Linux Cluster HOWTO

Table of Contents

Linux Cluster HOWTO **1**
Ram Samudrala (me@ram.org) 1
1. Introduction 1
2. Hardware 1
3. Software 1
4. Set up, configuration, and maintenance 1
5. Performing tasks on the cluster 1
6. Acknowledgements 1
7. Bibliography 1
1. Introduction 2
2. Hardware 2
2.1 Node hardware 2
2.2 Server hardware 3
2.3 Desktop and terminal hardware 4
2.4 Miscellaneous/accessory hardware 4
2.5 Putting−it−all−together hardware 4
2.6 Costs 4
3. Software 4
3.1 Operating system: Linux, of course! 4
3.2 Networking software 5
3.3 Parallel processing software 5
3.4 Costs 5
4. Set up, configuration, and maintenance 5
4.1 Disk configuration 5
4.2 Package configuration 6
4.3 Operating system installation and maintenance 6
Personal cloning strategy 6
Cloning and maintenance packages 7
FAI 7
SystemImager 7
DHCP vs. hard−coded IP addresses 7
4.4 Known hardware issues 7
4.5 Known software issues 7
5. Performing tasks on the cluster 7
5.1 Rough benchmarks 8
5.2 Uptimes 8
6. Acknowledgements 8
7. Bibliography 8

Linux Cluster HOWTO

Ram Samudrala (me@ram.org)

v1.5, September 5, 2005

How to set up high−performance Linux computing clusters.

1. Introduction

2. Hardware

- 2.1 Node hardware
- 2.2 Server hardware
- 2.3 Desktop and terminal hardware
- 2.4 Miscellaneous/accessory hardware
- 2.5 Putting−it−all−together hardware
- 2.6 Costs

3. Software

- 3.1 Operating system: Linux, of course!
- 3.2 Networking software
- 3.3 Parallel processing software
- 3.4 Costs

4. Set up, configuration, and maintenance

- 4.1 Disk configuration
- 4.2 Package configuration
- 4.3 Operating system installation and maintenance
- 4.4 Known hardware issues
- 4.5 Known software issues

5. Performing tasks on the cluster

- 5.1 Rough benchmarks
- 5.2 Uptimes

6. Acknowledgements

7. Bibliography

1. Introduction

This document describes how we set up our Linux computing clusters for high−performance computing which we need for our research.

Use the information below at your own risk. I disclaim all responsibility for anything you may do after reading this HOWTO. The latest version of this HOWTO will always be available at http://www.ram.org/computing/linux/linux_cluster.html.

Unlike other documentation that talks about setting up clusters in a general way, this is a specific description of how our lab is setup and includes not only details the compute aspects, but also the desktop, laptop, and public server aspects. This is done mainly for local use, but I put it up on the web since I received several e−mail messages based on my newsgroup query requesting the same information. Even today, as I plan another 64−node cluster, I find that there is a dearth of information about exactly how to assemble components to form a node that works reliably under Linux that includes information not only about the compute nodes, but about hardware that needs to work well with the nodes for productive research to happen. The main use of this HOWTO as it stands is that it's a report on what kind of hardware works well with Linux and what kind of hardware doesn't.

---

2. Hardware

This section covers the hardware choices I've made. Unless noted in the known hardware issues section, assume that everything works *really* well.

Hardware installation is also fairly straight−forward unless otherwise noted, with most of the details covered by the manuals. For each section, the hardware is listed in the order of purchase (most recent is listed first).

2.1 Node hardware

32 machines have the following setup each:

- 2 XEON 2.66GHZ 533FSB CPUs
- Supermicro 6013A−T 1u case and motherboard
- 2 512MB PC2100 DDR REG ECC RAM
- 1 80GB SEA 7200 SATA HD
- 1 250GB SEA 7200 SATA HD

32 machines have the following setup each:

- 2 XEON 2.4GHZ 533FSB CPUs
- Supermicro X5DPR−1G2 motherboard
- 2 512MB PC2100 DDR REG ECC RAM
- 1 40GB SEA 7200 HD
- 1 120GB SEA 7200 HD
- Supermicro Slim 24X CDROM
- CSE−812 400 C/B 1U case

32 machines have the following setup each:

- 2 AMD Palamino MP XP 2000+ 1.67 GHz CPUs
- Asus A7M266−D w/LAN Dual DDR motherboard
- 2 Kingston 512mb PC2100 DDR−266MHz REG ECC RAM
- 1 41 GB Maxtor 7200rpm ATA100 HD
- 1 120 GB Maxtor 5400rpm ATA100 HD
- Asus CD−A520 52x CDROM
- 1.44mb floppy drive
- ATI Expert 2000 Rage 128 32mb
- IN−WIN P4 300ATX Mid Tower case
- Enermax P4−430ATX power supply

32 machines have the following setup each:

- 2 AMD Palamino MP XP 1800+ 1.53 GHz CPUs
- Tyan S2460 Dual Socket−A/MP motherboard
- Kingston 512mb PC2100 DDR−266MHz REG ECC RAM
- 1 20 GB Maxtor UDMA/100 7200rpm HD
- 1 120 GB Maxtor 5400rpm ATA100 HD
- Asus CD−A520 52x CDROM
- 1.44mb floppy drive
- ATI Expert 98 8mb AGP video card
- IN−WIN P4 300ATX Mid Tower case
- Intel PCI PRO−100 10/100Mbps network card
- Enermax P4−430ATX power supply

32 machines have the following setup each:

- 2 Pentium III 1 GHz Intel CPUs
- Supermicro 370 DLE Dual PIII−FCPGA motherboard
- 2 256 MB 168−pin PC133 Registered ECC Micron RAM
- 1 20 GB Maxtor ATA/66 5400 RPM HD
- 1 40 GB Maxtor UDMA/100 7200 RPM HD
- Asus CD−S500 50x CDROM
- 1.4 MB floppy drive
- ATI Expert 98 8 MB PCI video card
- IN−WIN P4 300ATX Mid Tower case

2.2 Server hardware

Two servers for external use (dissemination of information) with the following setups:

- 2 AMD Opteron 240 1.4 GHz CPUs
- RIOWORKS HDAMB DUAL OPTERON motherboard
- 4 KINGSTON 512MB PC3200 REG ECC RAM
- 80GB MAX 7200 UDMA 133 HD
- 6 200GB WD 7200 8MB HD
- ASUS 52X CD−A520 CDROM
- 1.44mb floppy drive
- Antec 4U22ATX550EPS 4u case

- 2 AMD Palamino MP XP 2000+ 1.67 GHz CPUs
- Asus A7M266−D w/LAN Dual DDR
- 4 Kingston 512mb PC2100 DDR−266MHz REG ECC RAM
- Asus CD−A520 52x CDROM
- 1 41 GB Maxtor 7200rpm ATA100 HD
- 6 120 GB Maxtor 5400rpm ATA100 HD
- 1.44mb floppy drive
- ATI Expert 2000 Rage 128 32mb
- IN−WIN P4 300ATX mid tower case
- Enermax P4−430ATX power supply

2.3 Desktop and terminal hardware

We have identified at least two kinds of users of our cluster: those that need (i.e., take advantage of) permanent local processing power and disk space in conjunction with the cluster to speed up processing, and those that just need only the cluster processing power. The former are assigned "desktops" which are essentially high−performance machines, and the latter are assigned dumb "terminals". Our desktops are usually dual or quad processor machines with the current high−end CPU being a 1.6 GHz Opteron, having as much as 10 GB of RAM, and over 1 TB of local disk space. Our terminals are essentially machines where a user can log in and then run jobs on our farm. In this setup, people may also use laptops as dumb terminals.

2.4 Miscellaneous/accessory hardware

We generally use/prefer Viewsonic monitors, Microsoft Intellimouse mice, and Microsoft Natural keyboards. These generally have worked quite reliably for us.

2.5 Putting−it−all−together hardware

For visual access to the nodes, we initially used to use KVM switches with a cheap monitor to connect up and "look" at all the machines. While this was a nice solution, it did not scale. We currently wheel a small monitor around and hook up cables as needed. What we need is a small hand held monitor that can plug into the back of the PC (operated with a stylus, like the Palm).

For networking, we generally use Netgear and Cisco switches.

2.6 Costs

Our vendor is Hard Drives Northwest (http://www.hdnw.com). For each compute node in our cluster (containing two processors), we paid about $1500−$2000, including taxes. Generally, our goal is to keep the cost of each processor to below $1000 (including housing it).

---

3. Software

3.1 Operating system: Linux, of course!

The following kernels and distributions are what are being used:

- Kernel 2.2.16−22, distribution KRUD 7.0
- Kernel 2.4.9−7, distribution KRUD 7.2
- Kernel 2.4.18−10, distribution KRUD 7.3
- Kernel 2.4.20−13.9, distribution KRUD 9.0
- Kernel 2.4.22−1.2188, distribution KRUD 2004−05

These distributions work very well for us since updates are sent to us on CD and there's no reliance on an external network connection for updates. They also seem "cleaner" than the regular Red Hat distributions, and the setup is extremely stable.

3.2 Networking software

We use Shorewall 1.3.14a ((http://www.shorewall.net) for the firewall.

3.3 Parallel processing software

We use our own software for parallelising applications but have experimented with PVM and MPI. In my view, the overhead for these pre−packaged programs is too high. I recommend writing application−specific code for the tasks you perform (that's one person's view).

3.4 Costs

Linux and most software that run on Linux are freely copiable.

---

4. Set up, configuration, and maintenance

4.1 Disk configuration

This section describes disk partitioning strategies. Our goal is to keep the virtual structures of the machines organised such that they are all logical. We're finding that the physical mappings to the logical structures are not sustainable as hardware and software (operating system) change. Currently, our strategy is as follows:

```
farm/cluster machines:

partition 1 on system disk     - swap  (2 * RAM)
partition 2 on system disk     - /     (remaining disk space)
partition 1 on additional disk - /maxa (total disk)

servers:

partition 1 on system disk        - swap  (2 * RAM)
partition 2 on system disk        - /     (4-8 GB)
partition 3 on system disk        - /home (remaining disk space)
partition 1 on additional disk 1  - /maxa (total disk)
partition 1 on additional disk 2  - /maxb (total disk)
partition 1 on additional disk 3  - /maxc (total disk)
partition 1 on additional disk 4  - /maxd (total disk)
partition 1 on additional disk 5  - /maxe (total disk)
partition 1 on additional disk 6  - /maxf (total disk)
partition 1 on additional disk(s) - /maxg (total disk space)

desktops:
```

```
partition 1 on system disk        - swap   (2 * RAM)
partition 2 on system disk        - /      (4-8 GB)
partition 3 on system disk        - /spare (remaining disk space)
partition 1 on additional disk 1  - /maxa  (total disk)
partition 1 on additional disk(s) - /maxb  (total disk space)
```

Note that in the case of servers and desktops, maxg and maxb can be a single disk or a conglomeration of disks.

4.2 Package configuration

Install a minimal set of packages for the farm. Users are allowed to configure desktops as they wish, provided the virtual structure is kept the same described above is kept the same.

4.3 Operating system installation and maintenance

Personal cloning strategy

I believe in having a completely distributed system. This means each machine contains a copy of the operating system. Installing the OS on each machine manually is cumbersome. To optimise this process, what I do is first set up and install one machine exactly the way I want to. I then create a tar and gzipped file of the entire system and place it on a bootable CD−ROM which I then clone on each machine in my cluster.

The commands I use to create the tar file are as follows:

```
tar -czvlps --same-owner --atime-preserve -f /maxa/slash.tgz /
```

I use a script called `go` that takes a machine number as its argument and untars the `slash.tgz` file on the CD−ROM and replaces the hostname and IP address in the appropriate locations. A version of the `go` script and the input files for it can be accessed at: http://www.ram.org/computing/linux/linux/cluster/. This script will have to be edited based on your cluster design.

To make this work, I use Martin Purschke's Custom Rescue Disk (http://www.phenix.bnl.gov/~purschke/RescueCD/) to create a bootable CD image containing the .tgz file representing the cloned system, as well as the `go` script and other associated files. This is burned onto a CD−ROM.

There are several documents that describe how to create your own custom bootable CD, including the Linux Bootdisk HOWTO (http://www.linuxdoc.org/HOWTO/Bootdisk−HOWTO/), which also contains links to other pre−made boot/root disks.

Thus you have a system where all you have to do is insert a CDROM, turn on the machine, have a cup of coffee (or a can of coke) and come back to see a full clone. You then repeat this process for as many machines as you have. This procedure has worked extremely well for me and if you have someone else actually doing the work (of inserting and removing CD−ROMs) then it's ideal. In my system, I specify the IP address by specifying the number of the machine, but this could be completely automated through the use of DHCP.

Rob Fantini (rob@fantinibakery.com) has contributed modifications of the scripts above that he used for cloning a Mandrake 8.2 system accessible at http://www.ram.org/computing/linux/cluster/fantini_contribution.tgz.

Cloning and maintenance packages

FAI

FAI (http://www.informatik.uni−koeln.de/fai/) is an automated system to install a Debian GNU/Linux operating system on a PC cluster. You can take one or more virgin PCs, turn on the power and after a few minutes Linux is installed, configured and running on the whole cluster, without any interaction necessary.

SystemImager

SystemImager (http://systemimager.org) is software that automates Linux installs, software distribution, and production deployment.

DHCP vs. hard−coded IP addresses

If you have DHCP set up, then you don't need to reset the IP address and that part of it can be removed from the `go` script.

DHCP has the advantage that you don't muck around with IP addresses at all provided the DHCP server is configured appropriately. It has the disadvantage that it relies on a centralised server (and like I said, I tend to distribute things as much as possible). Also, linking hardware ethernet addresses to IP addresses can make it inconvenient if you wish to replace machines or change hostnames routinely.

4.4 Known hardware issues

The hardware in general has worked really well for us. Specific issues are listed below:

The AMD dual 1.2 GHz machines run really hot. Two of them in a room increase the temperature significantly. Thus while they might be okay as desktops, the cooling and power consumption when using them as part of a large cluster is a consideration. The AMD Palmino configuration described previously seems to work really well, but I definitely recommend getting two fans in the case−−this solved all our instability problems.

4.5 Known software issues

Some tar executables apparently don't create a tar file the nice way they're supposed to (especially in terms of referencing and de−referencing symbolic links). The solution to this I've found is to use a tar executable that does, like the one from RedHat 7.0.

---

5. Performing tasks on the cluster

This section is still being developed as the usage on my cluster evolves, but so far we tend to write our own sets of message passing routines to communicate between processes on different machines.

Many applications, particularly in the computational genomics areas, are massively and trivially parallelisable, meaning that perfect distribution can be achieved by spreading tasks equally across the machines (for example, when analysing a whole genome using a technique that operates on a single gene/protein, each processor can work on one gene/protein at a time independent of all the other processors).

So far we have not found the need to use a professional queueing system, but obviously that is highly dependent on the type of applications you wish to run.

5.1 Rough benchmarks

For the single most important program we run (our *ab initio* protein folding simulation program), using the Pentium 3 1 GHz processor machine as a frame of reference, on average:

```
Xeon    1.7 GHz processor is about 22% slower
Athlon  1.2 GHz processor is about 36% faster
Athlon  1.5 GHz processor is about 50% faster
Athlon  1.7 GHz processor is about 63% faster
Xeon    2.4 GHz processor is about 45% faster
Xeon    2.7 GHz processor is about 80% faster
Opteron 1.4 GHz processor is about 70% faster
Opteron 1.6 GHz processor is about 88% faster
```

Yes, the Athlon 1.5 GHz is faster than the Xeon 1.7 GHz since the Xeon executes only six instructions per clock (IPC) whereas the Athlon executes nine IPC (you do the math!). This is however an highly non−rigourous comparison since the executables were each compiled on the machines (so the quality of the math libraries for example will have an impact) and the supporting hardware is different.

5.2 Uptimes

These machines are incredibly stable both in terms of hardware and software once they have been debugged (usually some in a new batch of machines have hardware problems), running constantly under very heavy loads. One common example is given below. Reboots have generally occurred when a circuit breaker is tripped.

```
2:29pm  up 495 days,  1:04,  2 users,  load average: 4.85, 7.15, 7.72
```

6. Acknowledgements

The following people have been helpful in getting this HOWTO done:

- Michal Guerquin (Michal Guerquin)
- Michael Levitt (Michael Levitt)

7. Bibliography

The following documents may prove useful to you−−−they are links to sources that make use of high−performance computing clusters:

- Ram Samudrala's research page (which describes the kind of research done with these clusters)
- RAMP web page
- RAMBIN web page