Linux Cluster HOWTO

Table of Contents

Linux Cluster HOWTO

Ram Samudrala (me@ram.org)

v0.3, August 21, 2001

How to set up high−performance Linux computing clusters.

1. Introduction

This document describes how I set up my Linux computing clusters for high−performance computing which I need for my 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. The main use 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, 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.

2.1 Node hardware

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
- Mid−tower case

2.2 Server hardware

1 external server with the following setup:

- 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
- 2 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
- Full−tower case

2.3 Desktop hardware

4 desktops with the following setup:

- 2 Pentium III 1 GHz Intel CPUs
- Supermicro 370 DE6 Dual PIII−FCPGA motherboard
- 4 256 MB 168−pin PC133 Registered ECC Micron RAM
- 3 40 GB Maxtor UDMA/100 7200 RPM HD
- Ricoh 32x12x10 CDRW/DVD Combo EIDE
- 1.4 MB floppy drive
- Asus V7700 64mb GeForce2−GTS AGP video card
- Creative SB Live Platinum 5.1 sound card
- Microsoft Natural Keyboard
- Microsoft Intellimouse Explorer
- Full−tower case

2 desktops with the following setup:

- 2 Pentium III 1 GHz Intel CPUs
- Supermicro 370 DLE Dual PIII−FCPGA motherboard
- 4 256 MB 168−pin PC133 Registered ECC Micron RAM
- 3 40 GB Maxtor UDMA/100 7200 RPM HD
- Mitsumi 8x/4x/32x CDRW
- 1.4 MB floppy drive
- Jaton Nvidia TNT2 32mb PCI
- Creative SB LIVE Value PCI
- Microsoft Natural Keyboard
- Microsoft Intellimouse Explorer
- Full−tower case

2 desktops with the following setup:

- 2 Pentium III 1 GHz Intel CPUs
- Supermicro 370 DLE Dual PIII−FCPGA motherboard
- 4 256 MB 168−pin PC133 Registered ECC Micron RAM
- 3 40 GB Maxtor UDMA/100 7200 RPM HD
- Asus CD−S500 50x CDROM
- 1.4 MB floppy drive
- Jaton Nvidia TNT2 32mb PCI
- Creative SB LIVE Value PCI
- Microsoft Natural Keyboard
- Microsoft Intellimouse Explorer
- Full−tower case

Backup:

- 2 Sony 20/40 GB DSS4 SE LVD DAT

Monitors:

- 4 21" Sony CPD−G500 .24mm monitor
- 2 18" Viewsonic VP−181 TFT−LCD monitor

2.4 Putting−it−all−together hardware

We use KVM switches with a cheap monitor to connect up and "look" at all the machines:

- 15" .28dp XLN CTL Monitor
- 3 Belkin Omniview 16−Port Pro Switches
- 40 KVM cables

While this is a nice solution, I think it's kind of needless. 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). I don't plan to use more monitor switches/KVM cables.

Networking is important:

- 1 Cisco Catalyst 3448 XL Enterprise Edition network switch.

2.5 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, including taxes. Generally, our goal is to keep each node to below $2000.00 (which is what our desktop machines cost).

3. Software

3.1 Linux, of course!

Specfically we use 2.2.17−14 kernel based on the KRUD 7.0 distribution. We use our own software for parallising applications but have experimented with PVM and MPI. In my view, the overhead for these pre−packaged programs is too high.

3.2 Costs

Linux is freely copiable.

4. Set up and configuration

4.1 Disk configuration

This section describes disk partitioning strategies.

```
farm/cluster machines:

hda1 − swap  (2 * RAM)
hda2 − /     (remaining disk space)
hdb1 − /maxa (total disk)
```

```
desktops (without windows):

hda1 − swap  (2 * RAM)
hda2 − /     (4 GB)
hda3 − /home (remaining disk space)
hdb1 − /maxa (total disk)
hdd1 − /maxb (total disk)

desktops (with windows):

hda1 − /win  (total disk)
hdb1 − swap  (2 * RAM)
hdb2 − /     (4 GB)
hdb3 − /home (remaining disk space)
hdd1 − /maxa (total disk)

laptops (single disk):

hda1 − /win  (half the total disk size)
hda2 − swap  (2 * RAM)
hda3 − /     (4 GB)
hda4 − /home (remaining disk space)
```

4.2 Package configuration

Install a minimal set of packages for the farm. Users are allowed to configure desktops as they wish.

4.3 Operating system installation

Cloning

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 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 have a script called `go` that takes a hostname and IP address as its arguments 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 also use Tom's Root Boot package http://www.toms.net/rb/ to boot the machine and clone the system. The `go` script can be placed on a CD−ROM or on the floppy containing Tom's Root Boot package (you need to delete a few programs from this package since the floppy disk is stretched to capacity).

More conveniently, you could burn a bootable CD−ROM containing Tom's Root Boot package, including the `go` script, and the tgz file containing the system you wish to clone. You can also edit Tom's Root Boot's init scripts so that it directly executes the `go` script (you will still have to set IP addresses if you don't use DHCP).

Thus you can develop 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.

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.

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 single gene technique, each processor can work on one gene at a time independent of all the other processors).

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

6. Acknowledgements

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

- 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:

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