

This series is intended to provide documentation of a simple server cluster based on two physical servers and four virtual servers per machine, and configured to host an instance of the Moodle Learning Management System.



### Topics

- [Introduction](#)
- [System hardware and Virtual Server environments](#)
- [Load Balancing and failover](#)
- [Web Server, sessions, and updates](#)
- [File server, repositories, synchronisation and network issues](#)
- Database implementation, tuning, and replication
- System monitoring and operations
- System trouble shooting

### System Hardware

For this series I have built the systems on servers that have quad core CPU's and 8G ram. In production, and depending upon use, these can be scaled as required. I have not put emphasis on the HDD configuration, as those issues are dealt with better elsewhere. Needless to say, a production environment will need a RAID configuration to suit your requirement.

The servers will require 2 network ports, and these need to be 1 gigabyte capable.

The system requires a 1G Switch at the front end, where access is gained to the cluster. In a really redundant environment you will require redundant feeds, however that goes beyond the scope of this article.

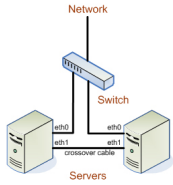
The Virtual Machines will communicate with each other by bridging to the second network port. If you were to scale this design to individual physical machines, you would be able to have individual networks for the connections, but adding additional network cards in this configuration is out of scope.

### Cluster Design

For the purposes of this series of articles, I am defining a "cluster", as a ***grouping or configuration of a group of servers, so that from the outside, they appear to operate as a single entity, however providing the facility to remove one ( or more in a larger configuration) without impacting on the capability of the hosted application(s)***

Note that the capacity of the system may be reduced - ie slower response - but the overall capability must not be compromised, and data integrity must be maintained.

So this design is meant to handle the complete failure of one server, or to permit it's extraction from the cluster for service and maintenance, without impacting on the operation of the Moodle application.



### Virtual Server Environment

Most of my experience has been with VMware, for historical reasons, however I will try and include Xen configuration as well, as I believe this is an excellent alternative.

The design requires four Virtual machines per server, providing the following functions:

- **Failover Load Balancing**

This will be implemented using Linux Virtual Server for load balancing and heartbeat for failover operations. For the heartbeat I'll use the network option, however you can implement this using serial comms to isolate this function from the failover process.

- **Web Server**

This will be implemented using Apache 2 web server, with session handling delegated to the Moodle application. Rsync will be used to maintain the application codebase.

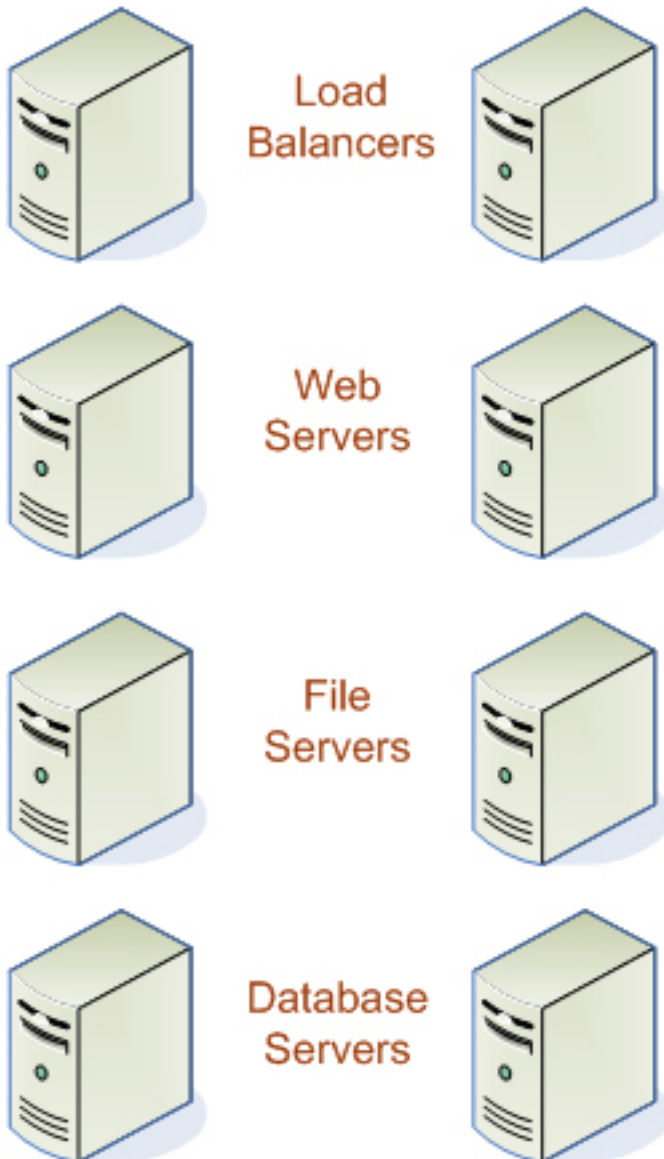
- **File Server**

This will be implemented using NFS, and data will be synchronised using DRDB.

- **Database Server**

The database server will be MySQL, and if time permits Postgres. As a separate project I will be implementing Moodle on the open source Firebird database, as I believe that its heritage from Borland's Interbase, plus a vibrant developer community, and a near Oracle like capability, make it an excellent choice for this sort of application. But more of that later.

# Virtual Machines



## Considerations for Virtualisation Resources

Each of the Virtual Machines require a certain amount of resources. For this design, which is entry - level cluster design, will require the following as a minimum:

- **Load Balancer** - mainly network traffic handling at a kernel level, so allocation of 1 cpu and 500M ram. HDD requirement is minimal.
- **Web server** - mainly application processing, so allocation of 2 cpu and 2G ram. HDD requirement is minimal.
- **File server** - mainly network and HDD operations, so allocation of 1 cpu and 1G ram.
- **Database server** - mainly query processing and HDD access, so allocation of 2 cpu and 4G ram.

This configuration is mindful of the target Moodle application, and the way it works. Any further suggestions or considerations on these parameters are welcome.

### Networking

#### Load Balancer

The eth0 network interface must be allocated an IP address and gateway relevant to the network it is connected to. Note that both Load Balancers are given identical IP numbers, however the reasons for this is covered in the next article.

The eth1 interface is assigned to the internal network (eg, 192.168.10.0/24) which we will use in this example.

The Load Balancer in operation will only communicate with the web servers for normal operation, and only with the file and database servers for management and maintenance purposes.

#### Web server

## A Simple Server Cluster Design - Part 2

Written by Administrator

---

The eth1 interface will be allocated to the internal network ( 192.168.10.11 and 192.168.10.21 respectively)

eth0 will not be used in this configuration.

### **File server**

The eth1 interface will be allocated to the internal network ( 192.168.10.15 and 192.168.10.25 respectively)

eth0 will not be used in this configuration.

### **Database server**

The eth1 interface will be allocated to the internal network ( 192.168.10.18 and 192.168.10.28 respectively)

eth0 will not be used in this configuration.

## A Simple Server Cluster Design - Part 2

Written by Administrator

---

At this stage I will suggest that the reader set up their Virtual environment of choice. In the future I will document this process.