Creating Web Farms with Linux
(Linux High Availability and Scalability)

Horms (Simon Horman)
horms@verge.net.au

October 2000

http://verge.net.au/linux/has/
http://ultramonkey.sourceforge.net/

Introduction: In the Begining

May 1998: "Creating Redundant Linux Servers" presented at Linux Expo.

November 1998: "fake" released.

- Arguably the first HA software available for Linux.
- Implements IP address takeover.

Alan Robertson started a Linux High Availability page focusing on the Linux High Availability HOWTO.

Introduction: Linux High Availability Now

A myriad of closed and open source solutions are now available for Linux.

Fake has largely been superseded by Heartbeat.

"Alan Robertson's HA Page" is now known as "Linux High Availability" can be found at `www.linux-ha.org`

Intelligent DNS and Layer 4 switching technologies are available.

Introduction

This presentation will focus on:

What technologies are available for High Availability and Scalability.

A look at some of the software available.

Examine an a web farm, an application of High Availability and Scalability.

Briefly examine some issues that still need to be resolved.

What is High Availability?

In the context of this paper:

The ability to provide some level of service during a situation where one or more components of a system has failed.

The failure may be scheduled or unscheduled.

The terms fault tolerance and high availability will be used interchangeably.

What is High Availability?

The key to achieving high availability is to eliminate single points of failure.

A single point of failure occurs when a resource has only one source:

- A web presence is hosted on a single Linux box running Apache.
- A site that has only one link to the internet.

What is High Availability?

Elimination of single points of failure inevitably requires provisioning additional resources.

It is the role of high availability solutions to architect and manage these resources.

The aim is that when a failure occurs users are still able to access the service.

This may be a full or degraded service.

What is Scalability?

Scalability refers to the ability to grow services in a manner that is transparent to end users.

Typically this involves growing services beyond a single chassis.

DNS and layer 4 switching solutions are the most common methods of achieving this.

Data replication across machines can be problematic.

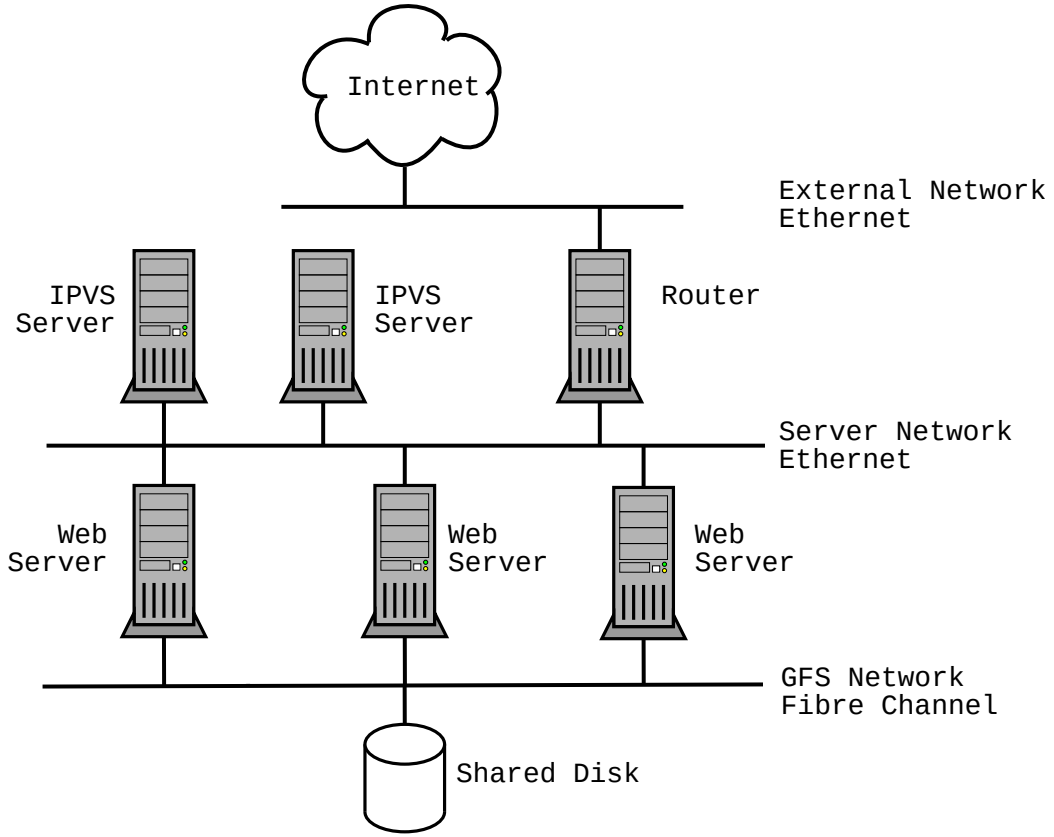
Web Farms

When a service grows beyond the capabilities of a single machine, groups of machines are often employed to provide the service.

In the case of HTTP and HTTPS servers this is often referred to a Web Farm.

Web farms typically employ both high availability and scalability technologies.

Sample Web Farm



Sample Web Farm

Web farms can take many forms.

The three tiered approach is a useful model for explaining how a web farm works.

This sample web farm uses IPVS to handle multiplexing incoming traffic.

Other layer 4 switching technologies are equally applicable.

A DNS based solution would omit the top layer of servers.

Sample Web Farm: Top Layer

Top layer of servers handles the multiplexing of incoming clients to web servers.

Incoming traffic travels through the router to the active IPVS server.

The other IPVS server is a hot stand-by.

IPVS server then routes the client to one of the back-end web servers.

Sample Web Farm: Top Layer

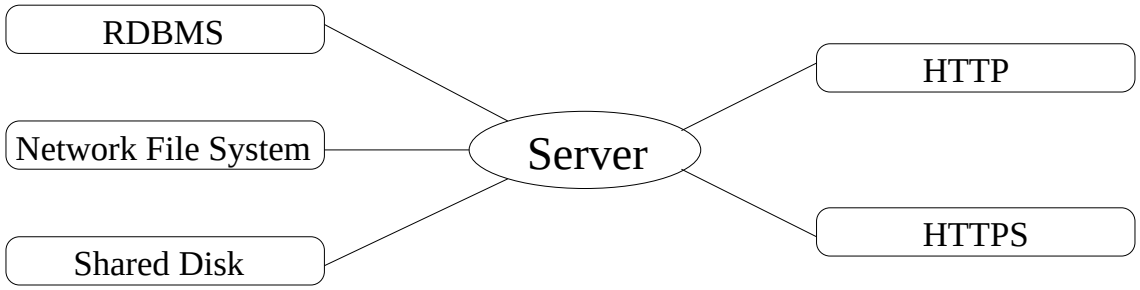
Other host-based layer 4 switching technologies include the Cisco LocalDirector and F5 BIG/ip.

Layer 4 switching servers may also be the gateway routers to the network.

Separating routing and multiplexing functionality enables greater flexibility in how traffic is handled.

A layer 4 switch would eliminate the IPVS servers and form all or part of the switching fabric for the server network.

Sample Web Farm: Middle Layer



The middle layer contains the web servers.

This is the layer where Linux servers are most likely to be found today.

Typically, this layer contains the largest number of servers and these servers contain no content and very little state.

These servers can be thought of as RDBMS or network file system or shared disk to HTTP or HTTPS converters.

Sample Web Farm: Middle Layer

Any complex processing of requests is done at this layer.

Processing power can easily be increased by adding more servers.

Where possible state should is stored on clients by either using cookies or encoding the state into the URL.

- Client doesn't need to repeatedly connect to the same server.
- More flexible load-balancing.
- Session can continue even if a server fails.

Sample Web Farm: Bottom Layer

The bottom layer contains the data or truth source.

There are many options here:

- Servers for network file systems such as NFS and AFS.
- Database Servers for RDBMSs such as Oracle, MySQL, mSQL and PostgreSQL.

In the future server independent storage such as that supported by GFS is likely to be utilised in this layer.

Geographically Distributed Web Farms

Intelligent DNS solutions such as Resonate and Eddieware return the IP address of one of the servers.

A central web server can handle all incoming requests and distribute them using an HTTP redirect.

The rewrite module that ships with the Apache HTTP Server is a very useful method of achieving this.

EBGP can be used to advertise the same network in more than one place and let the routing topology route customers to the most appropriate web server for them.

Geographically Distributed Web Farms

Any instance of a web server in this discussion can be replaced with a web farm.

Yielding a web farm of web farms :-)

Technologies

There are several key technologies that are implemented in many Linux high availability solutions.

The names of these terms can be misleading and even be used to refer to different technologies in other contexts.

Technologies: IP Address Takeover

If a machine, or service running on a machine, becomes unavailable, it is often useful to substitute another machine.

The substitute machine is often referred to as a hot stand-by.

In the simplest case, IP address takeover involves two machines.

Each machine has its own IP address for administrative access.

A floating IP address that is accessed by end-users.

The floating IP address will be assigned to one of the servers, the master.

IP Address Takeover: Interface Initialisation

IP address takeover begins with the hot stand-by bringing up an interface for the floating IP address.

This may be done by using an IP alias.

Once the interface is up, the hot stand-by is able to accept traffic, and answer ARP requests, for the floating IP address.

This does not, ensure that all traffic for the floating IP address will be received by the hot stand-by.

IP Address Takeover: ARP Problems

The master host may still be capable of answering ARP requests for the hardware address of the floating IP address.

If this occurs then each time a host on the LAN sends out an ARP request there will be a race condition.

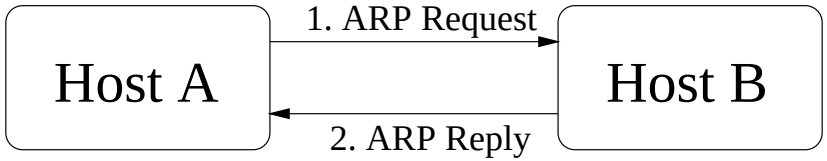
Potentially packets will be sent to the master which has been determined to have failed in some way.

IP Address Takeover: ARP Problems

Even if the master host does not issue ARP replies, traffic will continue to be sent to the interface on the master host.

This will continue until the ARP cache entries of the other hosts and routers on the network expire.

IP Address Takeover: ARP



Host A sends out an ARP request for the hardware address of an IP address on host B.

Host B sees the request and sends an ARP reply containing the hardware address for the interface with the IP address in question.

Host A then records the hardware address in its ARP cache.

Entries in an ARP cache typically expire after about two minutes.

IP Address Takeover: Gratuitous ARP

A gratuitous ARP is an ARP reply when there was no ARP request.

If addressed to the broadcast hardware address, all hosts on the LAN will receive the ARP reply and refresh their ARP cache.

If gratuitous ARPs are sent often enough:

- No host's ARP entry for the IP address in question should expire.
- No ARP requests will be sent out.
- No opportunity for a rouge ARP reply from the failed master.

IP Address Takeover: Restoring the Master

The interface on the hot stand-by for the floating address should be taken down.

Gratuitous ARP should be issued with the hardware address of the interface on the master host with the floating address.

It may be better to use the old master as a hot stand-by and make what was the hot stand-by the master.

If this is done a heartbeat is needed to mediate possession of the floating IP address.

Technologies: Gratuitous ARP Problems

Gratuitous ARP can be used to maliciously take over the IP address of a machine.

Some routers and switches ignore, or can be configured to ignore gratuitous ARP.

For gratuitous ARP to work, the equipment must be configured to accept gratuitous ARP or flush the ARP caches as necessary.

No other known problems with using IP address takeover on both switched and non-switched ethernet networks.

Technologies: Layer 4 Switching

Layer 4 switching is a term that has almost as many meanings as it has people using the term.

For this presentation, it refers to the ability to multiplex connections received from end-users to back-end servers.

Technologies: Layer 4 Switching

This can be implemented in an ethernet switch such as the Alteon Networks ACESwitch.

It can also be done in a host such as:

- The Linux Virtual Server
- Cisco LocalDirector
- F5 BIG/ip
- IBM WebSphere

Layer 4 Switching: Virtual Server

A Virtual Server is the point of contact for by end-users and is typically advertised through DNS.

A virtual server is defined by:

- An IP address.
- A port.
- A protocol: UDP/IP or TCP/IP.

Layer 4 Switching: Scheduling Algorithm

The virtual service is assigned a scheduling algorithm.

The scheduling algorithm determines which back-end server an incoming TCP/IP connection or UDP/IP datagram will be sent to.

Packets for the life of a TCP/IP connection will be forwarded to the same back-end server.

Optionally, subsequent TCP/IP connections or UDP/IP datagrams from a host or network to be forwarded to the same back-end server.

This is useful for applications such as HTTPS where the encryption used relies on the integrity of a handshake made between the client and a server.

Layer 4 Switching: Forwarding

When a packet is to be forwarded to a back-end server, several mechanisms are commonly employed:

- Direct Routing
- IP-IP Encapsulation
- Network Address Translation

Technologies: DNS Methods

One of the simplest ways to effect fail-over is to manually modify the DNS records for a host.

If a server fails then a DNS lookup for the host can return the IP address of another machine.

DNS can also be used to implement scalability by assigning multiple IP addresses to a single hostname in DNS.

Modern DNS servers such as BIND 8.x will deterministically issue the different IP addresses assigned to mail.bigisp.com in a round-robin fashion.

DNS Methods: Problems

The time to live (TTL) on the zone files needs to be turned down severely to to reduce the time for which results are cached.

- The longer the TTL, the less control there is over which IP addresses that end-users are accessing.

- The shorter that TTL, the greater the potential for congestion on the DNS server.

Users may access servers using an IP address rather than a host name.

Users may use non-DNS methods such as an /etc/hosts file to map server host names to IP addresses.

DNS Methods: Round-Robin Specific Problems

The DNS daemon cannot differentiate between a request for a one-off hit, and a request that will result in many hits.

When using round-robin DNS there is no way to assign weights to servers.

DNS Methods: For Geographic Distribution

Intelligent DNS servers can take into account server load, capacity and availability.

Intelligent DNS servers may also take into account the parameters relating to the client.

A robust and simple way to transparently direct clients to geographically separates sites.

Technologies: Heartbeat

A heartbeat is a message sent between machines at a regular interval of the order of seconds.

If a heartbeat isn't received for a time, the machine that should have sent the heartbeat is assumed to have failed.

Used to negotiate and monitor the availability of a resource, such as a floating IP address.

Technologies: Heartbeat

Typically when a heartbeat starts on a machine it will perform an election process with other machines.

On heartbeat networks of more than two machines it is important to take into account network partitioning.

When partitioning occurs it is important that the resource is only owned by one machine, not one machine in each partition.

Heartbeat: Transport

It is important that the heartbeat protocol and the transport that it runs on is as reliable as possible.

Effecting a fail-over because of a false alarm may be highly undesirable.

It is also important to react quickly to an actual failure.

It is often desirable to have heartbeat running over more than one transport.

Existing Solutions

What follows will briefly outline some of the solutions, both open and closed source, that are currently available for Linux.

This is by no means a comprehensive list.

Heartbeat

Author	Alan Robertson
Site	http://linux-ha.org/download/
Licence	GNU General Public Licence

Heartbeat implements a heartbeat protocol.

Runs over raw serial, PPP and UDP/IP over ethernet.

In the case of fail-over, heartbeat effects IP address takeover.

When a fail-over occurs heartbeat can activate or deactivate resources.

A resource is a programme that is executed by heartbeat.

Linux Virtual Server Project (LVS)

Author	Wensong Zhang
Site	http://linux-vs.org/
Licence	GNU General Public Licence

Also known as IPVS: Internet Protocol Virtual Server

Implementation of layer 4 switching in the Linux kernel.

Forwarding Methods:

- Direct Routing
- Tunnelling
- Network Address Translation

Linux Virtual Server Project

Scheduling algorithms:

- Least Connected
- Weighted Least Connected
- Round Robin
- Weighted Round Robin

Eddieware

Vendor	Ericsson
	The Royal Melbourne Institute of Technology
Site	http://eddieware.org/
Licence	Erlang Public Licence

Intelligent HTTP Gateway

- Runs on front-end servers for a site.
- Multiplexes incoming connections to back-end servers.
- Measures load an availability of back-end servers.
- Quality of service metrics can be defined.

Eddieware

Enhanced DNS Server

- Can be used to distribute sites geographically.
- Name server for the domain or domains to be distributed.
- Receives load information from the front-end servers at each site.

TurboCluster

Vendor	TurboLinux, Inc.
Site	http://turbocluster.com/
Licence	Kernel Modules: GNU General Public Licence
	User Level Applications: Closed Source

Supports layer 4 switching and fail-over in much the same way as IPVS and Heartbeat combined.

It is shipped as a modified TurboLinux distribution.

GUI configuration utilities.

Tunnelling and direct routing forwarding mechanisms.

Weighted and non-weighted round-robin scheduling algorithms.

Resonate

Vendor	Resonate, Inc.
Site	http://resonate.com/
Licence	Closed Source

Central Dispatch

- Analogous to the intelligent HTTP gateway component of Eddieware.
- Round robin or resource based scheduling supported.
- Resource based scheduling requires the data in the IP packet be examined.
- User-defined rules based on content and resource availability.

Resonate

Global Dispatch

- Similar in operation to the enhanced DNS server of Eddieware.
- Takes into account network latency between the client's local DNS server and the available sites.

Piranha

Vendor	Red Hat, Inc.
Site	http://redhat.com/
Licence	GNU General Public Licence

Piranha is a suite of tools to configure and manage an IPVS based service.

Older versions of Piranha support a GTK+ front end which is being phased out in favour of an HTML-based configuration tool.

Piranha comes with its own heartbeat tool.

Piranha also has its own tool for monitoring the health of back-end servers and manipulating the IPVS virtual servers.

Ultra Monkey

Vendor	VA Linux Systems, Inc.
Site	http://ultramonkey.sourceforge.net/
Licence	GNU General Public Licence

Disclaimer: This is my project.

Suite of tools to enable a load balanced, highly available farm of servers on a LAN.

LVS for load balancing.

Heartbeat for High Availability betweeen load balancers

Ldirectord monitors the health of real servers

Sample topologies for Higly Available and/or Load Balanced netowrks are provided.

Single Point of Contact for an Open Source solution.

Developing Technologies: The Global File System

The Global File System facilitates access to shared fibre channel disks.

No master node for mediation of resources.

Meta-data stored on the disks using dlocks.

No host acting as a single point of failure or bottleneck.

Disks still represent single points of failure, bit this can be eliminated by using a fibre channel RAID device.

The failure of a fibre channel switch should, at worst, prevent access to part of the fabric.

Developing Technologies: The Global File System

GFS is currently functional.

A network with in excess of 1Tb of storage was recently demonstrated at Linux World.

Still under heavy development

Future Technologies: Generic Framework

Lots of discussion about the need for a generic high availability framework for Linux.

To date the foremost proponent of this has been Stephen Tweedie.

Provide a generic framework instead of specific solutions.

Long term project.

Generic Framework: FailSafe

SGI and SuSE have released that SGI's FailSafe ported to Linux.

Released under the GNU GPL and LGPL.

FailSafe offers a generic API for high availability.

Not as extensive as that proposed by Stephen Tweedie et al.

Conclusion

The support for high availability and scaling services under Linux continues to grow.

At this stage, Linux is well accepted as the middle layer of a scaled service, for example the stateless compute power for a web farm.

Technologies such as layer 4 switching and intelligent DNS implemented under Linux are helping to push Linux towards the front-end of a service.

In the long term, emerging technologies will help Linux to fulfill all the requirements for a highly available, scaled service.