different fashions during install
  - Sometimes, you will need to select the minimum install and add individual packages
  - May need to remove items the OS selects by default
  - Different node types need different packages by usage

Best Practices

**Risk Avoidance Practices:**

- Define the specific services and service goals that are offered
  - Which services are available
  - Who has access to them
  - What type of service will we provide
  - What don't we provide
  - What is the process for change
  - Document
  - **Disable or remove the services that aren’t offered**
Best Practices

**Risk Avoidance Practices:**

- **Limit user access to nodes**
  - Only for the duration of a job on compute nodes
    - Change the user shell on the node
    - Add/remove user in access.conf
  - These are done during job startup/shutdown
    - May need to be checked after a job failure
  - Not at all on service or management nodes
    - Setting the user shell to /bin/false removes login, but allows access for gridftp, Globus online, etc

Best Practices

**Risk Avoidance Practices:**

- **Limit node access to management node**
  - Should be one way from mgmt node to other nodes
    - Restricted mgmt host based access via ssh
    - Need to set this for all networks
  - Can even require different auth methods for mgmt node access
    - OTP systems
    - Mgmt node access only via bastion hosts
Best Practices

Risk Avoidance Practices:

• Limit user access **on nodes**
  • Time limit sessions on login nodes
    • Set and export TMOUT to add idle limits
    • `pam_time` can be used to set authorized hours of access for a set of users
  • Set user limits for max ram, swap and processes
    • `ulimit -a` # shows the limits
    • `limits.conf` # set the limits
    • Don’t let the user DOS the node

Risk Avoidance Practices:

• Keep the nodes synced
  • Do not want to DOS ourselves
  • Have all of them set to the same time
    • `ntp` service
    • Cron script
  • Have them use the same user account info
    • `rsync`, shared filesystem
  • Have them all use the same DNS methods
    • `/etc/hosts, /etc/resolv.conf`
Best Practices

**Risk Avoidance Practices:**

- Limit the number of OS images available
  - If you offer multiple OS images
  - Each extra one is that much more work for the admin
  - Changing Distros is even more work
  - Older versions may not get needed security updates

Best Practices

**Deterrence Principles:**

- Let your users know what acceptable use is
- Impress on others that your systems are monitored and hard to hack

*If you tell people what the rules are, they may just follow them.*
Best Practices

**Deterrence Practices**

Active measures:
- Set login messages stating the acceptable access policy (/etc/motd)
  - Sometimes a simple warning is enough to deter hackers
- Publish Acceptable use policies
  - Deter users from “exploring” by telling them upfront what you expect of them

Passive measures: “Hide” the system
- Set the firewall to block or blackhole known contact attempts
- Look for and block unknown, systematic attacks
  - Network Intrusion Detect Systems (IDS) can do this automatically
  - Snort, Suricata, sshguard for automated IDS
Best Practices

**Deterrence Practices**

- These measures reduce the “visibility” of the cluster
  - Make it less attractive to gain access
  - Reduce the Internet footprint of a cluster
- IDS systems can also be considered as a preventative or detection measure

**Prevention Principles:**

- Fix known issues (Patching)
- Correctly configure services to meet goals
- Restrict User Access and Authority
- Document actions and changes via a log

If you fix it beforehand, an attacker can not exploit it.
Best Practices

**Prevention Practices: Patching**

- On a server, we just run `yum/zypper update`
- On a cluster, it’s not so simple
  - You might have a master image to patch
  - You may have to patch 100 installed static nodes
  - You may several OS’s to patch
  - You may have your own repository
  - You will have installed apps that depend on certain versions, and may need to rebuild them

---

Best Practices

**Prevention Practices: Patching**

1. Set a policy and let the users know that work might be interrupted by required updates
2. Plan to have test nodes to try out new changes
   1. We see the most annoying issues with filesystems and version dependency
   2. Required Kernel updates affect many things
3. Be prepared to accept an in-place workaround if you can’t patch for some reason
Best Practices

Prevention Practices: Patching
4. Announce PM’s in advance if possible
5. Have a status page available and keep it updated
6. Be prepared to rollback to the last version if it doesn’t work
7. Need to document what is planned to be done and what actually was done

Prevention Practices: Service Configuration
• Cluster services need to deal with multiple networks, and which nodes need what
• If the service is public facing, put it on a separate node
• If the service is for cluster users only, limit it’s access to only cluster users
Best Practices

Prevention Practices: Restrict User Access

• Create separate user spaces on the various shared and per node filesystems
• Restrict access to cluster-provides services on as-needed basis (e.g. MySQL service)
• Allow access to compute nodes only during job runs for non-dedicated nodes
  • Set a user reservation on a nodeset and let the scheduler give access for long duration work

• Use a more secure authentication method
  • Two factor authentication- RSA or VIP
  • One-time password system- DUO Security
  • Kerberos or AD passwordless methods
    • These time out, so a compromised machine is less of a problem
    • Allowing access via passwordless ssh keys alone is not recommended
Best Practices

**Prevention Practices: Document**
- Have a common system log
  - A simple text file with an rcs system will work
  - We use a home-grown perl-based one (RCSVI)
  - Make this readable only by the admins
  - Sync a copy of this off the system daily
- Require the admins to make entries that include why, not just what and when
- Can automate some entries into the log
- Should do logwatch emails to all the admins with daily summaries, at least

**Detection Principles:**
- Monitor your cluster, both manually and automatically
- Watch for feedback from the users
- Set alerts and automated blocking

**If you know about it, you can take action.**
Best Practices

**Detection Practices: Monitor**
- Automated Monitors—Ganglia and nagios
- Manual Monitoring
  - Inspection of captured /var/log/messages, dmesg
  - Occasional Realtime “watch the traffic flow” of your cluster monitor

---

**Detection Practices: User Feedback**
- Users will alert you if they feel something is wrong
  - Give them an easy means to do so
  - They may not be able to specifically identify the problem, but they can alert you that one exists
- Email mailing list for the admins
- Wiki with automated update emails to the admins
- Ticket systems—rt and jira
Best Practices

Detection Practices: Alerts
• Automated
  • Log digest and email alerts- logwatch
  • Generic Events- SimpleEventCorrelator (SEC)
  • Host Based IDS- OSSEC, samhain, OS tripwire
  • *Data Loss Prevention- OpenDLP

Best Practices

Recovery Principles:
• Need good, tested backups
• Need good documentation
• Be prepared to lose information depending on how often you archive
• May be able to remove intrusion
• Should plan for full restore in any case

If you have a security incident, you must be able to recover from it.
Best Practices

Recovery Practices: Backups
- Need bare-metal backup of the management node
  - Or a copy of the VM image it’s running in
  - Or really good documentation/revision control of all changes, ever
- Need to decide what data loss is acceptable
  - Filesystem design is key here

Recovery Practices: comprised user account
- Remove the account & home directory
- Remove files/directory that the user created
- Remove any files/directories that the user could have overwritten
- This can be time consuming and painful
- Limiting user access by design is key
- Keeping separate system and data backups can help
Best Practices

Recovery Practices: comprised root account

- Restore a known good backup
  - If the OS is on a virtual image, this can be trivial
  - If not, will need some bootable tool
    (SystemRescueCD, Clonezilla)
- Or Rebuild the system
  - You already documented this, right?
- Keeping separate system and data backups can help in either restore method
- As can good documentation

Security Tools

- Logwatch- https://help.ubuntu.com/community/Logwatch
- Snort- http://www.snort.org/
- Sshguard- http://www.sshguard.net/
- Ganglia- http://ganglia.sourceforge.net/
- SEC- http://simple-evcorr.sourceforge.net/
- OSSEC- http://www.ossec.net/
- Samhain- http://www.la-samhna.de/samhain/
Security Tools cont’d

- OpenDLP- [https://code.google.com/p/opendlp/](https://code.google.com/p/opendlp/)
- SystemRescueCD- [http://www.sysresccd.org/SystemRescueCd_Homepage](http://www.sysresccd.org/SystemRescueCd_Homepage)
- Clonezilla- [http://clonezilla.org/](http://clonezilla.org/)

References

- [http://www.stellarcore.net/logwatch/tabs/docs/HOWTO-Customize-LogWatch.html](http://www.stellarcore.net/logwatch/tabs/docs/HOWTO-Customize-LogWatch.html)
Wrap-up and Q & A

This was a lot of information and it may have been confusing.

• Do you have points that you wish us to clarify or elaborate on?
• Do you have other questions?
• Do you have other topics you’d like us to touch upon?