A high performance Linux cluster

Gary Benner
Beca Applied Technology/Waiariki Institute of Technology, New Zealand
gary.benner@waiariki.ac.nz

Karam Khokra
Waiariki Institute of Technology, New Zealand
karam.khokra@waiariki.ac.nz


Abstract

A Linux based cluster server consisting of eight discarded Pentium II 350Mhz computers was created at the Waiariki Institute of Technology (School of Business and Computing); the lecturers involved used Linux and other Open Source software. The initial goal of the project was to demonstrate 'high availability' and 'high reliability', achieving three to five 'nines of availability' over a period of a semester. A subsequent stage will be to create a Beowulf cluster from more discarded units with the goal of creating a high performance central processing unit.

The content of the paper has been disseminated internally within the Waiariki Institute of Technology.

Keywords

Linux cluster, Open Source software

1. Introduction

As reported in Benner and Khokra (2004), this project was set up to test the feasibility of creating a Linux based cluster, using discarded computer equipment that was readily available to our staff after a three yearly upgrade of computers at Waiariki. The project would use freely available Open Source software that could be easily used to provide a "highly available" and "highly reliable" platform for the provision of services to students and staff.

A second stage is planned to add "high performance", by creating a Beowulf cluster at the core of the system, to increase the throughput of the central processing area of the cluster.

It was not a concern that we were using "underpowered" computers, as the actual result in terms of throughput was not considered important to the goal of the project at this stage. Once proven, the design could then be implemented using more powerful and intrinsically reliable equipment.

The system would be 'exercised" using an existing software package that all participants were familiar with, the e-learning package called Moodle. The level of reliability is scaled by the number of nines in the percentage uptime, eg 3 "nines of availability" equates to 99.9% uptime and a downtime of 8h 46m per year. 5 "nines of availability" equates to 99.999% uptime and a downtime of 5m 15 sec per year. "3 nines" would be adequate for a top end ISP, and "5 nines" would be relevant for a banking network. "6 nines" with a downtime of 30 secs per year would be required of a military defence system.

2. System Description

The system design of the first stage includes three main levels (Figure 1). Each level would provide a level of redundancy, so that if any one machine at that level was switched off, or failed
during operation, it would no impact on the functionality of the operation of the system as a whole.

2.1 Level One - Load Balancing

This comprises two computers, each connected to the site firewall, and hence to the internet. Only one operates at any one time, however they have a "heartbeat" link so that if the one currently serving the system fails, the other will immediately take over.

Both machines share the same IP numbers, but only one is active at a time so that there is no conflict.

2.2 Level Two - Web Server

This comprises three computers, all of which are connected to the Level 1 Load Balancing computers via dedicated network hosted by a 100Mb/s switch. They are also connected via a separate network / switch to the Level Three computers (see below).

Each of these computers works in parallel providing a degree of high performance operation, as well as providing the design requirement of "high redundancy".

2.3 Level Three - Database & File Servers

This comprises of two computers acting as database and file servers. They are connected as described above to the Web servers in level two. They replicate data to each other on a peer to peer basis. Both servers will be available to the Web servers, providing both redundancy and performance improvements under load.

2.4 Supervisor Level

There is one machine used to manage each of the networks comprising the system. This machine has three network cards, connected to each of the above networks, and is fitted with software to monitor each of the networks, and the state of the system as a whole. This level is not involved in the operational "availability" or "reliability", and serves only to manage and monitor the system.
The software features (type, brand names and source) are summarised in Table 1.

### Table 1. Open source software used

<table>
<thead>
<tr>
<th>Software type</th>
<th>Name</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System (all machines)</td>
<td>Red Hat 9.0</td>
<td><a href="http://www.redhat.com">http://www.redhat.com</a></td>
</tr>
<tr>
<td>Database</td>
<td>MySQL,</td>
<td><a href="http://www.mysql.com">http://www.mysql.com</a></td>
</tr>
<tr>
<td>System Application</td>
<td>Moodle</td>
<td><a href="http://www.moodle.org">http://www.moodle.org</a></td>
</tr>
</tbody>
</table>

#### 3. Methodology

First, to recapitulate, we are testing for the feasibility of creating such a system as described herein. Secondly we wish to test for high availability. We wish also to test for high reliability and (to some extent) for high performance.

##### 3.1 Feasibility

Our approach in testing for feasibility was to implement the Moodle e-learning system. Waaiariki already use this product for the online e-campus site, so we are quite familiar with it's operation and setup, and the live site provides a reference for functionality.

##### 3.2 High Availability

Our approach in testing for high availability was to look purely within the limits of the system itself. We appreciate that there is a single connection to the Internet (hence a single point of failure), however that is not what we are considering at this stage. We were concerned that our system could be stable and functional, in the event that any ONE machine at any level was switched off or unplugged. Theoretically, three machines (each on a different level) could fail and the system as a whole should still operate.

##### 3.3 High Reliability

Our approach in testing for high reliability was to look at how the system would react under various conditions that could be reasonable expected in normal use. We were looking at issues related to network connectivity etc, and were not considering hardware environment, hence UPS and disk mirroring (RAID) were not the subject of the research. Reliability is related to fault tolerance, and hence the methodology is much the same as we adopted for High Availability. In addition, we subjected the system to a sustained series of hits (in as much as our Internet connection allowed).

##### 3.4 High Performance

Our approach to testing performance, was to check for acceptable performance considering the nature of the equipment we were using.

#### 4. Results and Conclusion

We have implemented the Load Balancing, Webservers, and a single Database Server. The Moodle software was installed and operated in the same manner as our e-campus Website. We were able to remove one of the load balancing or Web server computers from each of the levels and the system remained stable. The system has been operating now for some time and has not exhibited any failure, proving that it can be considered a "highly available" system. We are now refining the setup for database replication and file system mirroring. We initially experienced difficulty with using discarded machines, as many had faults that required replacement of parts. It took 20
machines to create 8 reliable, well resourced machines. In the time the system has been running to date, the only downtime has been during upgrades, and a site power outage.

We can conclude with what has been achieved so far, that the approach outlined in this document provides a feasible solution to providing a "high availability", "highly reliable" system, with the potential to provide a "high performance" solution where computing resources are required.

We expect to complete the file system and database replication phase shortly, and will advise details on our Website http://cluster.waiariki.school.nz. Once that is achieved, we will then be running stress testing on the Websites.

Our next step will be to concentrate on the performance of the central processing unit, in particular the Web server and database areas of the system.

References


Copyright © 2004 Benner, G. and Khokra, K.

The author(s) assign to NACCQ and educational non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The author(s) also grant a non-exclusive licence to NACCQ to publish this document in full on the World Wide Web (prime sites and mirrors) and in printed form within the Bulletin of Applied Computing and Information Technology. Authors retain their individual intellectual property rights.

Copyright ©2004 NACCQ.
Krassie Petrova, Michael Verhaart & Colin Boswell (Eds.)
An Open Access Journal, DOAJ # 11764120 , ( zotero)