



**Universitat Autònoma
de Barcelona**

Clúster de alta disponibilidad y balanceo de carga sobre un servidor web

Memoria del proyecto
de Ingeniería Técnica en
Informática de Sistemas
realizada por
Alfred Gutiérrez Sanmiguel
y dirigida por
Yolanda Benítez Fernández

Escola d'Enginyeria
Sabadell, *septiembre de 2012*

La abajo firmante *Yolanda Benítez Fernández*,
profesora de “l’Escola d’Enginyeria” de la UAB,

CERTIFICA:

Que el trabajo al que corresponde la siguiente memoria
ha estado realizado bajo su dirección por
Alfred Gutiérrez Sanmiguel.
Y para que conste firme el presente.
Sabadell, **Septiembre** del 2012

Firmado: *Yolanda Benítez Fernández*

HOJA RESUMEN-PROYECTO DE FIN DE CARRERA DE LA ESCUELA DE INGENIERÍA

Título del proyecto: Clúster de alta disponibilidad y balanceo de carga sobre un servidor web.	
Autor: Alfred Gutiérrez Sanmiguel	Fecha: <i>Septiembre 2012</i>
Tutora: Yolanda Benítez Fernández	
Titulación: Ingeniería técnica informática de sistemas	
Palabras clave: <ul style="list-style-type: none">• Catalán: Clúster, balanceo de carga, alta disponibilidad, fallida de sistema, servidor web.• Castellano: Clúster, balanceig de càrrega, alta disponibilitat, fallida de sistema, servidor web.• Inglés: Cluster, load balancer, high availability, failover, web server.	
Resumen del proyecto <ul style="list-style-type: none">• Catalán:<p>El present projecte neix degut a la necessitat de dotar a un pàgina web de una estructura sòlida així com una millor resposta als usuaris. El projecte es pot fraccionar en 2 blocs, els quals son:</p><ul style="list-style-type: none">-La creació de un clúster de alta disponibilitat de manera que assegurem que en cas de fallida de un dels servidors físics la web seguirà sent visible per al públic.-La creació de un clúster de balanceig de càrrega que s'encarrega de redirigir les peticions per a veure la web al node del clúster que sigui més ociós en aquell instant de temps• Castellano:<p>El presente proyecto nace debido a la necesidad de dotar a una página web de una estructura sólida donde asentarse así como una mejora en la respuesta ofrecida a los usuarios de tal servicio. El proyecto se puede fraccionar en 2 bloques, estos son:</p><ul style="list-style-type: none">-La creación de un clúster de alta disponibilidad de manera que aseguramos que en caso de fallida de un de los servidores físicos la web seguirá siendo visible para el público.-La creación de un clúster de balanceo de carga que realiza la función de redirigir las peticiones al nodo del clúster que este más ocioso en aquel instante de tiempo.• Inglés:<p>This project is done due to the need to make a web page become more stable and to improve the loading time from the user's point of view. This project can be clearly separated into 2 blocks:</p><ul style="list-style-type: none">-The creation of a high availability cluster that way we ensure that the web page will be available even if one physical server is down.-The creation of a load balancer cluster who will redirect the web requests to the lest used node at that moment.	

Índice

1. Introducción	13
1.1 Introducción a la memoria	13
1.2 Descripción	13
1.3 Objetivos generales	13
1.4 Beneficios	13
1.5 Motivaciones personales	13
1.6 Estructura de la memoria	14
2. Estudio de viabilidad	15
2.1 Introducción	15
2.1.1 Tipología y palabras clave	15
2.1.2 Descripción	15
2.1.3 Objetivos del proyecto	15
2.1.4 Clasificación de los objetivos	16
2.1.5 Definiciones, acrónimos i abreviaciones	16
2.1.6 Partes interesadas	17
2.1.7 Referencias	18
2.1.8 Documentación del proyecto	18
2.2 Estudio de la situación actual	19
2.2.1 Contexto	19
2.2.2 Lógica del sistema	19
2.2.3 Descripción física	20
2.2.4 Diagnóstico del sistema actual	20
2.2.5 Normativa i legislación	20
2.3 Requisitos del sistema	21
2.3.1 Requisitos	21
2.3.2 Restricciones del sistema	22
2.3.3 Catalogación y priorización de los requisitos	22
2.4 Alternativas y selección de la solución	23
2.4.1 Alternativa 1: VMware	23
2.4.2 Alternativa 2: Microsoft Hyper-V	24
2.4.3 Alternativa 3: Desarrollo propio en Linux	24
2.4.4 Alternativa 4: Desarrollo propio en Windows	25
2.4.5 Conclusiones	25
2.5 Planificación del proyecto	27
2.5.1 Recursos del proyecto	27
2.5.2 Tareas del proyecto	28
2.5.3 Planificación temporal	29
2.6 Evaluación de riesgos	30
2.6.1 Lista de riesgos	30
2.6.2 Catalogación de riesgos	30
2.6.3 Plan de contingencia	31

2.7 Presupuesto	32
2.7.1 Estimación de coste material	32
2.7.2 Estimación de coste personal	33
2.7.3 Resumen y análisis coste beneficio	33
2.8 Conclusiones	34
2.8.1 Beneficios	34
2.8.2 Inconvenientes	34
3. Análisis	35
3.1 Introducción	35
3.2 Análisis clúster Alta Disponibilidad	35
3.3 Análisis clúster Balanceo de carga	39
3.4 Análisis programa empleados	40
3.5 Conclusión del análisis	44
4. Diseño	47
4.1 Introducción	47
4.2 Diseño de la infraestructura física	47
4.3 Alternativas a la infraestructura física	49
4.4 Diseño de la infraestructura virtual	50
4.5 Alternativas a la infraestructura virtual	52
4.6 Conclusión del diseño	53
5. Implementación	55
5.1 Introducción	55
5.2 Instalación y configuración del clúster	55
5.3 Conclusiones de la implementación	61
6. Pruebas	63
6.1 Introducción	63
6.2 Pruebas	63
6.3 Resultados Obtenidos	63
6.4 Conclusiones	66
7. Conclusiones	67
7.1 Conclusiones finales	67
7.2 Desviaciones temporales	67
7.3 Posibles ampliaciones y modificaciones	69
7.4 Valoración personal	69

8. Bibliografía	71
9. Glosario	73
10. Anexo	75

Índice de tabla y figuras

<i>Tabla 01: Clasificación de los objetivos</i>	16
<i>Tabla 02: Stakeholders</i>	17
<i>Tabla 03: Usuarios encargados del desarrollo</i>	17
<i>Tabla 04: Equipo del proyecto</i>	18
<i>Tabla 05: Clasificación de los requisitos</i>	22
<i>Tabla 06: Relación entre objetivos y requisitos del proyecto</i>	22
<i>Tabla 07: Análisis de alternativas</i>	26
<i>Tabla08: Recursos Humanos disponibles</i>	28
<i>Tabla09: Lista de riesgos en el desarrollo del proyecto</i>	30
<i>Tabla10: Impacto de los riesgos en el desarrollo del Proyecto</i>	31
<i>Tabla11: Plan de contingencia</i>	31
<i>Tabla12: Estimación de coste material</i>	32
<i>Tabla13: Coste total de los recursos humanos</i>	33
<i>Tabla14: Definición de nodos</i>	49
<i>Figura 01: Lógica del sistema actual</i>	19
<i>Figura 02: Definición física del sistema</i>	20
<i>Figura 03: Tareas del proyecto</i>	28
<i>Figura 04: Diagrama de Gantt del proyecto</i>	29
<i>Figura 05: Configuración básica clúster Alta disponibilidad</i>	36
<i>Figura 06: Ejemplo Alta Disponibilidad configuración activo/activo</i>	37
<i>Figura 07: Ejemplo Alta Disponibilidad configuración activo/pasivo</i>	38
<i>Figura 08: Ejemplo de failover en clúster de Alta Disponibilidad</i>	39
<i>Figura 09: Clúster de balanceo de carga</i>	40
<i>Figura 10: Funcionamiento de pacemaker en configuración activo/activo</i>	42
<i>Figura 11: Funcionamiento de DRBD</i>	43
<i>Figura 12: Diseño de la infraestructura física</i>	48
<i>Figura 13: Ejemplo de funcionamiento de IP flotante</i>	49
<i>Figura 14: Funcionamiento LAMP</i>	51
<i>Figura 15: Ejemplo de funcionamiento heartbeat en modo de balanceo de carga</i>	51
<i>Figura 16: Ejemplo de funcionamiento pacemaker en modo de activo/pasivo</i>	52
<i>Figura 17: Ejemplo de funcionamiento heartbeat en modo activo/pasivo</i>	52
<i>Figura 18: Configuración particiones de disco en Ubuntu</i>	56
<i>Figura 19: Instalación LAMP server & OpenSSH Server</i>	56
<i>Figura 20: Configuración IP fija nodo ha1-lb</i>	57
<i>Figura 21: Configuración hosts de nodo principal</i>	57
<i>Figura 22: Configuración certificados SSH</i>	58

<i>Figura 23: Configuración heartbeat</i>	58
<i>Figura 24: Configuración claves heartbeat</i>	59
<i>Figura 25: Asignación de permisos certificado y transferencia a nodo secundario heartbeat</i>	59
<i>Figura 26: Configuración Nginx</i>	60
<i>Figura 27: Configuración CRM</i>	61
<i>Figura 28: Configuración CRM II</i>	61
<i>Figura 29: Monitorización del clúster</i>	64
<i>Figura 30: Diagrama de Gantt del proyecto con desviación temporal</i>	68

1. Introducción

1.1 Introducción a la memoria

Este documento define el proyecto en su totalidad de manera que se convierte en una parte indispensable para la comprensión y la evaluación de este y , tal vez en un futuro, se convierta en la guía lo más completa posible para una posterior mejora u modificación.

1.2 Descripción

El proyecto realizado consiste en la elaboración de un clúster de alta disponibilidad con balanceo de carga sobre el servicio web de una empresa.

El montaje de esta infraestructura esta enfocado a ser la sólida base del servicio web, permitiendo poder construir encima de estos más servidores web para una respuesta más rápida a los clientes, exportar el diseño empleado a otros servicios internos y para iniciar la incursión del software libre en la empresa.

1.3 Objetivos generales

El objetivo principal es tener un clúster que soporte de manera estable el servicio web ofrecido a los clientes. A su vez ha de disponer de un método que permita bascular la carga de las peticiones de visitas a la web entre los diversos servidores disponibles.

1.4 Beneficios

Una vez cumplidos los objetivos propuestos, el beneficio obtenido más directo es la estabilidad ofrecida así como una mejora en el tiempo de respuesta de la web corporativa.

De manera secundaria se ve afectada la infraestructura de los servidores de la empresa, mejorando la organización y documentando paso por paso los puntos a seguir para poder usar el método empleado en otros servicios si se requiriese.

1.5 Motivaciones personales

A nivel personal este proyecto me da la oportunidad de probar soluciones administrativas que nunca antes había llevado a cabo, situación que conlleva un reto personal.

El entorno Debian-like me es favorable dada mi experiencia en entornos Ubuntu y la administración de estos servidores me resulta un atractivo dado que habitualmente la configuración de estos es ligeramente más compleja que en sus homólogos de Microsoft.

Por otro lado llegar a comprender de manera total el funcionamiento de este complejo sistema por seguro que me ayudara en mi desarrollo profesional y a satisfacer mis inquietudes informáticas.

1.6 Estructura de la memoria

La memoria se divide en cuatro bloques:

Inicialmente se ha realizado una introducción para describir tanto el proyecto como la memoria y situarnos en contexto así como cerciorarnos de los motivos y los objetivos de manera global.

En el segundo bloque encontramos el estudio de viabilidad donde se especifican los objetivos, los requisitos y las alternativas. Al analizar la viabilidad del proyecto también debemos incluir las personas que intervienen en su desarrollo y su normativa.

Posteriormente detallaremos la planificación del proyecto donde definiremos los riesgos, los costes y los recursos necesarios para el desarrollo de este.

De aquí en adelante, el tercer bloque, los contenidos son de tipo más técnico puesto que están centrados en explicar el funcionamiento del clúster así como el proceso empleado en la elaboración del mismo y las pruebas realizadas a este.

Para finalizar se elabora una conclusión, cuarto y último bloque, que analiza la realización de este proyecto.

A parte de estas secciones se especifica la bibliografía, un anexo y un índice de tablas y figuras.

2. Estudio de viabilidad

2.1 Introducción

Cada vez el mundo de la informática está más y más presente en todos los ámbitos de la vida. Tanto es así que las páginas webs corporativas se han vuelto indispensables en las empresas de hoy en día. Más aun si son del ámbito de la informática y las telecomunicaciones.

Con el fin de mejorar el máximo posible los Servicios que ofrece nuestra empresa, se ha decidido crear un clúster de alta disponibilidad y balanceo de carga para el servidor web. Dado que es una las estructuras más usadas en las empresas para garantizar la disponibilidad de alguno de sus servicios por sus numerosas ventajas frente a la estructura clásica sin el clúster.

2.1.1 Tipología & Palabras clave

Tipo de proyecto: Implantación de sistemas

Palabras clave: Clúster, High Availability, alta disponibilidad, balanceo de carga, servidor web, virtualización, Linux, Gnu, máquinas virtuales, software libre.

2.1.2 Descripción

Partiendo de la base que la página web de una empresa es su cara visible en el mundo digital, si esta empresa se dedica a la informática debe disponer de una a la altura de las expectativas, tanto a nivel de diseño, como de codificación así como de infraestructura.

Con la finalidad de ganar robustez, estabilidad y rapidez en el servicio ofrecido así como causar buena impresión de cara al cliente potencial la empresa ha decidido implantar un clúster de alta disponibilidad con balanceo de carga para soportar el servicio web garantizando la disponibilidad de este, salva catástrofe, y una mejora en la percepción del usuario.

2.1.3 Objetivos del proyecto

A continuación se detallan los objetivos que pretende conseguir este proyecto:

- O1. Aumentar la robustez del sistema
- O2. A partir del entorno dado hacer el clúster de alta disponibilidad
- O3. Monitorizar información del uso del clúster
- O4. Balanceo de carga del clúster según uso del procesador del nodo
- O5. Conseguir balancear la carga de peticiones a los diferentes nodos del clúster

2.1.4 Clasificación de los objetivos

En la siguiente tabla se clasifican los objetivos según su relevancia:

Objetivo	Crítico	Prioritario	Secundario
O1	X		
O2	X		
O3		X	
O4			X
O5	X		

Tabla 01: Clasificación de los objetivos

2.1.5 Definiciones, acrónimos y abreviaciones.

Las definiciones, acrónimos y abreviaturas usadas durante este documento son:

Clúster: Conjunto de ordenadores construidas mediante la utilización de hardware común y que se comportan como si fuesen una única computadora.

Nodo: Cada uno de los ordenadores que forman parte del clúster.

Máquina virtual: Software que emula un ordenador y puede ejecutar programas como si fuese un ordenador real.

Clúster de alta disponibilidad: Clúster formado por dos o más máquinas que comparten una serie de servicios y están constantemente monitorizándose entre si.

Balanceo de carga: Concepto informático empleado en la técnica usada para compartir el trabajo entre varios procesos, ordenadores u otros recursos.

La nube: metodología que permite ofrecer servicios a través de internet.

2.1.6 Partes interesadas

A continuación se describen las partes interesadas en el proyecto. Se considera parte interesada del proyecto a todas las personas o grupos de personas que tienen alguna relación en un sentido o en otro con el proyecto.

Stakeholders

Nombre	Descripción	Responsabilidad
S1	Director del proyecto	Supervisa el trabajo del alumno. Evalúa su proyecto. Aprueba los requerimientos y las funcionalidades.
S3	Responsable de sistemas	Define los requisitos y funcionalidades del proyecto.

Tabla 02: Stakeholders

Perfiles de usuarios

Nombre	Descripción	Responsabilidad
U1	Ingeniero informático	Diseña el clúster y lo integra con el sistema operativo dado.
U2	Usuario genérico	Uso del servicio web ofrecido por la empresa.
U3	Administrador Sistemas	Supervisa la correcta utilización del sistema ya existente.

Tabla 03: Usuarios encargados del desarrollo

Equipo del proyecto

Nombre	Descripción	Responsabilidad
JP	Jefe del proyecto	Supervisa el proyecto y gestiona el desarrollo.
A	Administrador Sistemas	Analiza el proceso de implantación y participa en el proceso de implantación.
AR	Administrador de Redes	Analiza el proceso de implantación y participa en el proceso de implantación y diseño.
TP	Técnico de pruebas	Participa en el diseño y en las pruebas de control de calidad.

Tabla 04: Equipo del proyecto

2.1.7 Referencias

Las referencias que se ha de tener en cuenta es la siguiente:

Normas de entrega del proyecto de ingeniería técnica informática

http://www.uab.cat/Document/541/595/Normativa_PFCNovembre2010.pdf

2.1.8 Documentación del proyecto

A la finalización del proyecto se entregará memoria de este. A su vez, se realizará una exposición a nivel informativo para dar a conocer el clúster.

2.2 Estudio de la situación actual

2.2.1 Contexto

Inicialmente el servicio web de la empresa esta alojado en la nube a través de una empresa subcontratada con ese fin. Al vencer tiempo de uso establecido por esta tercera empresa se plantea que hacer con la web corporativa. Al no estar satisfecho con el comportamiento de la web en la nube se decide migrarla dentro de la propia empresa.

Dada la necesidad de ofrecer un servicio sólido se plantea la posibilidad de crear un clúster en el centro que iguale o supere las características ofrecidas en la nube.

Con la finalidad de alcanzar las expectativas se plantea crear un clúster de alta disponibilidad, a través de este método garantizaríamos que el servicio permanecerá encendido en caso de catástrofe.

Una vez solventado el problema de la disponibilidad nos centramos en la rapidez de respuesta. Para solventarlo se pretende usar la misma estructura creada con anterioridad basada en un clúster y dispersar la carga de las solicitudes de la web a todos los nodos de la manera más homogénea posible.

2.2.2 Lógica del sistema actual

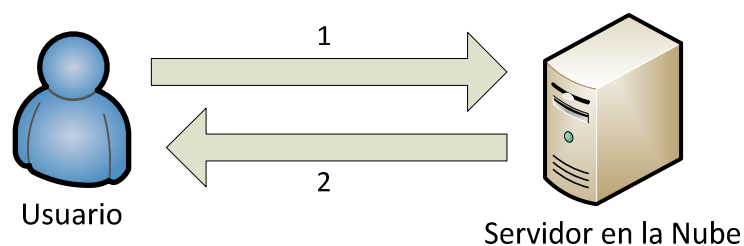


Figura 01: Lógica del sistema actual

Descripción de los pasos:

1. El usuario solicita visualizar la web
2. El servicio en la nube proporciona el servicio

2.2.3 Definición física

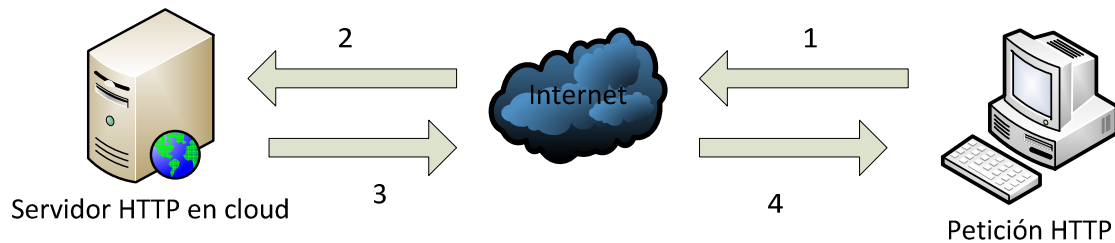


Figura 02: Definición física del sistema

Descripción de los pasos:

1. Realización de la petición HTTP
2. Captación de la petición HTTP por el servidor en cloud
3. Respuesta de la petición HTTP por el servidor cloud
4. Recepción de la petición HTTP

2.2.4 Diagnostico del sistema actual

Deficiencias

- Fragilidad de los servidores
- Información sin copias de seguridad
- Falta de documentación de los servidores
- Información dispersa

Mejoras

- Robustez del sistema
- Rapidez de respuesta
- Duplicado de la información
- Documentación del sistema
- Centralización de la información en un único clúster

2.2.5 Normativa y legislación

Licencias SO

<http://www.gnu.org/licenses/>

Licencias Vmware

<http://www.vmware.com/help/legal.html>

2.3 Requisitos del sistema

En este apartado se definirán los requisitos del sistema y se definirán las restricciones de este. Con este propósito realizaremos un análisis del clúster que nos permitirá obtener buenos resultados y detectar errores.

Debemos hacer inciso que hay casos que no podemos solventar, como podría ser el caso de un fallo eléctrico total, el de la caída masiva de todos los servidores en la empresa o el de una avería en la línea telefónica contratada para dar servicio.

2.3.1 Requisitos

Los requerimientos describen el comportamiento básico del clúster y especifican las operaciones necesarias para lograr que funcione de manera correcta.

- Ofrecer servicio web.
- Monitorizar, el administrador debe tener las herramientas necesarias para controlar el estado de sus componentes y percatarse de los fallos.
- Copias de seguridad de los datos albergados en él.
- El nuevo clúster debe contar con un sistema transparente y escalable que pueda crecer cuando sea necesario.
- Debe ser capaz de mantener el servicio a pesar del fallo de cualquiera de sus nodos.
- Debe gestionar la carga de peticiones HTTP al servidor web más ocioso.
- El clúster debe ser capaz de procesar el volumen actual de trabajo basándose en uso del hardware disponible.
- Se establece una batería de pruebas encaminados a probar la estabilidad, el rendimiento y la usabilidad del sistema propuesto.

2.3.2 Restricciones del sistema

Las restricciones del sistema son las siguientes:

- El clúster ha de estar montado sobre un servidor VMware ESXi
- Debe de ser creado con herramientas de software libre.
- Ha de ser fácilmente monitorizarle.
- El proyecto ha de estar finalizado para septiembre del 2012

2.3.2 Catalogación y priorización de los requisitos.

Prioridades de los requisitos:

	Esencial	Condicional	Opcional
R1	X		
R2	X		
R3		X	
R4	X		
R5	X		
R6		X	
R7			X
R8	X		

Tabla 05: Clasificación de los requisitos

Relación entre requisitos y objetivos.

	R1	R2	R3	R4	R5	R6	R7	R8
O1	X	X		X	X	X		X
O2			X	X	X	X		X
O3	X	X	X					
O4							X	
O5	X	X	X	X	X	X		X

Tabla 06: Relación entre objetivos y requisitos del proyecto

2.4 Alternativas y selección de la solución.

Partiendo de la base que todo el software usado debe ser bajo licencia gratuita, salvo notable mejora en el resultado final, y que debe pesar lo menos posible para buscar la máxima rapidez esta es la lista de alternativas a considerar.

2.4.1 Alternativa 1: *VMware ESXi*

Vmware ESXi es una plataforma que permite virtualizar a nivel de centro de datos. Este componente se encuentra en el nivel inferior de la capa de virtualización. Esta compuesto de un sistema operativo muy reducido, llamado hypervisor, que ofrece los servicios para la gestión y administración de máquinas virtuales. Sus características y funcionalidades serian las siguientes:

- De gran integración en la arquitectura actual del sistema de la empresa.
- Ligereza de infraestructura.
- VMware ofrece soluciones muy correctas y funcionales.
- Ofrece la posibilidad de crear piscinas de máquinas virtuales que funcionen entre si como si de un clúster de alta disponibilidad se tratase.
- A su vez permite mover un usuario entre máquinas virtuales si el uso de una de estas está siendo elevado, no basándose en el número de peticiones web.
- VMware es una empresa con una base muy sólida en cuanto a virtualización se refiere, esto garantiza seguridad a la hora de trabajar con estas.
- Robusto sistema de ayuda online y páginas de ayuda.
- Algunos de los componentes de VMware son de pago y esto comportaría pagar las licencias necesarias para el uso de este.
- La administración del clúster seria a través de interfaz gráfica, cosa que facilita la faena.
- Esta solución tan sólo nos gestionaría las máquinas virtuales que ofreciesen el servicio web, una vez creadas deberíamos elegir sistema operativo, servidor web y corregir los posibles errores

generados por el hecho de tener varios servidores ofreciendo servicio web.

2.4.2 Alternativa 2: *Microsoft Hyper-V*

Al igual que su homólogo en la compañía VMware, Microsoft ha creado un hypervisor que nos permite crear, gestionar y administrar máquinas virtuales desde un administrador. Sus características y funcionalidades serían las siguientes:

- De gran funcionalidad dado que ofrece prácticamente las mismas herramientas que VMware.
- Mucho más pesado que el anterior, más del triple de tamaño.
- Incompatibilidad directa entre máquinas virtuales de VMware & Microsoft.
- No cumple con la restricción de estar montado en ESXi.
- Permite hacer una piscina de máquinas virtuales que funcionen entre sí como un clúster de alta disponibilidad, así como un balanceo de carga en función de los recursos de estas máquinas. Al igual que la anterior alternativa el balanceo de carga no se basa en el número de peticiones web.
- Extenso sistema de ayuda online, Microsoft dispone de su propia web donde da soporte a este tipo de desarrollo.
- El sistema que permite crear y albergar las máquinas virtuales es gratuito, pero muchos de los componentes son de pago y a partir de un número de licencias gratuitas empleadas se vuelve de pago.

2.4.3 Alternativa 3: Desarrollo propio en Linux.

Esta alternativa se basa en la creación de varias máquinas virtuales dentro de la infraestructura ya existente en la empresa, es decir sobre el servidor VMware ESXi ya existente y que, a través de software libre, lograsen el objetivo deseado. Sus características y funcionalidades serían las siguientes:

- Sistema operativo base Ubuntu Server y todos los programas usados libres de coste por licencia, dado que se distribuyen bajo la licencia gratuita GNU.
- Desarrollo a medida, se ajusta a los requisitos deseados.

- Escalable en un futuro basándose en la documentación del proyecto.
- El coste derivará únicamente de la planificación prevista.
- Amplia ayuda online a través de foros de Ubuntu con extensos tutoriales.
- Conjunto de sistema operativo y programas muy ligero puesto que se basa en versiones de servidores sin tan siquiera interfaz gráfica.
- Aumento de la complejidad de la solución con respecto a las alternativas 1 & 2.

2.4.4 Alternativa 4: Desarrollo propio en Windows.

Alternativa basada en la creación de un clúster entre máquinas virtuales usando Windows y reutilizando parte de los recursos ya existentes, como el servidor VMware ESXi. Sus características y funcionalidades serían las siguientes:

- Sistema operativo base Windows Server 2008 y los programas usados con el fin de crear clústeres requieren de una versión determinada con un coste por licencia elevado.
- Desarrollo a medida, se ajusta a los requisitos deseados al igual que en la alternativa anterior.
- Escalable en un futuro basándose en la documentación del proyecto.
- Amplia ayuda online, Microsoft dispone de su propia web donde da soporte a este tipo de desarrollo.
- Conjunto de sistema operativo y programas más pesado que la alternativa anterior.
- Aumento de la complejidad de la solución con respecto a las alternativas 1 & 2.

2.4.5 Conclusiones

En la siguiente comparativa se ponen en común los aspectos identificados como más relevantes de las cuatro alternativas para la selección de la solución más óptima.

	Coste adquisición	Costes adaptación	Escalabilidad	Ajuste requerimientos	Complejidad	Formación
A1	Moderada	Medio/Bajo	Alta	Medio	Baja	Se desconoce
A2	Elevada	Alto	Alta	Bajo	Media	Se desconoce
A3	0€	Bajo	Alta	Alto	Alta	Incluida en el proyecto
A4	Elevada	Bajo	Alta	Medio	Alta	Se desconoce

Tabla 07: Análisis de alternativas

A la vista de las características de las cuatro alternativas, los costes y la tabla comparativa, se ha decidido escoger la **alternativa 3** (Desarrollo propio en Linux) porque es la que mejor se adapta a nuestras necesidades tanto en el coste de adquisición como en el coste de adaptación a la infraestructura ya existente. Por último porque podemos observar que es la mejor manera de llegar a obtener los requisitos que hemos establecido con anterioridad.

2.5 Planificación del proyecto

- **Calendario del proyecto:** Por motivos de disponibilidad, el desarrollo del proyecto comenzará el 15 de Enero del 2012, con una dedicación de 15 horas semanales, (4 horas cada día de lunes a viernes). El total de horas dedicadas al proyecto serán cercanas a las 285.
- **Fecha de inicio:** 15 de Junio del 2012.
- **Fecha de finalización:** En un principio la fecha prevista para la finalización del proyecto es el día 11 de Septiembre del 2012.
- **Herramientas de planificación de control:** Microsoft Project.

2.5.1 Recursos del proyecto

Respecto a los recursos materiales, se utilizarán programas bajo licencias libres. La maquinaria es propia de la empresa y consta de las siguientes características:

-Maquinaria desarrollo:

- 2 x Servidor HP Proliant Microserver ML35
 - Servidores con hypervisor Vmware ESXi
 - 8 de RAM
 - 128 Gigas de disco duro
 - Intel xenón de 2 núcleos de 1,5 ghz
 - Conexión a la red interna & a internet
 - SAI iDialog Plus

-Maquinaria Producción:

- 2 x Servidor HP Proliant server g5
 - Servidores con hypervisor Vmware ESXi
 - 8 de RAM
 - 320 Gigas de disco duro (Controladora Raid 0 a nivel de HW)
 - Intel xenón de 4 núcleos de 2,5 ghz
 - Conexión a la red interna & a internet
 - SAI iDialog Plus

-Programas & Sistemas Operativos:

- Ubuntu versión LTS 10.04
- Servidor web: Apache
- Base de datos: MySQL
- Balanceo de carga: Nginx
- Alta Disponibilidad: Pacemaker, Heartbeat
- Conexión SSH: Open SSH

- Conexión FTP: VSFTPD
- Editor de textos: Nano
- Documentación: OpenOffice, MS Project

- Recursos humanos

Función	Coste/h
Director de Proyecto	60€
Jefe de proyecto	55€
Técnico de sistemas	30€
Técnico de pruebas	20€

Tabla08: Recursos Humanos disponibles

2.5.2 Tareas del proyecto

Teniendo en cuenta que se trabajan 4 horas diarias y de manera secuencial, la planificación aproximada de las tareas del proyecto es la siguiente:

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Cluster de alta disponibilidad y balanceo de carga	62,25 días	vie 15/06/12	mar 11/09/12	
Planificación	4,25 días	vie 15/06/12	jue 21/06/12	
Estudio de viabilidad	15 horas	vie 15/06/12	mié 20/06/12	
Aprobación del estudio de viabilidad	1 hora	jue 21/06/12	jue 21/06/12	3
Definición de requisitos y soluciones	7,25 días	vie 22/06/12	mar 03/07/12	
Especificación de requisitos	10 horas	vie 22/06/12	mar 26/06/12	4
Definición de la solución	5 horas	mié 27/06/12	jue 28/06/12	6
Documentación de requerimientos y solución	5 horas	vie 29/06/12	lun 02/07/12	7
Aceptación de requerimientos y solución	1 hora	mar 03/07/12	mar 03/07/12	8
Diseño, desarrollo y pruebas del cluster	33,25 días	mié 04/07/12	lun 20/08/12	
Diseño & análisis	60 horas	mié 04/07/12	mar 24/07/12	9
Implementación	60 horas	jue 26/07/12	mié 15/08/12	11
Proves	8 horas	jue 16/08/12	vie 17/08/12	12
Aceptación e implantación	1 hora	lun 20/08/12	lun 20/08/12	13
Transición	5 días	mar 21/08/12	lun 27/08/12	
Puesta en funcionamiento	20 horas	mar 21/08/12	lun 27/08/12	
Elaboración de la memoria	32 horas	mar 28/08/12	jue 06/09/12	16
Presentación del PFC	8 horas	vie 07/09/12	lun 10/09/12	17
Cierre del proyecto	1 hora	mar 11/09/12	mar 11/09/12	18

Figura 03: Tareas del proyecto

2.5.3 Planificación temporal

Tal y como se ha comentado anteriormente, en el diagrama de Gantt que tenemos a continuación podemos observar que se sigue un orden secuencial a la hora de realizar las tareas.

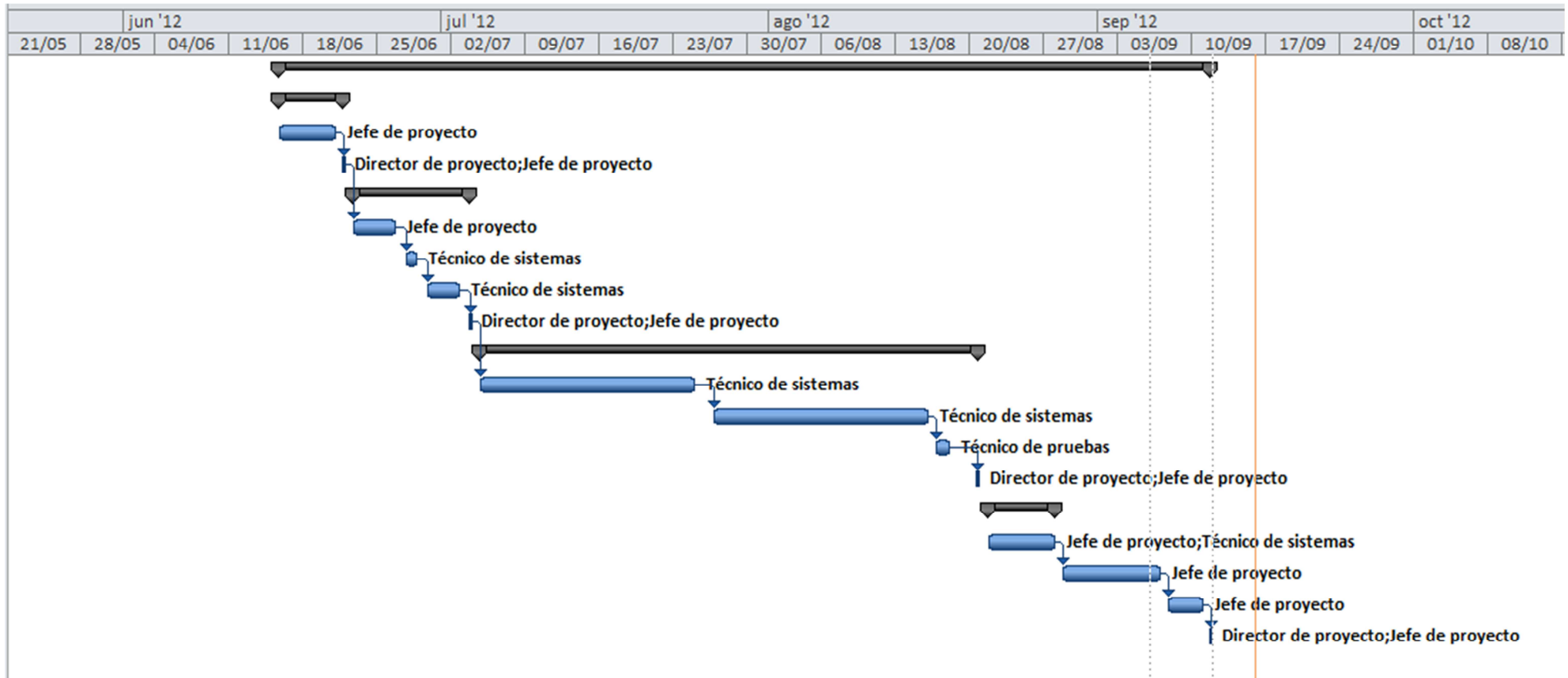


Figura 04: Diagrama de Gantt del proyecto

2.6 Evaluación de riesgos

A continuación se valorarán los riesgos que pueden hacer que el desarrollo del proyecto no se ejecute según lo planificado, teniendo en cuenta su impacto y su posible solución.

2.6.1 Lista de riesgos

En la siguiente tabla vemos una lista de riesgos a los que está sometido el proyecto en las diferentes fases del desarrollo así como sus consecuencias.

Nº	Nombre	Afecta a	Consecuencias
R01	Planificación temporal optimista	Planificación de proyecto	No se acaba en la fecha estimada provocando un aumento de los recursos
R02	Falta de alguna tarea necesaria	Planificación de proyecto	No se cumplen los objetivos marcados
R03	Presupuesto no ajustado	Planificación de proyecto	Posibles pérdidas económicas y repercusión negativa en la calidad
R04	Cambio de requisitos	Estudio de viabilidad y fase de análisis	Demora en el desarrollo
R05	Equipo de proyecto reducido	Planificación de proyecto	Demora en la finalización del proyecto
R06	Programas empleados inadecuados	Fase de desarrollo	Retraso en la finalización del proyecto, menor calidad.
R07	Descuidar al fase de test	Fase de desarrollo e implementación	Deficiencia operativa, insatisfacción por parte de la empresa y pérdida económica.
R08	Carencia de medidas de seguridad	Estudio de viabilidad, fase de análisis y desarrollo	No se cumplen los objetivos.
R09	Abandono	Al proyecto	Pérdidas económicas y frustración.

Tabla09: Lista de riesgos en el desarrollo del proyecto

2.6.2 Catalogación de riesgos

Procedemos a mostrar la tabla que detalla la probabilidad de incidencia de los riesgos detectados y el impacto que tendría en el desarrollo del proyecto.

Nº	Probabilidad	Impacto
R01	Alta	Crítico
R02	Alta	Crítico
R03	Alta	Crítico
R04	Alta	Marginal
R05	Alta	Crítico
R06	Baja	Crítico
R07	Alta	Crítico
R08	Alta	Crítico
R09	Mediana	Catastrófico

Tabla10: Impacto de los riesgos en el desarrollo del proyecto

2.6.3 Plan de contingencia

En la *Tabla 11* se presentan las posibles soluciones a los diversos riesgos expuestos con anterioridad.

Nº	Solución
R01	Retrasar algunas fechas, afrontar posibles pérdidas.
R02	Revisar la planificación del proyecto, modificar la planificación.
R03	Renegociar con la empresa, afrontar posibles pérdidas.
R04	Renegociar con la empresa, modificación planificación y presupuesto.
R05	Pedir retraso en la finalización, negociar con empresa nuevos miembros.
R06	Mejorar la formación del equipo, prevenir herramientas alternativas.
R07	Diseñar los test con antelación, dar garantías y afrontar pérdidas.
R08	Revisar la seguridad en cada fase, aplicar políticas de seguridad activas.
R09	Sin solución.

Tabla11: Plan de contingencia

2.7 Presupuesto

A continuación se muestra los costes estipulados para el desarrollo del proyecto.

2.7.1 Estimación de coste material

En la siguiente tabla se enumeran los costes derivados de la compra de los recursos materiales necesarios para el desarrollo i la explotación del proyecto.

Recurso	Coste
Servidor desarrollo	0€ (Ya amortizado por la empresa)
Servidor Producción	0€(Ya amortizado por la empresa)
Ubuntu Server 10.04 LTS	0€
NGINX	0€
Pacemaker	0€
HeartBeat	0€
OpenSSH	0€
VSFTPD	0€
NANO	0€
OpenOffice	0€
MS Project	40€

Tabla12: Estimación de coste material

Tal como podemos observar, la mayoría de los recursos utilizados son de licencia gratuita, esto no implica que su funcionalidad ni su rendimiento sea más limitado que su homólogo de pago.

La amortización de los servidores corre a cuenta de la empresa que contrata al administrador de sistemas para desarrollar esta solución informática. El uso de MS Project es el único valor por el que hemos de desembolsar dinero, puesto que su homólogo en OpenOffice no esta a la altura esperada.

2.7.2 Estimación de coste personal

En la *Tabla 13* se define el coste del personal implicado directamente en el proyecto.

Función	Coste/h	Horas	Coste total
Director de Proyecto	60€	4	240€
Jefe de proyecto	55€	89	4894€
Técnico de sistemas	30€	150	4500€
Técnico de pruebas	20€	20	160€
Total			9794€

Tabla13: Coste total de los recursos humanos

2.7.3 Resumen y análisis coste beneficio

Una vez calculados los costes de material y de los recursos humanos, podemos calcular cual será el aproximadamente el coste total del proyecto.

- Coste material.....40€
 - Coste personal.....9794€
- Total.....9834€

El proyecto tiene un coste elevado pero, invertir en una buena infraestructura de sistemas dentro de una empresa es un requisito indispensable de cara a una auditoria o simplemente a generar un servicio de calidad a los usuarios de la web corporativa. A su vez es una infraestructura que puede ser plagiada en otros servicios así que se puede reusar el diseño por completo del clúster generado.

2.8 Conclusiones

En este apartado, una vez realizado prácticamente todo el estudio de viabilidad, es el momento de poner sobre la mesa tanto los beneficios como los inconvenientes de la realización del proyecto y se decida si el proyecto es viable o no.

2.8.1 Beneficios

- Mejora en el tiempo de respuesta de la web gracias al balanceo de carga generado por el clúster.
- Aumento de la robustez del sistema dado que debería haber una fallida total de electricidad para tumbar el servicio web ofrecido.
- Aumento de experiencia en redes y clústeres por parte del alumno
- Posibilidades de utilizar esta infraestructura en otros servicios de la empresa.
- Derecho a obtener un título universitario

2.8.2 Inconvenientes

- Inversión inicial elevada
- Se requiere de un tiempo para el desarrollo del clúster

En vista que los beneficios que se obtienen son muy gratificantes tanto para el alumno como para la empresa y teniendo sus inconvenientes, siempre cumpliendo los requisitos necesarios, podemos concluir que el proyecto es:

Beneficios + Inconvenientes = Proyecto viable

3. Análisis

3.1 Introducción

El objetivo de este capítulo es evaluar un clúster, familiarizarse con los diferentes tipos que existen y la funcionalidad que estos ofrecen.

Esta solución es cada vez más demandada por las empresas con la finalidad de cubrir sus expectativas de calidad de servicio y a su vez controlar una importante inversión en sistemas que les permita crecer y, a la misma vez, ofrecer nuevas soluciones en el mundo de los sistemas informáticos.

Esta solución suele ser más barata que la compra de un único ordenador individual y suele ofrecer características de rendimiento y/o disponibilidad iguales o superiores.

Los tipos de clústeres eminentemente son los siguientes o combinaciones de estos:

Alto rendimiento, diseñado para dar altas prestaciones en cuanto a capacidad de cálculo.

Alta disponibilidad, conjunto de dos o más máquinas que comparten servicios y que se monitorizan constantemente entre si.

Balanceo de carga, conjunto de dos o más máquinas que comparten el peso del trabajo a realizar entre si.

Focalizándonos en nuestro caso particular no tiene sentido analizar el clúster de alto rendimiento puesto que no lo usaremos, así pues nos centraremos en el análisis de los dos tipos restantes.

3.2 Análisis clúster Alta Disponibilidad

La principal característica de un clúster de alta disponibilidad es mantener una serie de servicios compartidos y que cada uno de los nodos que forman el clúster sepa en todo momento que esta haciendo el otro.

Este clúster debe tener un sistema de comunicación, el software del clúster, entre hosts para su correcta monitorización, así como un método para abstraer los servicios de un host concreto, cosa que permite que se desplacen entre diversos nodos de manera transparente para la aplicación y los usuarios.

En *Figura 02* se muestra todos los requisitos que debe cumplir un software para clusterizar en alta disponibilidad para que sea fructífero.

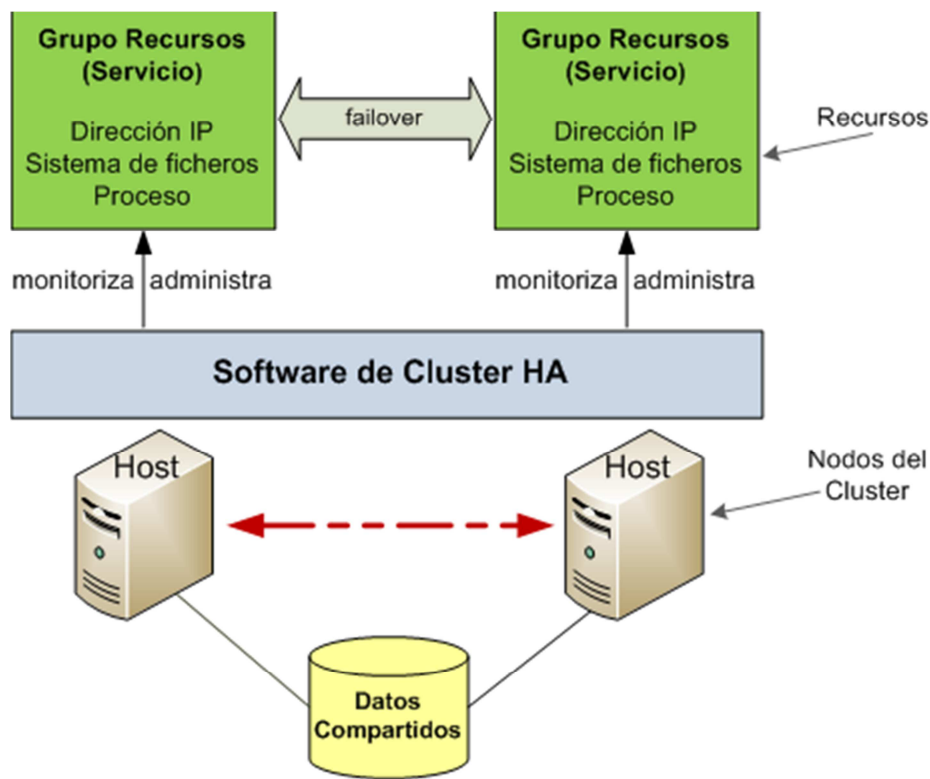


Figura 05: Configuración básica clúster alta disponibilidad

Las configuraciones más comunes en entornos de clústeres de alta disponibilidad son las configuraciones activo/activo y la configuración activo/pasivo.

-Configuración activo/activo

En esta configuración todos los nodos del clúster pueden ejecutar los mismos recursos simultáneamente. Los nodos poseen los mismos recursos y pueden acceder a estos independientemente de los otros nodos del clúster.

Si un nodo falla y deja de estar disponible, sus recursos siguen estando accesibles a través de los otros nodos del clúster.

La principal ventaja de esta configuración es que los nodos en el clúster son más eficientes ya que pueden trabajar todos a la vez. Pero cuando uno de los nodos deja de estar disponible su carga de trabajo pasa a los nodos restantes, esto produce una degradación en el servicio ofrecido.

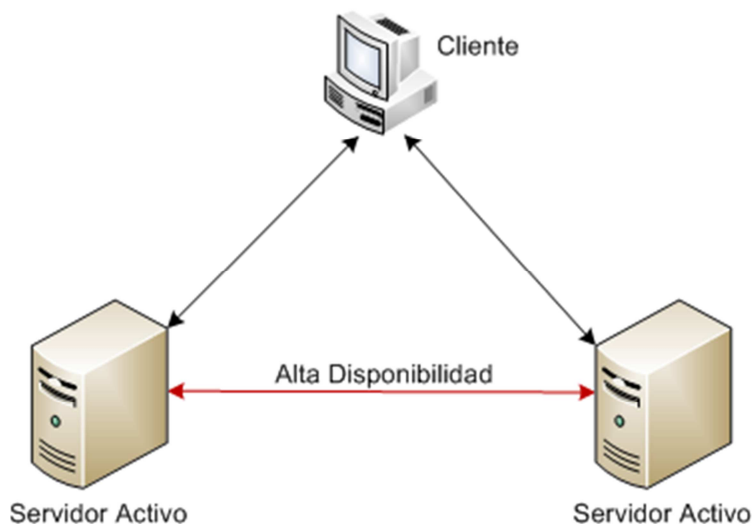


Figura 06: Ejemplo Alta Disponibilidad configuración activo/activo

-Configuración activo/pasivo

Esta configuración consiste en un nodo principal que posee los recursos del clúster y otros nodos secundarios que son capaces de acceder a estos recursos, pero no son capaces de tomar el control de estos hasta que el propietario de los recursos ya no está disponible.

Esta configuración tiene la ventaja que no hay degradación de servicios, el nivel ofrecido es constante, y que estos servicios solo se reinician cuando el nodo principal deja de estar disponible.

La principal desventaja de este sistema con respecto al anterior es que tenemos todos los nodos secundarios ociosos mientras están a la espera del fallo del nodo principal, haciendo que esta solución sea menos eficiente.

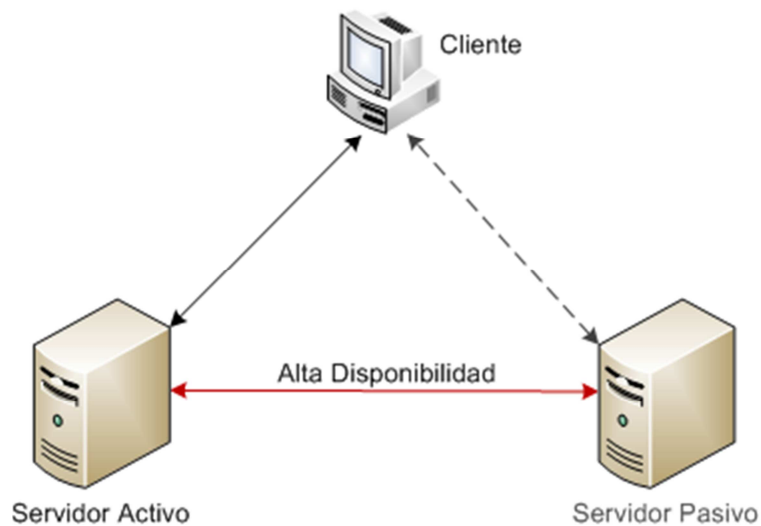


Figura 07: Ejemplo Alta Disponibilidad configuración activo/pasivo

En caso de que un nodo ya no este disponible o dañado o fallido entramos en estado de failover, es decir, en estado de migración de recursos.

El software del clúster reacciona migrando el recurso o grupo de recursos a otro nodo disponible del clúster. De este modo el tiempo de inactividad por el posible fallo es mínimo y así se puede seguir proporcionando el correspondiente servicio.

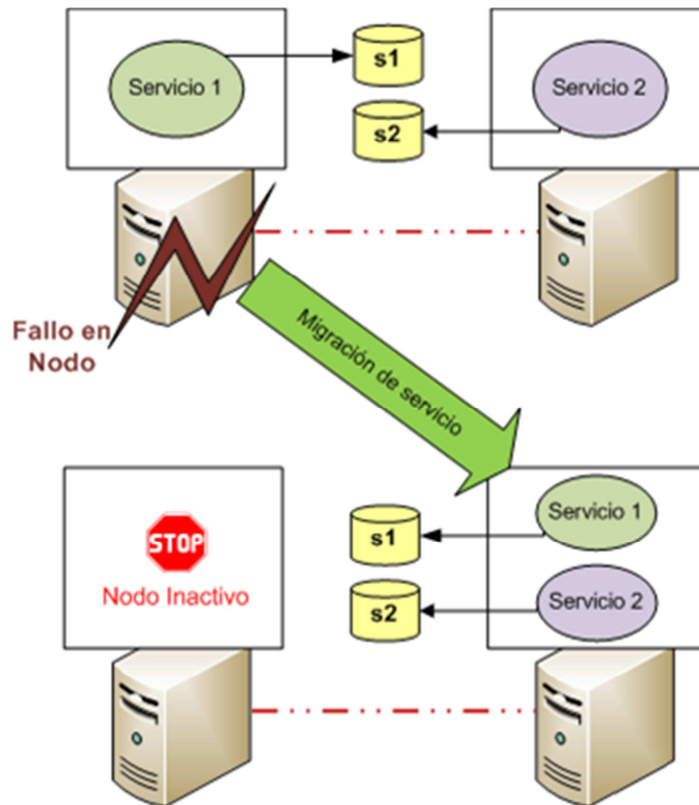


Figura 08: Ejemplo de failover en clúster de Alta Disponibilidad

3.3 Análisis clúster Balanceo de carga

Un clúster de balanceo de carga está compuesto de varios nodos que actúan como la cara visible o front end de este. Estos nodos se encargan de repartir las peticiones de servicio recibidas entre los diversos nodos existentes.

Este sistema tiene varias ventajas como la de ampliar la capacidad fácilmente tan sólo sumando nodos al clúster, cosa que hace que tengamos más capacidad para balancear y por tanto mayor capacidad de respuesta frente a peticiones.

A su vez este sistema ofrece robustez puesto que ante la caída de uno de los nodos el servicio de balanceo de carga sigue funcionando mientras queden nodos disponibles.

En nuestro caso el servicio ofrecido es un servidor web, por lo tanto los clientes solicitarán la web mediante una URL y el clúster se encargará de redistribuir a los nodos más ociosos las peticiones, tal y como se muestra en la *Figura 06*

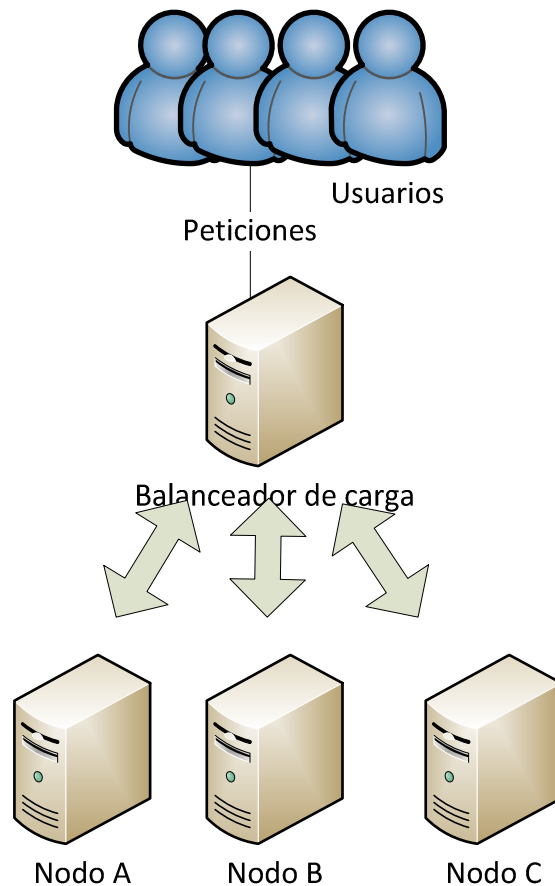


Figura 09: Clúster de balanceo de carga

3.4 Análisis programa empleados

Con el fin de encontrar los programas que más se ajustan a las necesidades de la empresa que quiere el montaje de esta infraestructura se realiza un análisis de los componentes a usar.

Para empezar partimos de la base que los nodos de nuestro clúster deben ser máquinas virtuales sobre diferentes servidor VMware ESXi que ya emplean para virtualizar servidores con múltiples servicios en el tales como el ERP empresarial o Servidor Mail entre otros.

Partiendo de esta base, las máquinas virtuales que emplearemos como nodos del clúster deben ser lo más ligeras posibles en cuanto a uso de recursos se refiere sin que esto nos limite a la hora de crear los clústeres, deben tener un sistema operativo Linux, con licencia gratuita y todos los programas empleados deben ser gratuitos también.

Así pues como sistema operativo base para los nodos barajamos estas alternativas:

Ubuntu Server 10.04 LTS

- Licencia gratuita.
- Basado en Debian.
- Diferencia entre usuario administrador y usuario súper administrador capaz de realizar cambios profundos en el servidor.
- Ofrece pocos componentes a mantener cosa que permite una administración del servidor simple y directa.
- Actualizaciones automáticas, después de la primera configuración se puede elegir si se quiere esta opción, cosa que facilita todavía más la labor de administración.
- En cuanto a la instalación de software se refiere utiliza el comando *apt* con paquetes DEB, cosa que permite empezar, parar, cerrar y gestionar con un simple comando todas las aplicaciones de nuestro servidor.
- Sin interfaz gráfica.
- Permite crear clústeres tal y como nos piden los requisitos.

CentOS 6

- Licencia gratuita
- Basado en RHEL (Red Hat Enterprise Linux) cuenta con las mismas características y funcionalidades pero no es mantenido por Red Hat.
- En cuanto a la instalación de software se utiliza el gestor de paquetes YUM con paquetes RPM
- Permite crear clústeres tal y como nos piden los requisitos.

Dado que ambos sistemas operativos son validos para la creación del clúster, ambos consumen pocos recursos y permiten crear ambos clústeres la decisión de tomar uno u otro se basa plenamente en la experiencia previa en estos entornos.

El entorno CentOS es completamente nuevo y requeriría de un tiempo de aprendizaje mucho más extenso que la solución basada en Ubuntu. Así pues se toma la decisión de usar el sistema operativo base **Ubuntu Server** y la **versión 10.04 LTS** dado que ofrece más soporte al ser una versión LTS (Long Time Supported) garantizando

así que en caso que cualquier problema o inconveniente exista documentación vigente de esta versión en concreto.

Una vez tomada la decisión de usar un sistema operativo determinado debemos estudiar los programas que nos permiten crear los clústeres que queremos y el servidor web que debe albergar en el.

Clúster alta disponibilidad:

Pacemaker es una solución de código abierto que gestiona los recursos de un clúster de alta disponibilidad tanto a nivel de pruebas y entornos pequeños como a nivel de producción y entornos grandes. Permite configuración de nodos Activa/Activa y Activa/Pasiva. Requiere de heartbeat para funcionar. Este programa es también llamado CRM o cluster resource manager, permite también la monitorización de los recursos de este.

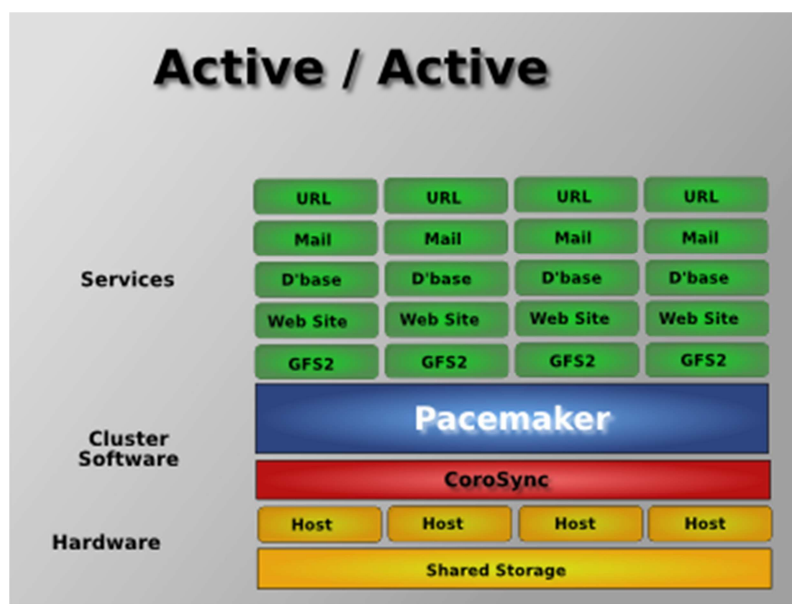


Figura 10: Funcionamiento de pacemaker en configuración activo/activo

DRBD es un bloque cuya función es la de formar un clúster de alta disponibilidad. Este efecto se consigue haciendo un duplicado de todo el bloque a nivel de red. Se puede entender como un raid 1 en red.

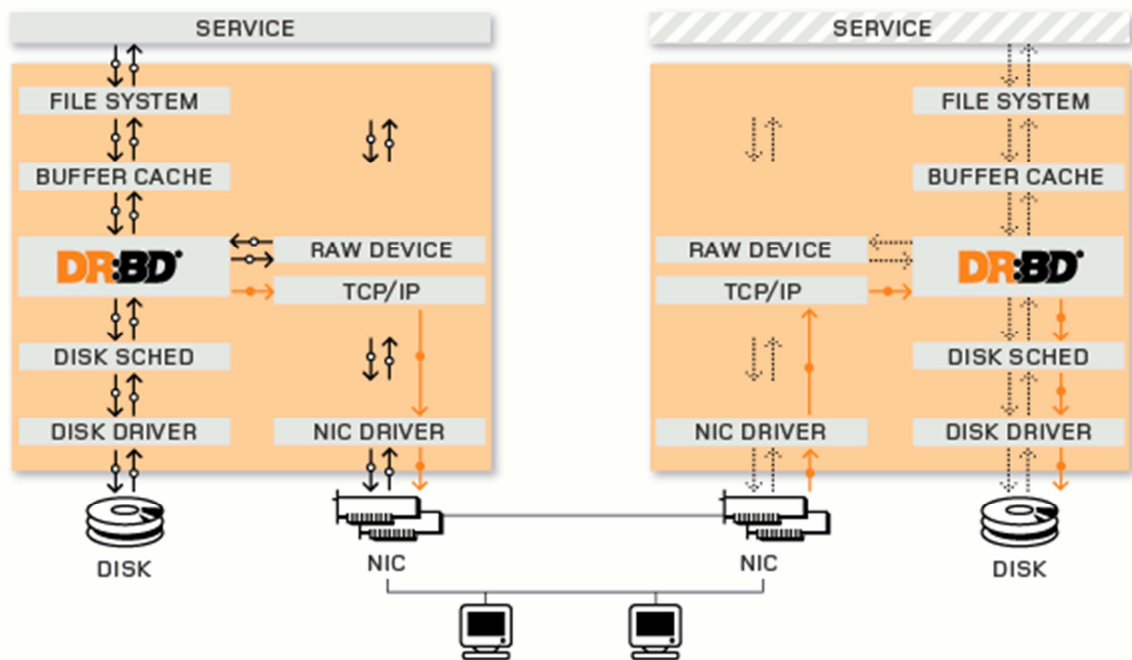


Figura 11: Funcionamiento de DRBD

HeartBeat es un servicio que proporciona infraestructura de clúster a nivel de comunicación y pertinencia al grupo a los nodos. Esto permite a los nodos de la presencia, o ausencia, de servicios o procesos en otros nodos e intercambiar mensajes entre estos.

Para que el servicio de heartbeat funcione se debe combinar con un gestor de clúster el que se encarga del encendido y apagado de los servicios. Pacemaker es el recomendado desde la web oficial del servicio.

Clúster de balanceo de carga:

Nginx es un servidor web proxy inverso, ligero y de alto rendimiento, a su vez es un proxy para protocolos de correo electrónico IMAP y POP3. Es software libre y de código abierto.

En nuestro caso específico usaremos la funcionalidad de proxy inverso con la finalidad de filtrar las peticiones web al nodo menos cargado.

HAProxy es una solución gratis, muy rápido y estable. Ofrece soluciones de alta disponibilidad y balanceo de carga así como proxy para TCP y aplicaciones HTTP.

Su uso queda desestimado ya que su ámbito queda más enfocado a clústeres más grandes y centrados en una arquitectura de hardware muy diferente a las máquinas virtuales, más enfocado a infraestructuras distribuidas con cabinas de discos y al uso en clústeres más enfocados al alto rendimiento como clúster principal o base.

Para el **servidor web** emplearemos el clásico servidor **LAMP** que se puede elegir por defecto en la instalación del sistema operativo Ubuntu.

Otros programas usados:

Openssh server es una implementación libre del paquete de protocolos SSH/SecSSH que provee cifrado para los servicios de res, como acceso remoto o transferencia de archivos remota de un sistema codificado.

En nuestro caso lo usaremos con el fin de que ambos nodos puedan comunicarse entre ellos sin la necesidad de una autenticación previa puesto que crearemos certificados que permitan a estos identificarse en red.

Vsftpd es un protocolo de red para la transferencia de archivos entre Sistemas conectados a una red TCP, basado en la arquitectura del cliente servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos independientemente del sistema operativo usado en cada ordenador.

El uso que le daremos es el de migrar la página web a los nodos de nuestro servidor, esta herramienta nos permite independencia de sistemas operativos así que es muy útil en nuestro caso ya que no se dispone de otras máquinas Linux dentro de la propia empresa.

3.5 Conclusión del análisis

Después de analizar las herramientas necesarias para poder llevar a cabo el proyecto, es decir su diseño y su implantación, se ha concluido que los programas adecuados para llevar a cabo el proyecto son:

Sistema operativo Ubuntu server 10.04 LTS

Programas de Clúster de alta disponibilidad:

- Pacemaker
- Heartbeat

Programa de clúster de balanceo de carga:
Nginx

Otros programas
OpenSSH
VSFTPD

Se han escogido porque cumplen con los requisitos de código libre, ligereza de recursos y por las funcionalidades que estos ofrecen.

4. Diseño

4.1 Introducción

En este capítulo se diseñará el prototipo para la implementación propuesta de este proyecto.

Inicialmente se diseñará la infraestructura física del clúster, así como la definición de los recursos de las máquinas necesarias para la implementación y el soporte de este.

A continuación se diseñará la infraestructura virtual cuyos nodos, o máquinas virtuales, formarán parte de del clúster.

Tanto del diseño físico como del diseño virtual se propondrán alternativas para solucionar posibles problemas en su implantación. A su vez se definirán una serie de pruebas que debe ser capaz de soportar el diseño final.

4.2 Diseño de la infraestructura física

En este apartado se diseñará cual será la mejor implementación de los clústeres basándonos en los recursos existentes en la empresa. Según las características que se han definido en el capítulo de análisis se usarán máquinas virtuales con una nueva instalación del sistema operativo Ubuntu 10.04 LTS y se trabajará sobre estos con unos programas con el fin de hacer viable la creación del clúster.

El resultado esperado es crear un conjunto de máquinas virtuales, llamadas nodos, que funcionen entre sí de manera que actúen como su de un servidor web se tratase pero que pudiesen interaccionar entre ellas con el fin de que el servicio ofrecido no dejase de proporcionarse en el momento en que uno de estos nodos dejase de funcionar. A su vez deben ser capaces de balancear la carga entre estos distintos nodos.

Esta implementación nos permite solucionar problemas derivados de malas configuraciones o caídas de servidores y pueden mostrar de manera inmediata la información en cualquier de los nodos dado que la configuración de estos en el clúster de alta disponibilidad es de activo/activo.

Inicialmente el clúster estará formado por dos nodos ya que no se disponen de más servidores conectados a un SAI de larga duración, sistema de alimentación ininterrumpida, que garanticen el funcionamiento 24 horas 7 días a la semana del servidor. Lo que no debe quitarle capacidad para ser escalable en un futuro.

Estos servidores a su vez están dotados de un sistema de raid 0 por hardware, sistema de discos redundantes por el cual toda la información que se graba en el disco se copia a otro de manera inmediata para protegerlo contra fallos, que ofrecen, a través de una alta disponibilidad física, seguridad en caso de fallo de hardware.

La red de la empresa es una fast ethernet 100Mbps con una conexión a internet mediante una máquina virtual que realiza las función de firewall y que conecta nuestra red 192.168.12.0/24 con el enrutador de la línea adsl.

Tanto el enrutador como el rack, que no es más que un soporte metálico destinado a alojar equipamiento electrónico, en nuestro caso de enrutadores de res llamados habitualmente switches, que conectan los servidores al enrutador, están debidamente protegidos contra caídas de electricidad con un SAI de larga duración.

Así pues la red que albergará nuestro clúster tendrá el siguiente aspecto:

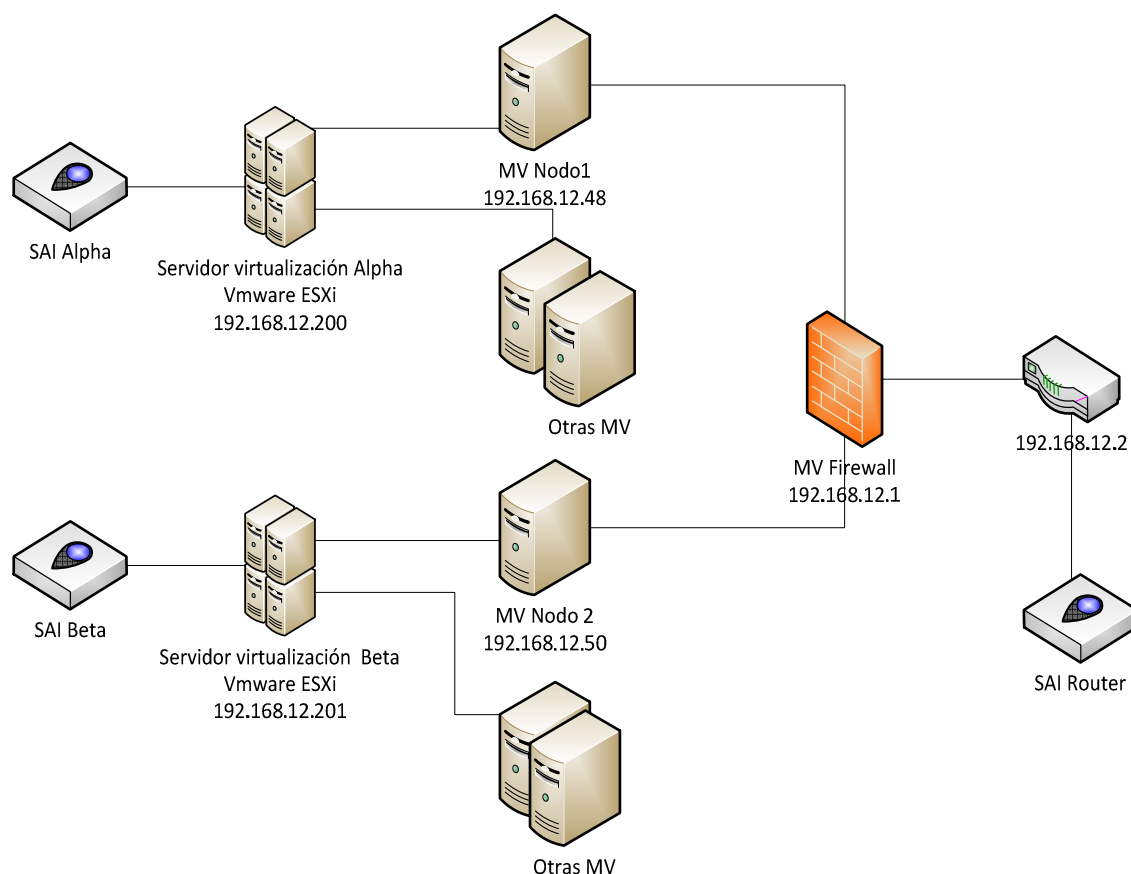


Figura 12: Diseño de la infraestructura física

Una vez definida la red que usaremos definiremos los nodos, sus características y los recursos empleados en cada uno de ellos.

Nodo	Recursos
Ha1-lb IP:192.168.12.48	1 Gb de RAM 16 Gb de disco duro 1 núcleo a 2,5 Ghz 1 tarjeta de red independiente
Ha2-lb IP:192.168.12.50	1 Gb de RAM 16 Gb de disco duro 1 núcleo a 2,5 Ghz 1 tarjeta de red independiente

Tabla14: Definición de nodos

Además cabe explicar que también aremos uso de una tercera IP: 192.168.12.102, una IP Flotante, que se pasarán entre nodos y a la que apunta el nombre del dominio, inicialmente se le asignará al primer nodo, y que se pasarán entre si en caso de falla de uno de los dos nodos.

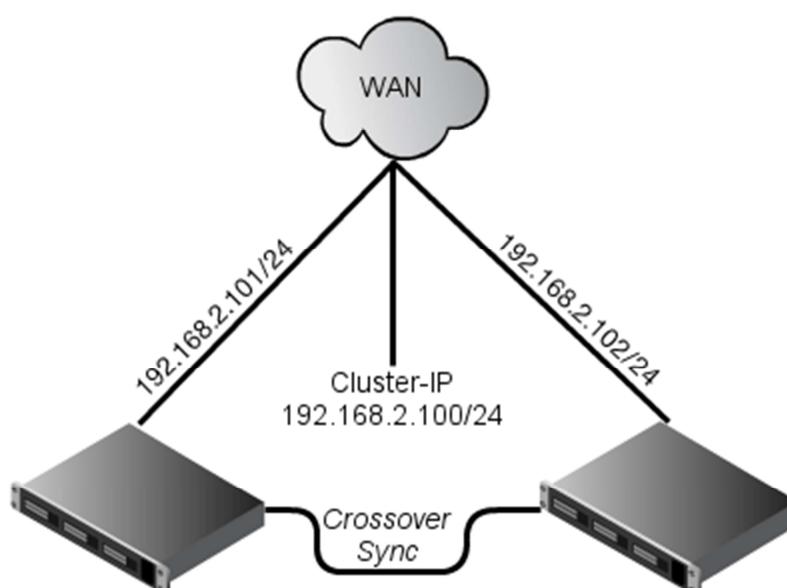


Figura 13: Ejemplo de funcionamiento de IP flotante

4.3 Alternativas a la infraestructura física

Se ha de crear una alternativa por si se detectan problemas a la hora de implementar la infraestructura del clúster.

En caso de falla completa de uno de los servidores se debería migrar el nodo del clúster a otro servidor en otro edificio, también de propiedad de la compañía. Como deben formar parte de la misma red privada, o más comúnmente llamada lan, se debería crear una vpn,

red privada virtual, que nos permite extender nuestra lan de manera segura a otras redes.

Este paso es costoso y la infraestructura obtenida es igual a la anterior con la diferencia que un nodo esta conectado a un servidor en otro edificio pero, lógicamente, esta situado dentro de la misma lan.

En nuestro caso particular no contamos con un sistema de archivos distribuidos o NTS. Esto nos proporcionaría la posibilidad de compartir entre varios ordenadores y máquinas virtuales una porción de disco duro, de manera que un solo cambio en los archivos desde uno de los ordenadores repercutiría en todo el sistema de manera inmediata.

Este seria un cambio notable en caso que nuestra web tuviese contenido dinámico y muchos cambios a diario o semanalmente. Pero dado que esta web es de carácter estático y no tiene previsto ningún cambio el copiar el contenido completo de esta entre los diferentes nodos es una opción aceptable basándonos en la navaja de Ockhan.

En caso de querer añadir más nodos al clúster tan sólo bastaría con clonar los nodos ya existentes y configurarlos, cambiarle la IP, el nombre y cambiar parámetros dentro de los programas que se encargan de hacer clústeres para hacerlos parte de este.

4.4 Diseño de la infraestructura virtual

Una vez tengamos la infraestructura montada y funcionando se ha de diseñar el sistema virtual que nos permitirá realizar el clúster, monitorizarlo y hacerle las pruebas que se describirán más adelante en este capítulo.

El diseño virtual que se propone es crear sobre una instalación limpia de un Ubuntu 10.04 LTS todos los servicios necesarios para llevar a cabo el proyecto, es decir, la instalación de un LAMP, el heartbeat, el pacemaker, el nginx así como el OpenSSH server y el VSFTPD. Este diseño se debe repetir en función del número de nodos que se quieran crear.

Dado que la instalación del servidor web LAMP es el primero en la instalación, es una opción en la instalación del sistema operativo, mostraremos su funcionamiento primero. El servidor Linux Ubuntu proporciona los servicios Apache, encargado del servidor web, y el servicio de MySql, encargado de las bases de datos usadas, el servidor apache aparte de interpretar los lenguajes clásicos de programación web es capaz de interpretar código PHP.

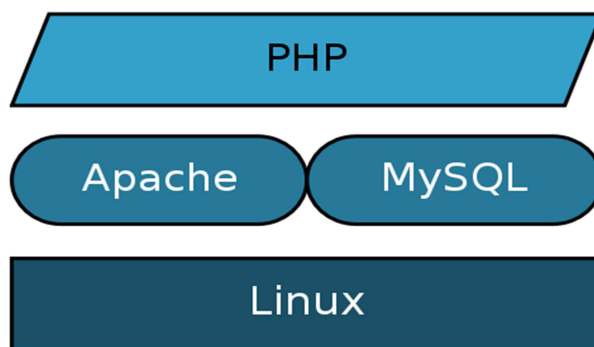


Figura 14: Funcionamiento LAMP

La siguiente figura muestra el funcionamiento del servicio heartbeat en respecto al clúster de alta disponibilidad.

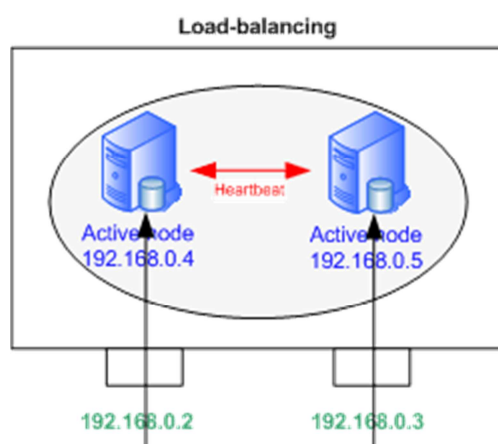


Figura 15: Ejemplo de funcionamiento heartbeat en modo de balanceo de carga

Existen otras configuraciones pero basándonos en el uso que le daremos esta es nuestra opción principal.

El funcionamiento del servicio pacemaker elegida es el de la configuración activa/activa que explicamos en el apartado de análisis de programas empleados también lo usaremos para monitorizar los recursos.

En cuanto a nginx se refiere usaremos la configuración de distribución round robin de las peticiones HTTP entrantes a nuestros servidores web, que en nuestro caso serán los mismos que ofrecen todos los demás servicios.

Tanto los servicios de OpenSSH y el de VSFTPD son de funcionamiento muy trivial. Existe un nodo que es servidor y el otro cliente en el caso que el cliente requiera acceder a información del otro nodo. Estos papeles se pueden invertir en cualquier momento si el nodo contrario requiere información del otro.

En cuanto a OpenSSH se deben crear unos certificados de confianza entre los nodos de manera que pueda acceder el uno al otro sin necesidad de contraseña.

4.5 Alternativas a la infraestructura virtual

Una alternativa a la configuración del sistema anteriormente mencionado es la configuración de un clúster de manera activo/pasivo. Para realizar ese cambio deberíamos configurar el servicio pacemaker tal que así:

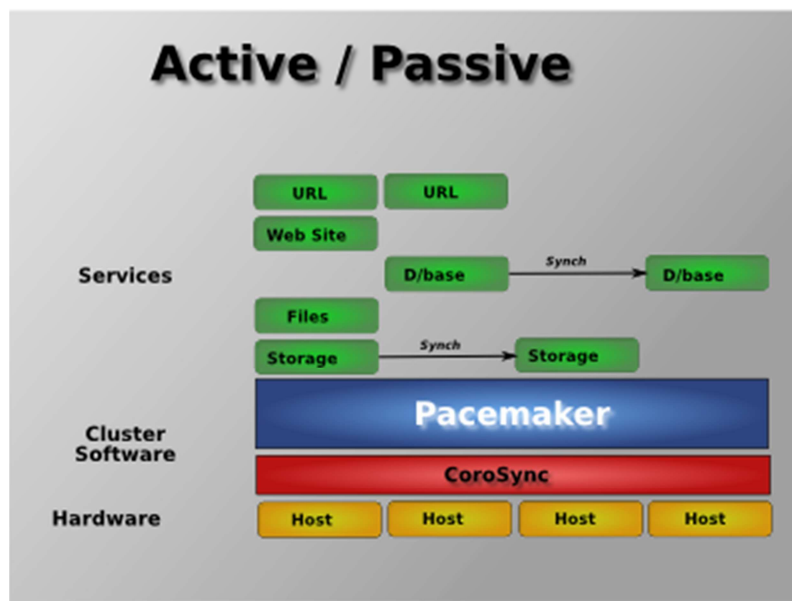


Figura 16: Ejemplo de funcionamiento pacemaker en modo activo/pasivo

Y deberíamos configurar el servicio de heartbeat para su funcionamiento acorde con el funcionamiento del pacemaker. La siguiente figura muestra el cambio de configuración.

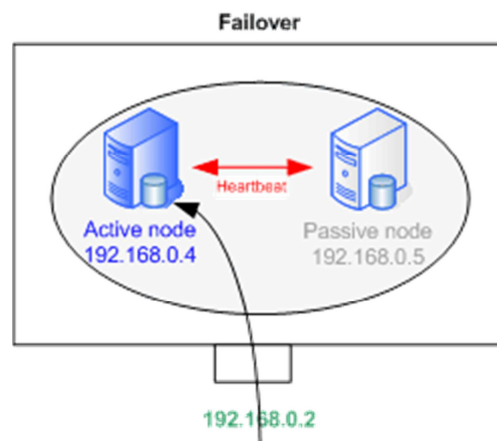


Figura 17: Ejemplo de funcionamiento heartbeat en modo de activo/pasivo

En cuanto a la configuración de LAMP sería exactamente la misma que la anterior dado que no requiere ninguna reconfiguración si el nodo principal es el único encendido y el resto permanecen cerrados. Tan sólo en caso de fallo del nodo principal se activarían el nodo secundario, y así sucesivamente en caso de tener más fallos y mas nodos, y con el sus servicios.

El principal inconveniente de esta alternativa es que no podríamos hacer uso del balanceo de carga puesto que sólo hay un nodo activo a la vez y esto es un inconveniente muy importante. Por lo tanto esta alternativa queda desestimada.

4.6 Conclusión del diseño

Una vez tenemos claro el diseño a usar en la elaboración del proyecto podemos iniciar la fase de implementación. Este diseño inicial permite ver de manera ordenada como se distribuyen los elementos que forman parte del clúster a diferentes niveles.

Los apartados de diseño alternativo serán usados en caso de que no podamos realizar la opción inicial por algún tipo de problema, en el caso de la infraestructura virtual no permitiría el balanceo de carga pero se podrá seguir adelante con la configuración del clúster de alta disponibilidad. Esta última opción de todas maneras es de un impacto brutal en el cumplimiento de los objetivos del proyecto.

5. Implementación

5.1 Introducción

En este capítulo ya podemos empezar a implementar el diseño tanto físico como el virtual descrito en los apartados anteriores.

Una vez implementada la infraestructura, debidamente configurada y montada sobre los servidores, deberá soportar las pruebas descritas con anterioridad para comprobar tanto su funcionalidad como su robustez.

5.2 Instalación y configuración del clúster

El primer paso es descargar el sistema operativo base de los nodos, en este caso la distribución Ubuntu Server 10.04 LTS. Se ha de instalar y configurar de manera idéntica en los dos nodos.

Creemos así dos máquinas virtuales en los dos servidores que tenemos en la empresa con los recursos acordados en la sección de diseño.

Inicialmente nos pregunta el instalador a cerca del teclado y la distribución horaria, en nuestro caso lo ajustaremos a Español. El nombre del nodo, usuario y contraseña que nos pida será usado en la gestión del nodo así que debemos elegir cuidadosamente los nombres.

Para el primer nodo se ha elegido el nombre de ha1-lb y para el segundo ha2-lb.

En el apartado de la selección de método de partición elegimos: Guided –use entire disk and set up LVM. Esto formateará el disco y creará 3 particiones, una de root, una de swap y otra de disco, esta arquitectura permitiría, si se requiriese, compartir la parte de disco en la que albergamos la web entre varios nodos a través de algún método de compartición como el DRBD explicado anteriormente. Por ahora sólo dejaremos esta arquitectura pensando en una posible ampliación en un futuro.

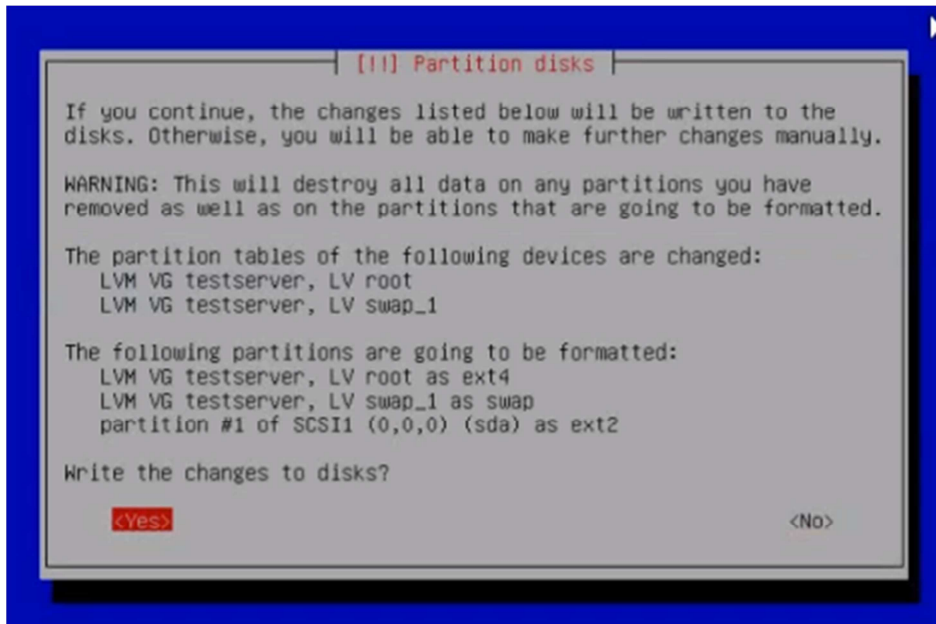


Figura 18: Configuración particiones de disco Ubuntu

En cuanto a actualizaciones se refiere, marcaremos que no deseamos ningún tipo de actualización automática puesto que podría afectar al funcionamiento del sistema.

A la hora de seleccionar el software a instalar marcaremos la opción de LAMP server, así nos instalará todos los componentes anteriormente explicados necesarios para funcionar como servidor web. También podemos seleccionar la opción del OpenSSH server, como lo necesitamos también seleccionamos esta opción.



Figura 19: Instalación LAMP server & OpenSSH Server

A partir de este punto nos preguntará si deseamos instalar GRUB, que es el gestor de arranque de Ubuntu, aceptamos se instalarán

algunos componentes más y ya tenemos la instalación personalizada acabada.

Una vez tenemos los 2 nodos creados con el sistema operativo instalado debemos disponer de varios elementos:

- 2 IPs privadas para asignar a los nodos
- 1 IP pública flotante que compartirán entre si
- Un nombre de domino para la web

Ahora asignamos IPs estáticas a ambos nodos de manera que evitemos el cambio de IP que asigna de manera aleatoria el servidor DHCP. Modificamos así pues el archivo `/etc/network/interfaces` par asignarle su IP fija tal como se muestra a continuación.

```
auto lo
iface lo inet loopback

auto eth0:0
iface eth0:0 inet static
address 192.168.12.48
netmask 255.255.255.0
```

Figura 20: Configuración IP fija nodo ha1-lb

Con los pasos anteriores claros procedemos a elegir un nodo como principal, ha1-lb, y modificamos el archivo `/etc/host` para que reconozca el nombre de todos los nodos en el clúster en vez de escribir por IP.

```
127.0.0.1      localhost.localdomain    localhost
192.168.12.48  ha1-lb
192.168.12.50  ha2-lb
```

Figura 21: Configuración de nodo principal

Procedemos a repetir este proceso en los demás nodos del clúster para que se conozcan entre ellos a través de su nombre. De esta manera consigues una configuración mucho más humana y comprensible.

Una vez tenemos los host configurados debemos crear los certificados para que se reconozcan entre si los nodos sin necesidad de usuario y contraseña. Esto lo realizaremos a través del servicio ssh que nos proporciona esta capacidad. Creamos dos certificados, uno para la autenticación del nodo ha1-lb a ha2-lb y otro de ha2-lb a ha1-lb.

Aquí el código empleado para la creación y el traspaso de un nodo al otro.

```
ssh-keygen -t rsa
scp ~/.ssh/id_rsa.pub root@ha2-lb:/root/ha1_key.pub
ssh root@ha2-lb "ssh-keygen -t rsa"
ssh root@ha2-lb "echo \cat ~/ha1_key.pub\` >>
~/ssh/authorized_keys2"
ssh root@ha2-lb "rm ~/ha1_key.pub"
scp root@ha2-lb:/root/.ssh/id_rsa.pub /root
cat ~/id_rsa.pub >> ~/ssh/authorized_keys2
rm ~/id_rsa.pub

scp /etc/ssh/ssh_host* root@ha2-lb:/etc/ssh/
rm ~/.ssh/known_hosts
ssh root@ha2-lb "/etc/init.d/ssh restart"

scp /etc/hosts root@ha2-lb:/etc/hosts
echo "ha1-lb" > /etc/hostname
hostname -F /etc/hostname
ssh root@ha2-lb "echo \"ha2-lb\" > /etc/hostname"
ssh root@ha2-lb "hostname -F /etc/hostname"
```

Figura 22: Configuración certificados SSH

En este punto ya disponemos de los elementos necesarios para la comunicación eficiente de los nodos y procedemos a la instalación de los programas pacemaker, heartbeat y nginx.

Una vez descargados e instalados procedemos a la configuración de estos. Empezaremos por el heartbeat.

Creamos un archivo llamado ha.cf dentro de la carpeta */etc/heartbeat* el cual determinará los nodo implicados en el servicio así como parámetros técnicos y los nombres de los nodos del clúster.

```
logfacility daemon
keepalive 2
deadtime 15
warntime 5
initdead 120
udpport 694
ucast eth0 192.168.12.50
auto_failback on
node ha1-lb
node ha2-lb
use_logd yes
crm respawn
```

Figura 23: Configuración heartbeat

Repetimos el proceso en el otro nodo cambiando la IP por la del nodo principal. Ahora creamos los claves de autenticación entre los nodos,

en el nodo principal creamos el archivo authkeys dentro de la carpeta `/etc/heartbeat` y se le asigna una contraseña no trivial. En nuestro caso "q1qT7ce9!".

```
auth 1
1 sha1 q1qT7ce0!.
```

Figura 24: Configuración claves heartbeat

Ahora le asignamos los permisos correspondientes de lectura y escritura, se lo entregamos al segundo nodo y ponemos en marcha el servicio heartbeat.

```
chmod 600 /etc/ha.d/authkeys
/etc/init.d/heartbeat start
scp /etc/ha.d/authkeys root@ha2-lb:/etc/ha.d/
ssh root@ha2-lb "chmod 600 /etc/ha.d/authkeys"
ssh root@ha2-lb "/etc/init.d/heartbeat start"
```

Figura 25: Asignación de permisos certificado y transferencia a nodo secundario heartbeat

Con esto concluimos la configuración del heartbeat e iniciamos la configuración del balanceador de carga nginx.

El archivo que configura el servicio nginx se encuentra en la siguiente ruta `/etc/nginx/nginx.conf` . Lo modificamos para que distribuya la carga de peticiones entrantes entre los servidores web que tenemos a nuestra disposición, en este caso los dos nodos del clúster. Quitamos todos los demás servicios que nos ofrece nginx tales como el de servidor web o proxy y nos centramos en el de reverse proxy que es el que nos permite realizar la función deseada. La configuración final es la siguiente.

```

user www-data;
worker_processes 1;

error_log /var/log/nginx/error.log;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    sendfile on;
    keepalive_timeout 65;
    tcp_nodelay on;
    gzip on;
    gzip_disable "MSIE [1-6]\.(?!.*SV1)";

    upstream webservers {
        server 192.168.12.48 max_fails=3 fail_timeout=5s;
        server 192.168.12.50 max_fails=3 fail_timeout=5s;
    }

    server {
        server_name example.com www.example.com;
        location / {
            proxy_pass http://webservers;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_next_upstream timeout;
        }
    }
}

```

Figura 26: Configuración Nginx

Copiamos el archivo tal cual al segundo nodo en su carpeta correspondiente vía ssh y terminamos con la configuración de nginx.

Ahora tan sólo queda editar el crm que no es más que las siglas de gestor de recursos del clúster en inglés. De esta parte se encarga el pacemaker, aunque no lo mencionemos como la configuración específica de este.

Al editar el archivo de configuración del crm nos encontramos que es un archivo que tan sólo hemos de añadir información para que nos muestre lo que deseamos a la hora de monitorizar el clúster.

Al acceder al archivo de configuración a través de un comando propio de pacemaker "*crm configure edit*" debemos dotarle de la IP flotante pública que será usada para acceder a la web, en nuestro caso 99.99.99.99, evidentemente no es una IP pública válida ni la verdadera pero por petición de la empresa en cuestión se usara esta sólo a modo demostrativo. Añadimos las siguientes líneas entre secciones nodo del documento.

```

primitive ipl ocf:heartbeat:IPaddr2 \
    params ip="99.99.99.99" nic="eth0:1" \
    op monitor interval="5s"
primitive iplarp ocf:heartbeat:SendArp \
    params ip="99.99.99.99" nic="eth0:1"
primitive nginx ocf:heartbeat:anything \
    params \
        binfile="/usr/sbin/nginx" \
        cmdline_options="-c /etc/nginx/nginx.conf"
group HAServices ipl iplarp nginx \
    meta target-role="Started"
order ip-before-arp mandatory: ipl:start iplarp:start
order ip-before-nginx mandatory: ipl:start nginx:start

```

Figura 27: Configuración CRM

Añadimos en la sección de propiedades el parámetro `expected-quorum-votes` y lo asignamos a 1 puesto que el clúster tan sólo dispone de dos nodos, es un cambio mínimo pero sin él no funciona en absoluto.

```

property $id="cib-bootstrap-options" \
    dc-version="1.0.8-042548a451fce8400660f6031f4da6f0223dd5dd" \
    cluster-infrastructure="Heartbeat" \
    expected-quorum-votes="1" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"

```

Figura 28: Configuración CRM II

Una vez configurados todos estos puntos ya tenemos el clúster listo para funcionar, así pues copiamos via ssh nuestra web a la carpeta `/var/www` que es la que contiene los archivos a ser ejecutados por el servidor web apache ya instalado e iniciamos el servicio en ambos nodos. Ya tenemos el clúster montado y listo para hacer las pruebas pertinentes.

5.3 Conclusiones de la implementación

En este capítulo hemos visto el proceso de implementación así como la configuración de cada uno de los programas usados para una clonación exacta del sistema empleado o una futura expansión de los nodos del clúster.

La configuración final de los nodos ha seguido un proceso minucioso. Esto es tan sólo el resultado por el que los nodos proporcionan el comportamiento esperado, detrás de ello hay muchas horas de pruebas y algunas de desesperación.

Se ha visto que para el correcto funcionamiento de los nodos deben de ser configurados de manera clónica ya que esto afectaría de manera crítica al sistema.

Los programas empleados han funcionado de manera esperada y han creado el clúster deseado. Gracias sobretodo a una fase de diseño bien estructurada y una fase de análisis muy acertada.

6. Pruebas

6.1 Introducción

En este apartado de la memoria del proyecto elaboramos una lista de pruebas que realizaremos al proyecto con el fin de comprobar su correcto funcionamiento a la vez que intentar encontrar posibles mejoras.

No todas las pruebas deben ser obligatoriamente superadas. Resulta evidente que algunas, sobretodo las que se basan en el funcionamiento básico del clúster, sí deben cumplir nuestras expectativas pero otras de más exigentes pueden sentar las bases para una mejora en el clúster, tanto a nivel de diseño como de implementación como de uso de programas.

6.2 Pruebas

Se han diseñado diferentes pruebas para analizar el comportamiento deseado del clúster para poder refinarlo viendo sus carencias.

A la infraestructura se le aplicarán esta batería de pruebas:

- Comportamiento en caso de falla de uno de los nodos.
- La monitorización debe ser correcta en todo momento.
- Se probará la escalabilidad de la infraestructura física. El poder aumentar y disminuir los nodos y que siga con el funcionamiento deseado.
- Pruebas de balanceo de carga.
- Seguridad a la hora de acceder a los nodos.
- Parada de los servicios LAMP sin caída de nodo.

6.3 Resultados Obtenidos

Se han efectuado diferentes pruebas para analizar el comportamiento y ver los posibles errores e intentar solucionarlos. A continuación explicaremos el resultado obtenido en cada una de ellas.

- **Comportamiento en caso de falla de uno de los nodos.**

Para comprobar esta prueba realizaremos el cambio de un nodo a otro usando el comando `crm node standby ha1-lb` y el comando `crm node online ha1-lb` esto apaga y enciende el nodo ha1-lb de manera lógica y podemos apreciar que los servicios ofrecidos, servidor web y crm de clúster, cambian de un nodo al otro de manera correcta.

En caso de desconexión total del nodo, cerrado del servidor de manera busca fruto de una desconexión del sai o simplemente un reinicio inesperado de este, el resultado también fue positivo y vimos que cambiaron los servicios de un nodo a otro de manera correcta.

- **La monitorización debe ser correcta en todo momento.**

La monitorización es accesible desde cualquiera de los dos nodos, basta con escribir el comando `crm_mon` y nos aparecerá el estado de los nodos en tiempo real. El estado del clúster se muestra en una pantalla de los nodos sin más florituras.

```
=====
Last updated: Mon Sep 17 11:59:07 2012
Stack: Heartbeat
Current DC: ha2-lb (fff7f31a-4f03-440b-bf38-59794fd33593) - partition with quorum
Version: 1.0.8-042548a451fce8400660f6031f4da6f0223dd5dd
2 Nodes configured, 1 expected votes
1 Resources configured.
=====

Online: [ ha1-lb ha2-lb ]

Resource Group: HAServices
  ip1      (ocf::heartbeat:IPaddr2):      Started ha1-lb
  ip1arp   (ocf::heartbeat:SendArp):       Started ha1-lb
  nginx    (ocf::heartbeat:anything):      Started ha1-lb
-
```

Figura 29: Monitorización del clúster

- **Se probará la escalabilidad de la infraestructura física. El poder aumentar y disminuir los nodos y que siga con el funcionamiento deseado.**

Para realizar esta prueba debemos crear un nuevo nodo siguiendo los pasos anteriormente mencionados. Este debe ser idéntico a los otros dos en cuanto a configuración de la máquina virtual y clonar la configuración en la instalación del sistema operativo.

Se le asigna la IP fija y se crea el certificado de seguridad ssh para que se comunice con los demás nodos sin contraseña. Procedemos a instalar los programas pacemaker, heartbeat y nginx.

En el heartbeat clonamos la instalación pero modificamos la IP del segundo nodo para que apunte al tercero y la IP de la configuración del tercer nodo para que apunte al primero. Procedemos a instalar los certificados y acabamos con la configuración del heartbeat.

En cuanto al nginx calcamos la configuración con la excepción que debemos añadir un tercer servidor web en los tres documentos.

Finalmente a la hora de configurar el crm quitamos la línea de *"expected-quorum-votes"* ya que tenemos más de 2 nodos.

Una vez realizado este procedimiento realizamos pruebas para determinar si se gestionaban de manera correcta los recursos a la hora de cerrar uno de los nodos.

Los resultados fueron satisfactorios, la monitorización de los nodos era correcta y en cuanto los nodos ha1-lb y ha2-lb se cerraban el servicio era proporcionado por el tercer nodo.

- **Pruebas de balanceo de carga.**

Para realizar estas pruebas ejecutaremos un script que nos mostrará el número de conexiones que hay en el puerto en el que el servicio apache esta ejecutándose.

El comando en particular es *"netstat -plan | grep :80 | wc -l"* ejecutamos el comando en ambos nodos y accedemos a través de varios ordenadores a la web para generar tráfico y esperamos ver los resultados balanceados correctamente.

Para nuestra satisfacción el resultado fue correcto puesto que de cuatro solicitudes se dividieron en los dos nodos de manera ecuánime.

- **Seguridad a al hora de acceder a los nodos.**

Para acceder a los nodos tan sólo podemos acceder de manera interna y vía ssh. Se excluye cualquier acceso de manera externa. Este protocolo está correctamente protegido a través de una contraseña fuertemente cifrada con números, letras y caracteres alfanuméricos.

En cuanto a la protección de servidor web disponemos de un firewall que bloquea todos los puertos tanto de entrada como de salida a excepción del 80, que es el usado para dar servicio web.

Puesto que tenemos los flancos de acceso protegidos podemos concluir que el sistema de protección de acceso a los nodos es seguro en la medida de la magnitud de nuestro sistema.

- **Parada de los servicios LAMP sin caída de nodo.**

Esta prueba se ha realizado cerrando el servicio apache del nodo principal cuando este estaba activo y comprobando que esté como nodo primario en la monitorización.

Al desactivar el servicio apache del nodo principal deja de dar servicio web y se muestra en el navegador un error de nginx, exactamente el 502 bad Gateway.

Esto demuestra que el clúster todavía necesita de perfeccionamiento, una solución a esta incidencia sería desacoplar el servidor web del clúster para que no se muestre el error de nginx, sustituido por el clásico 404 url not found, e introducir el servicio apache en los servicios a controlar a la hora de monitorizar el clúster en los nodos y actuar en consecuencia a la hora de su parada.

Esta prueba sienta las bases para una mejora casi indispensable a la hora de realizar un sistema más sólido que el actual.

6.4 Conclusiones

Una vez realizadas las pruebas estas nos revelan que el clúster creado cumple con los objetivos fijados pero también se detectan mal funcionamientos. Estos sientan la base para una mejora en el clúster a nivel de configuración de los programas usados y a nivel de diseño.

Estas pruebas nos garantizan que el clúster responde a nuestras expectativas básicas y que se le ha dotado de los medios necesarios para desempeñar la función para la que estaba pensado inicialmente.

7. Conclusiones

7.1 Conclusiones finales

El objetivo de este proyecto es exponer las conclusiones finales sobre el proyecto exponiendo los objetivos alcanzados, las posibles mejoras o ampliaciones realizables basándose en este y una valoración personal.

El desarrollo del proyecto se ha centrado en gran medida a cumplir de forma precisa los objetivos expuestos en el estudio de viabilidad, los cuales han sido alcanzados en su totalidad, puesto que el clúster logra ofrecer alta disponibilidad en el servicio ofrecido y un balanceo de carga de peticiones web entre los nodos existentes.

No obstante no se ha dejado de lado los requisitos de la aplicación, puesto que nos guían a la hora de cumplir los objetivos debido a que muchos forman parte de estos.

Durante el diseño y el análisis se ha ido modificando el proyecto a las funcionalidades ofrecidas por el diseño deseado y las herramientas empleadas en él.

Como resultado final disponemos de un clúster de dos nodos plenamente operativo para su uso en un entorno de producción real en una empresa. Esta solución ha sido satisfactoria tanto para ellos como para mí, puesto que ha supuesto la superación de un reto a nivel personal y profesional.

Tan sólo falta añadir que la planificación se ha visto truncada a último momento puesto que la fecha de entrega real es posterior a la deseada.

7.2 Desviaciones temporales

Durante el transcurso del proyecto se ha visto modificada la planificación de manera sustancial. Algunas modificaciones han sido debidas a una mala planificación inicial, otras a causas personales y otras a unas largas horas de desesperación a la hora de la configuración.

La fecha de entrega se ha visto atrasada, el tiempo de diseño y análisis ha sido reducido, la memoria me ha ocupado más tiempo del estimado y el tiempo de implementación ha sido aumentado considerablemente.

Así pues la planificación inicial se ha visto modificada de la siguiente manera en cuanto a la planificación inicial se refiere.

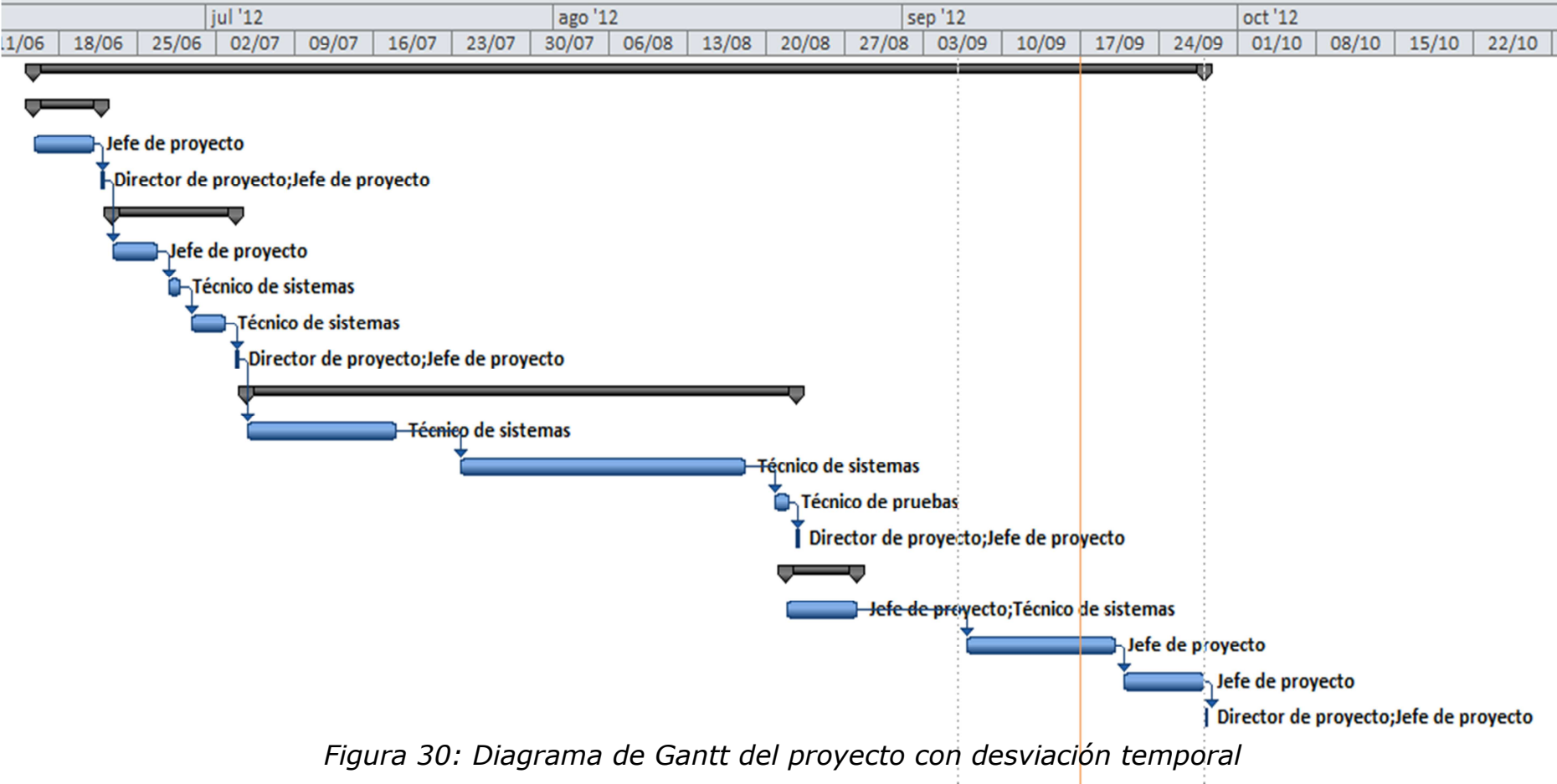


Figura 30: Diagrama de Gantt del proyecto con desviación temporal

7.3 Posibles ampliaciones y modificaciones

Basándonos en el diseño que hemos empleado, he ideado una serie de modificaciones en la infraestructura que podrían mejorar el funcionamiento de esta.

La primera modificación sería crear una serie de servidores web externos a los nodos creados para su balanceo de carga. Eliminar el servidor web de los nodos de alta disponibilidad y balanceo de carga, creando así una serie de servidores web ampliables a nuestra merced o al espacio disponible.

Esta solución aumentaría la cohesión y reduciría el acoplamiento de la infraestructura, a su vez ofrecería la posibilidad de aumentar la capacidad de balanceo de carga y reduciría el número de tareas en los nodos de los clústeres.

La problemática de esta solución viene dada en que estos servidores web deberían estar albergados en un sistema de almacenaje compartido y capaz de sincronizar los cambios de cualquier servidor a los demás en tiempo real. A su vez debe ser capaz de ser resistente, al igual que los nodos del clúster, a cortes de electricidad de algún método.

Aplicando esta solución se puede prever una mejora en la infraestructura así como una mejora en el tiempo de respuesta de peticiones web para un número elevado de usuarios.

7.4 Valoración personal

En cuanto a mis sensaciones a la hora de realizar el proyecto han sido muy satisfactorias, puesto que he podido poner en práctica los conocimientos aprendidos a lo largo de la carrera, los cuales me han ayudado a superar con éxito las dificultades que me han ido surgiendo a la hora tanto de diseñar, como de analizar como de implementar la solución.

Inicialmente me lo tome con muchas ganas e ilusión, estas ganas mermaron a medida que los problemas fueron surgiendo y las horas se convertían en estresantes momentos de incompreensión al ver que no realizaba correctamente las ordenes deseadas. A base de persistencia y ganas de acabar con el trabajo ya empezado se dio con el camino correcto y ahora ya se puede apreciar el satisfactorio resultado final.

Para concluir sólo tengo que añadir que estoy orgulloso con el resultado obtenido y espero que sea un catalizador para mi

trayectoria profesional en la rama que he elegido, ingeniería de sistemas.

8. Bibliografía

A continuación queda detalladas todas las fuentes de información usadas a lo largo del desarrollo del proyecto. Todas las consultas se han realizado en internet. El formato usado es el siguiente:

Título-Autor

Idioma-actualizado-fecha visita - contenido

URL

Clúster (informática)

ES - 09/2012 - 09/2012 - documentación

http://es.wikipedia.org/wiki/Cluster_%28inform%C3%A1tica%29

Linux Clustering

EN - 05/2012-6/2012 - documentación

<http://www.squidoo.com/linux-clustering>

Comparison of clúster software

En - 02/2012 - 09/2012 - Documentación

http://en.wikipedia.org/wiki/Comparison_of_cluster_software

Clúster de alta disponibilidad (HA) con heartbeat - netsultan

ES - 01/2008 - 6/2012 - Documentación & Implementación

<http://netsultan.wordpress.com/2008/01/22/clustering-de-alta-disponibilidadha-con-heartbeat/>

Heartbeat

EN - 01/2010 - 07/2012 - Documentación & diseño

<http://linux-ha.org/wiki/Heartbeat>

Como configurar un agrupamiento de alta disponibilidad con

heartbeat en CentOS 5 - Joel Barrios Dueñas

ES - 11/2011 - 07/2012 - Documentación

<http://www.alcancelibre.org/staticpages/index.php/como-cluster-heartbeat-centos>

Creando un clúster de alta disponibilidad en Microsoft Windows 2008

- Hector Herrero Hermida

ES - 03/2009 - 07/2012 - Documentación

<http://www.bujarra.com/creando-un-cluster-de-alta-disponibilidad-en-microsoft-windows-server-2008/>

Loadbalanced High-Availability Apache Cluster Using Ultramonkey

En - - 07/2012 - Documentación & implementación

<http://www.debianhelp.co.uk/ultramonkey.htm>

Highly Available HTTP Load Balancer – Matthew Cone
EN – 07/2012 – 08/2012 – Documentación, Implementación y diseño
<http://library.linode.com/linux-ha/highly-available-load-balancer-ubuntu-10.04>

HA cluster with DRBD and Heartbeat
EN – 06/2012 – 08/2012 - Documentación
http://wiki.openvz.org/HA_cluster_with_DRBD_and_Heartbeat

Pacemaker
EN – 06/2010 – 09/2012 – Documentación & implementación
<http://www.clusterlabs.org/wiki/Pacemaker>

DRBD
EN - 2011 -09/2012 – Documentación & diseño
<http://www.drbd.org/>

Logical Volume Manager
ES – 04/2012 -09/2012 - Documentación
http://es.wikipedia.org/wiki/Logical_Volume_Manager

SSH login without password
EN – 08/2010 – 08/2012 - Implementación
http://linuxconfig.org/Passwordless_ssh

9. Glosario

Debian-like

Este término lo uso para referirme a todos los sistemas operativos que usan de base el núcleo o kernel usado por el sistema operativo debían en primera instancia. Posteriormente han salido muchas distribuciones Linux que usan ese mismo kernel como Linux mint, Ubuntu y xubuntu entre otros.

Proxy

En nuestro caso nos referimos a un servidor proxy, es decir un servidor que actúa como proxy. Este servidor proxy actúa como intermediario entre una petición determinada y el recurso que ofrece esa petición. Así pues esta máquina es capaz de controlar las peticiones que se realizan sobre otra que ofrece un recurso ya sea con el fin de analizarlas, tratarlas, privarlas, filtrarlas o gestionarlas.

Reverse proxy

El reverse proxy tan sólo es un caso particular de configuración de un proxy. Esta configuración permita proporcionar a una máquina en particular servicios ofrecidos por una tercera máquina de manera transparente, es decir, una máquina no es necesario que ofrezca todos los servicios puesto que la carga de estos se puede dividir entre varias máquinas.

Servidor web

Un servidor web es una máquina que permite proporcionar conexiones bidireccionales, síncronas o asíncronas con un cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. En términos generales es una máquina que ofrece servicios que sustentan una página web a través de un programa que permite esta función.

LAMP

Son las siglas usadas para denominar un servidor web basado en Linux (Linux Apache MySQL PHP) con una base de datos MySQL y servidor web propiamente denominado Apache.

Patrón/Distribución Round Robin

Consiste en una distribución de carga entre múltiples elementos a través de turnos rotativos con un límite de tiempo máximo para cada uno de ellos previamente estipulado

Programas

Son varias instrucciones que una vez ejecutadas realizan una o varias tareas en la computadora.

Servicios

Son programas que una vez instalados quedan latentes en el sistema operativo y forman parte de las tareas y/o funciones que desempeña este.

10. Anexo

En este apartado se explicará como se ha instalado los programas necesarios para la realización del proyecto.

El servidor de máquinas virtuales ESXi es un sistema de creación muy intuitivo de estas máquinas. Para acceder a estas usamos el programa de VMware Vsphere client, de licencia gratuita, que nos permite gestionar, crear y destruir todas la máquinas virtuales así como ver el uso y la carga de trabajo de estas entre otras funcionalidades.

En nuestro caso la creación de la máquina virtual es simple y los parámetros usados están señalados en la siguiente figura:

The screenshot displays the configuration interface for a virtual machine named 'dhcppc13'. It is divided into several sections:

- General:** Guest OS: Ubuntu Linux (64-bit), VM Version: 8, CPU: 1 vCPU, Memory: 1024 MB, Memory Overhead: 67,18 MB, VMware Tools: Not running (Not installed), IP Addresses: (empty), DNS Name: (empty), State: Powered Off, Host: dhcppc13, Active Tasks: (empty), vSphere HA Protection: N/A.
- Resources:** Consumed Host CPU: (empty), Consumed Host Memory: (empty), Active Guest Memory: 20,00 MB (with a Refresh Storage Usage link), Provisioned Storage: 17,11 GB, Not-shared Storage: 16,00 GB, Used Storage: 16,00 GB.
- Storage:** A table showing 'datastore1' as a Non-SSD drive with a capacity of 227,75 GB.
- Network:** A table showing 'VM Network' connected to a 'Standard port group'.
- Commands:** Includes 'Power On' and 'Edit Settings' buttons.
- Annotations:** A section for notes with an 'Edit' button.

Instalamos el sistema operativo elegido, Ubuntu 10.04 LTS de la manera descrita en la sección de implementación. Exactamente como se ha descrito, si no es así en todos los nodos de manera clónica cabe la posibilidad que surjan errores aparentemente inexplicables pero que son debidos a la instalación de este.

Para acceder a los nodos creados se puede hacer vía ssh, puesto que ya hemos instalado el servidor ssh a la hora de instalar el sistema operativo o a través del vSphere client.

La instalación de los programas heartbeat, pacemaker y nginx puede realizarse antes incluso de la asignación de las IPs fijas e incluso antes de la modificación del archivo `/etc/hosts`. No obstante mi recomendación es realizar la instalación post modificación de los archivos anteriormente mencionados basándome en que así evitamos despistes a la hora de configurar el nodo.

Antes de empezar usaremos un comando muy útil cuando trabajamos en comandos en Ubuntu, este te proporciona permisos de super usuario a todos los comandos que uses sin necesidad de ir repitiendo al inicio de los comandos en cada línea.

El comando en cuestión es:

Sudo SU

Nos preguntará la contraseña y ya somos súper usuario con todos los privilegios disponibles.

Empezaremos pues por actualizar y upgradear a la última versión disponible los programas y repositorios del sistema del nodo principal con el comando:

apt-get update
apt-get upgrade -y

En el comando de upgrade añadimos la extensión `-y` para upgradear todos los programas de manera automática. Puesto que cuando nos pregunta si queremos upgradear por defecto seleccionamos la respuesta YES.

Una vez tenemos el entorno preparado instalamos a través del siguiente comando.

apt-get install -y heartbeat pacemaker nginx

Este comando aglutina los comandos de instalación de los tres programas en uno de solo, además ponemos la extensión `-Y` para que acepte por defecto todas las preguntas a la hora de la instalación.

Una vez instalados pararemos el servicio nginx para borrar el archivo de actualización que tiene para evitar que se actualice sin previo aviso y pueda modificar el comportamiento deseado de manera severa, a su vez el servicio de pacemaker es el que se encarga de encender y apagar el servicio nginx así que no tiene sentido este archivo. Los comandos a ejecutar son los siguientes:

```
/etc/init.d/nginx stop  
Update-rc.d -f nginx remove
```

Ya tenemos los programas instalados y somos capaces de configurarlos de manera que funcionen de la manera deseada con la configuración explicada en la sección de implementación.

Para replicar la instalación del nodo principal en los demás nodos podemos usar el protocolo ssh puesto que hemos creado un certificado que nos permite el acceso a los demás nodos de manera limpia, sin contraseñas y con permisos de administrador.

Los comandos a usar son los mismos que hemos empleado con anterioridad pero con la diferencia que los iniciaremos con el uso del protocolo ssh e identificando el usuario deseado del nodo. Los comandos siguen a continuación.

```
ssh root@ha2-lb "apt-get update"  
ssh root@ha2-lb "apt-get upgrade -y"  
ssh root@ha2-lb "apt-get install -y heartbeat pacemaker nginx"  
ssh root@ha2-lb "/etc/init.d/nginx stop"  
ssh root@ha2-lb "update-rc.d -f nginx remove"
```

Esta metodología también puede ser usada en la instalación en otros nodos una vez creado el certificado de confianza ssh.

Alfred Gutiérrez Sanmiguel
Sabadell, Septiembre del 2012