Developing a high performance Linux cluster at low cost

Gary Benner Karam Khokra John Dunlop Linda Fraser Deon Woods

School of Business and Computing
Waiariki Institute of Technology
Rotorua, NZ
gary.benner@waiariki.ac.nz

The Waiariki Institute of Technology, School of Business and Computing lecturers have created a Linux based cluster server consisting of eight (8) discarded Pentium II 350Mhz computers, using Linux and other open Source software.
The goal was to provide "high availability" and "high reliability", achieving five to six "nines of availability". The next stage will be to create a Beowulf cluster from more discarded units with the goal of creating a high performance central processing unit.

1. INTRODUCTION

This project was set up to test the feasibility, using discarded computer equipment that was readily available to our staff, of creating a Linux based cluster using Open Source software that could be easily used to provide a "high 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 of no concern that we were using "underpowered" computers, as the actual result in terms of throughput was not important to the goal of the project at this stage. Once proven, the design could then be implemented using more powerful equipment.

The system would be 'exercised" using an existing software package that all participants were familiar with, the e-learning package called Moodle.

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 not impact on the functionality of the operation of the system as a whole.

2.2 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.

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 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.

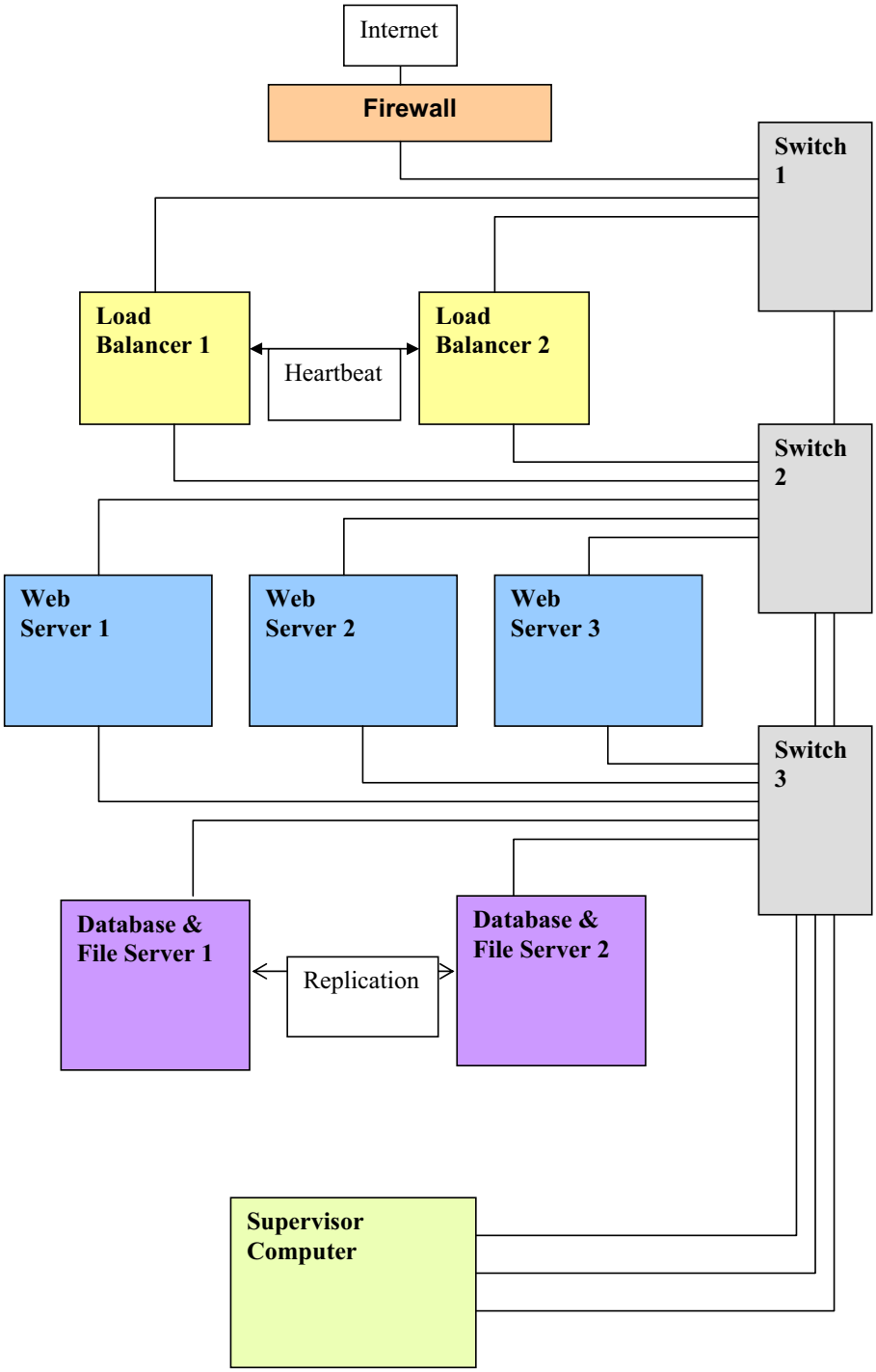

Figure 1: System diagram

2.4 Supervisor Level.

There is one machine used to monitor 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 monitor.

3. METHODOLOGY

Firstly, to recap, we are testing for the feasibility of creating such a system as described herein.

Secondly will test for high availability

Thirdly we will test for high reliability

Fourthly, we will test (in some manner) high performance.

3.1 Feasibility

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

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

We were able to successfully implement and operate the system as outlined. This demonstrated the feasibility of constructing such a system. The Moodle software was installed and operated in the same manner as our e-campus website. We were able to remove one of the 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 subjected our Moodle website implementation to a sustained series of hits (as much as our Internet connection would allow) with no failure of the system, and in fact we noticed that performance remained stable throughout.

It was hoped that test results would have been ready prior to publication but this is not the case. The team are hoping to have results to share at the Focused Discussion Group on July 7.

5. CONCLUSION

We conclude 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.

Our next step is to concentrate on the performance of the central processing unit, in particular the web server / database area of the system.

REFERENCES

1. Bookman C (2002) Linux Clustering: Building and Maintaining Linux Clusters Ist edition. Sams.
2. Buyya R (1999) High Performance Cluster Computing: Architectures and Systems Vol-1. NJ, USA; Prentice Hall PTR.
3. McCarty B (2003) Red Hat Linux Firewalls. Wiley US
4. Hatch B & Kolesnikov O (2002) Building Linux Virtual Private Networks Ist edition. Sams.
5. http://www.linuxjournal.com/
6. http://lcic.org/
7. http://www.beowulf.org/
8. http://www.linux.org/