

ARQUITECTURA DE ALTA DISPONIBILIDAD PARA LA PLATAFORMA LMS
DE MAAT-G

ESTEBAN DE JESUS HERNÁNDEZ BARRAGAN

UNIVERSIDAD AUTONOMA DE BUCARAMANGA
FACULTAD DE INGENIERIA DE SISTEMAS
MAESTRIA EN SOFTWARE LIBRE
BOGOTA
2010

ARQUITECTURA DE ALTA DISPONIBILIDAD PARA LA PLATAFORMA LMS
DE MAAT-G

ESTEBAN DE JESUS HERNÁNDEZ BARRAGAN U00046815

Tesis para optar el título de
Magíster en Software Libre

Director de Tesis

EDUARDO CARRILLO ZAMBRANO

Doctor en Tecnología de la Información y las Comunicaciones

UNIVERSIDAD AUTONOMA DE BUCARAMANGA

FACULTAD DE INGENIERIA DE SISTEMAS

MAESTRIA EN SOFTWARE LIBRE

BOGOTA

2010

Nota de aceptación:

Firma del presidente del Jurado

Firma del jurado

Firma del jurado

Ciudad y fecha (día, mes, año)

DEDICATORIA

A Dios el creador y arquitecto perfecto de tan indescriptible belleza llamada universo.

A mi esposa Azeneth, por su su paciencia y apoyo aún cuando mi única respuesta a sus solicitudes de esparcimiento era “estoy estudiando”.

A mis padres, Ofelia y Eulises, porque me infundieron desde niño el deseo de lucha aún cuando las circunstancias sean difíciles y colocaron en mí la semilla de una fe verdadera.

A Mr, Jym Bell Smyth, por su ejemplo y enseñanza de lo correcto y verdadero. Gracias a su fundamento he podido continuar aún en medio de ambientes hostiles.

AGRADECIMIENTOS

Quiero expresar especial reconocimientos y agradecimientos a las personas e instituciones que hicieron posible la consecución del presente proyecto de investigación.

Al Doctor Eduardo Carrillo, por hacer que sucedan las cosas. Sin su empuje, constancia, ánimo y disciplina que le impregnó al trabajo de tesis, hubiese sido imposible culminarla. Quisiera expresar que con más personas de su corte y dedicación la educación e investigación en Colombia tendría mayor atención mundial.

A la Universidad Autónoma de Bucaramanga, por el riesgo asumido al integrar dentro de sus programas una maestría en Software Libre. Son pocas las instituciones que a nivel nacional tienen verdadero compromiso para divulgar y brindar al mercado interno profesionales formados adecuadamente que brinden soluciones de gran impacto basadas en el pensamiento libre.

A la Universidad Sergio Arboleda, por abrirme la puerta a tan maravilloso mundo científico en donde la matemática más que algo misterioso y oculto, es algo gratificante que permite expresar desde la más pequeñas de las formas hasta los sistemas más complejos y caóticos.

A los compañeros de matemáticas por su constante apoyo, aún en tiempos de dificultad, brindando siempre ánimo y nuevos retos que hacían olvidar la dificultad del camino por recorrer.

Contenido

1 INTRODUCCIÓN.....	1
2 CAPITULO I.....	5
2.1 MARCO TEÓRICO.....	6
2.2 ESTADO DEL ARTE	6
2.3 DESCRIPCION DEL MÉTODO EMPLEADO.....	19
3 CAPITULO II.....	20
3.1 DESARROLLO DEL PROYECTO.....	21
3.1.1 Evaluación de los modelos existentes en el software libre y software propietario.	21
3.1.2 Identificación de las variables de mayor impacto dentro de la solución.	27
3.1.3 Cálculo del la disponibilidad del sistema	29
3.2 PROPUESTA DE UNA ARQUITECTURA BASE	32
3.2.1.1 Cluster para balanceo de canales.....	33
3.2.1.2 Soluciones de hardware:	33
3.2.1.3 Soluciones de software con hardware de comodidad.....	34
3.2.1.4 Cluster de servidores Linux LVS.	35
3.2.1.5 Arquitectura general de la solución.....	37
3.2.1.6 Esquema de direccionamiento de la solución	38
3.2.1.7 Granja de Servidores Apache, Perl y LMS Maat-g.	39
3.2.1.8 Cluster de servidores Mysql	39
3.3 INSTALACIÓN DE UN AMBIENTE SIMULADO PARA REALIZAR UNA PRUEBA DE CONCEPTO DE LA ARQUITECTURA PROPUESTA.	39
3.4 REALIZACIÓN DE PRUEBAS DE CARGA PARA MEDIR LA DISPONIBILIDAD Y EL COMPORTAMIENTO DE LA SOLUCIÓN.....	40
3.4.1 Resultados de las pruebas.....	40
3.4.1.1 Caída de 1 Nodo del Cluster LVS.....	41
3.4.1.2 Caída de 1 Nodo de la granja de servidores apache, perl y LMS.....	41
3.4.1.3 Caída de 1 Nodo del Cluster de Mysql.....	41
3.5 CRECIMIENTO DE LA PLATAFORMA PROPUESTA.	42
4 CAPITULO III.....	45
4.1 CONCLUSIONES.....	46
4.2 RECOMENDACIONES.....	48
5 ANEXO NO. 1.....	51

5.1 MANUAL DE CONFIGURACIÓN E INSTALACIÓN PLATAFORMA MAAT-G.	52
5.2 INSTALACIÓN Y CONFIGURACIÓN PLATAFORMA MAAT-G SOBRE SISTEMAS OPERATIVO CENTOS OS.	53
5.2.1 PREREQUISITOS.	53
5.2.1.1 Requerimientos Mínimos de Hardware:	53
5.2.1.2 Requerimientos Mínimos de Software:	53
5.2.1.3 Proceso de instalación de los productos.	53
5.2.1.4 INSTALACION DE MODULOS DE PERL.	57
5.2.2 INSTALACION Y CONFIGURACION DE PLATAFORMA MAAT-G SOBRE CENTOS	59
5.2.2.1 Pre-requisitos:	59
5.2.2.2 Instalación Base de Plataforma G.	61
5.2.2.3 Post-instalación de la plataforma g.	66
5.3 CONFIGURACIÓN DEL CLUSTER LVS CON HEARTBEAT	69
5.4 INSTALACIÓN DE CLUSTER MYSQL CON REPLICACIÓN	76
5.4.1 Instalación en cada Nodo.	76
5.4.2 Creación de usuarios para replicación.	78
5.4.3 Creación de archivo de configuración de replicación.	78
5.5 INSTALACION DE BALANCEADOR DE ENLACES CON LINUX IPFILTER.	84
5.6 INSTALACION CLUSTER DE SERVIDORES BALANCEADOS CON LVS.	88
5.6.1 Instalación de paquetes.	88
5.6.2 Activación de las reglas de routing.	97
5.6.3 Configuración del servicio de balanceo y alta disponibilidad.	100
5.7 REALIZACIÓN DE PRUEBAS DE DISPONIBILIDAD.	104
5.7.1 Creación del plan de pruebas.	105
5.7.2 Instalación de agentes remotos.	108
5.7.3 Realización de la prueba.	110
5.7.4 Resultado de las pruebas.	111
5.7.4.1 Caída de 1 nodo del Cluster LVS.	112
5.7.4.2 Caída de 1 Nodo de la granja de servidores apache, perl y LMS.	114
5.7.4.3 Caída de 1 Nodo del Cluster de mysql.	115
6 ANEXO 2.	116
6.1 MODELO DE ENTREVISTA REALIZADA.	117

Índice de ilustraciones

<i>Gráfico 1: Modelo HeartBeat para Alta Disponibilidad.....</i>	<i>35</i>
<i>Gráfico 2: Arquitectura General de la solución.....</i>	<i>36</i>
<i>Gráfico 3: Esquema de direccionamiento de la solución.....</i>	<i>37</i>
<i>Gráfico 4: Contenido del instalador de Maat-g.....</i>	<i>58</i>
<i>Gráfico 5: Elección de la versión de Maat-g.....</i>	<i>59</i>
<i>Gráfico 6: Validación de los módulos de Perl requeridos.....</i>	<i>60</i>
<i>Gráfico 7: Archivo en que se hacen los cambios para Maat-g.....</i>	<i>60</i>
<i>Gráfico 8: Confirmación de parámetros de actualización</i>	<i>61</i>
<i>Gráfico 9: Confirmación de la instalación de Maat-g Base.....</i>	<i>61</i>
<i>Gráfico 10: Menú de Inicio del servicio Maat-g.....</i>	<i>65</i>
<i>Gráfico 11: Heartbeat establecido en el nodo lvs1.....</i>	<i>71</i>
<i>Gráfico 12: Nodo de respaldo del HeartBeat.....</i>	<i>72</i>
<i>Gráfico 13: Generación de petición http al sistema balanceado.....</i>	<i>100</i>
<i>Gráfico 14: Informe agregado de pruebas de disponibilidad.....</i>	<i>101</i>
<i>Gráfico 15: Almacenamiento del test de pruebas en archivo.....</i>	<i>102</i>
<i>Gráfico 16: Configuración de agentes remotos, para prueba distribuida.....</i>	<i>104</i>
<i>Gráfico 17: Arquitectura para las pruebas de disponibilidad.....</i>	<i>105</i>
<i>Gráfico 18: Resultados de las pruebas sin elementos de falla.....</i>	<i>106</i>

<i>Gráfico 19: Peticiones erróneas por falla en un nodo del lvs.....</i>	<i>106</i>
<i>Gráfico 20: Inicio del traslado de recursos al nodo secundario lvs.....</i>	<i>107</i>
<i>Gráfico 21: Traslado completo de servicios al nodo de respaldo lvs.....</i>	<i>108</i>
<i>Gráfico 22: Peticiones fallidas por nodo caído en la granja de apache.....</i>	<i>108</i>
<i>Gráfico 23: Fallo de peticiones con caída de nodo mysql.....</i>	<i>109</i>

Índice de tablas

Tabla 1: Comparación soluciones de HA en Linux.....	23
Tabla 2: Factores de mayor impacto en una arquitectura HA según sector.....	27
Tabla 3: Niveles de disponibilidad por año.....	30

GLOSARIO

CLUSTER: Conjunto de servidores configurados de una manera particular para que el servicio que brindar pareciera ser ejecutado por un único elemento logrando mejores tiempos de respuesta y desempeño.

Grid: Computación en malla. Permite utilizar de forma coordinada un conjunto de recursos distribuidos a lo largo de organizaciones virtuales que ceden parte de su capacidad de cómputo local, para lograr capacidades de procesamiento y gestión mayores mediante la interconexión al sistema global o grid.

HA: High Availability. Sistema de cómputo que tiene niveles de disponibilidad altos y que es capaz de estar siempre activo, aún ante presencia de fallas en el mismo.

LMS (Learning Management System): Sistema de Gestión de Aprendizaje.

LVS: Linux Virtual Server. Proyecto desarrollado en software libre como parte de la soluciones de alta disponibilidad para hacer ver un conjunto de servidores como uno solo cobijado bajo una única IP.

MULTICORE: Línea de procesadores que en una mismo chips, insertan más de una CPU, permitiendo que compartan o no la caché de nivel 2.

N+1: Modelo de colocación de dispositivos que indica que cada elemento dentro de una arquitectura, tendrá uno de respaldo y que no existirá ningún elemento activo que por lo menos tenga un elemento adicional de respaldo.

SLA: System Level Agreement: Acuerdo de nivel de servicio. Compromiso de un proveedor de servicio que el ofrecimiento tendrá un porcentaje de disponibilidad es decir que siempre será accesible y estará funcional dentro de ese porcentaje de tiempo al año.

SOA: Arquitectura orientada a servicios.

WAP: Wireless Application Protocol

RESUMEN

En la era de la información, no tenerla disponible podría resultar fatal, por tal razón las aplicaciones de misión crítica de las compañías más importantes del sector productivo mundial, confían su presencia digital en las arquitecturas altamente disponibles que garanticen la no interrupción de la prestación de sus servicios electrónicos. En gran medida estas arquitecturas corresponden a modelos propietarios con software muy especializado de alto costo, sin embargo con el auge del software libre en el sector empresarial ya existen soluciones que correctamente ensambladas y configuradas puede brindar iguales o mejores prestaciones que las basadas en modelos propietarios. En la presente tesis se realiza un análisis de las principales arquitecturas encontradas y la aplicación de una en particular al software LMS de la empresa Maat-g.

Palabra claves: Arquitecturas de alto desempeño, alta disponibilidad para sistemas LMS, modelo de balanceo para software Maat-g.

ABSTRACT

In the information age, not having it available could prove fatal for this reason the mission-critical applications to major companies worldwide production sector, trust

its digital presence in highly available architectures ensure no interruption of the provision of their electronic services. To a large extent these models are proprietary architectures with highly specialized software expensive, however with the rise of free software in business and there are solutions that properly assembled and configured can provide equal or better performance than those based on proprietary models. In this thesis makes an analysis of the major architectures encountered and the application of a particular software company LMS Maat-g.

Keywords: High Performance architecture, High Availability for LMS Solutions
OpenSource Load Balancing model for Maat-g.

1 INTRODUCCIÓN

El incremento exponencial de consultas que año tras año se presentan hacia los portales corporativos expuestos e Internet, ha generado uno de los mayores problemas que en la actualidad enfrentan los departamentos de T.I, la disponibilidad de sus aplicaciones, especialmente aquellas cuya servicios se brindar completamente en línea. Las arquitecturas multicapa por su parte han permitido que las cargas puedan ser balanceadas dentro de componentes de negocio lógica y físicamente distribuidos en plataformas de distribución de peticiones . Sin embargo el interrogante sobre el límite posible de clientes soportados y el tiempo de respuesta aceptable para dichas peticiones prevalece. Este temor ha hecho que se establezcan contratos de prestación de servicios tecnológicos con SLA de disponibilidad específicos (en algunos casos del 99,9999%, es decir máximo 31 segundos en un año de indisponibilidad) en los cuales se mitiga el riesgo colocando sistemas altamente redundantes en modelos N+1 e incluso N+2.

En este panorama las soluciones propietarias juegan un papel muy importante debido a que son los mayores proveedores de soluciones en este campo en combinación de hardware, software y servicios, por supuesto asociado a costos

muy altos que solo son accesibles por negocios que inviertan gran cantidad de su presupuesto operativo al tema tecnológico. Sin embargo para aquellas empresas que proveen servicios en línea y que por su tamaño o presupuesto les resulta imposible acceder a este tipo de soluciones propietarias existen también soluciones que brindan esquemas de disponibilidad muy altos en una combinación de hardware de comodidad y software libre.

En la presente proyecto se realiza un análisis de los elementos que componen una arquitectura tecnológica altamente disponible y las opciones que dentro del software libre existen para soportarla. Después de realizar una revisión de los diferentes alternativas existentes que ofrece el mercado tanto de software libre como de software propietario se propone una arquitectura, se precisa cada componente y se realiza una implementación de prueba para el software LMS de la empresa Maat-g.

Es de aclarar que el presente proyecto no contempla elementos de seguridad de la información o seguridad de la plataforma, debido a que cuenta con su propia temática y metodología existentes, lo cual obliga a que se aborde de manera explícita utilizando sus propios métodos y herramientas. Para los interesados en el tema se recomienda realizar una lectura detallada del libro “*Designing Security Architecture Solutions*”

1 Ramachandran, Jay. Designing Security Architecture Solutions. Wiley. 2002.

Los objetivos del presente documento entonces se pueden enumerar de la siguiente manera:

- Habilitar una arquitectura que le permita a la plataforma de E-learning Maat-g ser implementada sobre un ambiente de alta disponibilidad, brindando servicio ininterrumpido, aún en caso de fallo.
- Elegir una alternativa de alta disponibilidad, dentro de la oferta de software libre que se integre con los servidores web y de aplicaciones de la plataforma Maat-g.
- Desarrollar un modelo de crecimiento que permita crecimiento tanto en concurrencia como en volúmenes de datos y que pueda integrar diferentes fabricantes de hardware.
- Brindar una solución de grado empresarial utilizando las herramientas de software libre existentes y de esta manera crear un antecedente a futuros trabajos para que puedan generar componentes que puedan ser operativos dentro del mundo empresarial.

El presente proyecto se encuentra distribuido en 3 capítulos de la siguiente manera:

Capítulo 1. Marco Teórico: Dentro de él se incluye el estado del arte en cuanto a

soluciones disponibles, junto con un trasfondo teórico de cada uno de los elementos que componen una solución altamente disponible.

Capítulo 2. Desarrollo del proyecto. En este capítulo se describe la manera como fue elaborada la arquitectura, los elementos que se tuvieron en cuenta para la elección de cada uno de los componentes y las manera como cada uno de ellos se ha configurado. Los procesos de configuración descritos en este capítulo, se presentan como una anexo a modo de manuales.

Capítulo 3. Conclusiones y recomendaciones. Se presentan las conclusiones y recomendaciones sobre las experiencia de diseñar, implementar y probar la arquitectura propuesta.

2 CAPITULO I

2.1 MARCO TEÓRICO

El proyecto denominado “Arquitectura de alta disponibilidad para la plataforma LMS de Maat-g”, corresponde al segundo proyecto de maestría en software libre que sobre la plataforma LMS de Maat-g se realiza. En el primero de ellos realizado en el año 2008 se presenta un estudio comparativo de la plataforma LMS Maat-g con una solución representativa del mundo del software libre, considerando aspectos como funcionalidad, rendimiento y usabilidad.

El presente proyecto se enfoca en proveer a dicha plataforma de las facilidades que le permitan ser competente y de clase empresarial, siendo provista de los factores de alta disponibilidad y balanceo de carga para atender al incremento de usuarios de la misma.

2.2 ESTADO DEL ARTE

El advenimiento de los sistemas multicore, ha producido sistemas de cómputo mucho más rápidos, robustos y con sistemas de intercomunicación que brindan mejores anchos de banda y velocidades de acceso, los ambientes de software se enfrentan a nuevos desafíos para los cuales no siempre están preparados.

Las tendencias tecnológicas actuales indican que la virtualización y los sistemas multicore seguirán en aumento en los próximos cuatro años² y que solo podrán ser aprovechados de mejor manera si se integran técnicas de programación paralela dentro para el desarrollo de software, permitiendo que la velocidad de reloj del procesador no sea el único indicador del nivel de desempeño. El ahorro de energía con un medio ambiente seriamente afectado por el consumo incesante de recursos se convierte en un necesidad y exigencia; las velocidades de reloj más altas producen mayores consumos de energía, por tal razón con mayores necesidades de cómputo, las soluciones tienden a usar menores velocidades de reloj integrando mayor números de cores y realizando software que aproveche al máximo las plataformas de soporte.³

La plataforma de E-learning Maat-g, consciente de estos nuevos desafíos, especialmente aquellos que le imponen las nuevas necesidades de sus clientes, que en la actualidad compiten por tener un mejor servicio al cliente y especialmente presentar mejores tiempos de respuesta en cada una de las transacciones ofrecidas, al igual que garantizar los más altos niveles de

2 Virtualization and Multicore Innovations Disrupting the Worldwide Server Market, According to IDC. Disponible en Internet.

http://www.redorbit.com/news/technology/875609/virtualization_and_multicore_innovations_disrupting_the_worldwide_server_market_according/index.html

3 HERLYHY, Mauricie. The Art of Multiprocessor programming. Morgan Kauman. Burlingtown, USA. 2008

disponibilidad requiere que se pueda desarrollar e implementar los mecanismos que le permitan cumplir con dichos objetivos, afectando las bases ya instaladas en la menor manera posible.

Una vez realizado los análisis respectivos y entendiendo que el problema de balanceo de carga dentro de la plataforma debe ser un servicio no de la parte transaccional de la misma, sino más bien de un elemento independiente con bajo acoplamiento pero que permita ser integrado en la totalidad de los procesos, se puede concluir que la elaboración o adaptación de un módulo de apache presenta las más grandes ventajas de adaptación, desempeño, confiabilidad y soporte.

Para solucionar los problemas de concurrencia y tiempo de respuesta, se han desarrollado mecanismos de distribución, permitiendo así que se puedan dividir las cargas dentro de los denominados “Clusters” de procesamiento. Este tipo de solución permitió tener mejores niveles de procesamiento sin embargo el tiempo de respuesta seguía siendo un problema sin atacar, sumado a la complejidad de sincronizar los datos que se procesaban en cada uno de los miembros del Cluster para evitar duplicidad o reescritura de información. Surgieron mecanismos de distribución basados en direccionamiento también llamados “Cluster DNS” con lo cual bajo un mismo nombre de dominio se podía acceder a grajas de servidores; esta solución sin embargo era muy poco inteligente debido a que por el modelo de delegación de DNS parte se almacena localmente en los clientes y las

subsiguientes solicitudes siempre llegarían al mismo miembro del Cluster, con lo cual el comportamiento del balanceo era inesperado, llegando a los mismos problemas de concurrencia y tiempo de respuesta lentos.

En paralelo a este tipo de problemas, también se debía resolver el problema de la utilización de un único mecanismo de control para las salidas a internet, especialmente en aquellas corporaciones que empezaban a desarrollar políticas de uso del mismo y restricciones en tipo de contenido. El problema aquí fue solucionado entre otros con el desarrollo e implementación de proxys que capaces de mantener el control del origen y el destino, además de aplicar algunas políticas por el tipo de contenido.

Este desarrollo sirvió de base para proponer mecanismos que permitieran balancear las peticiones del usuario, realizar control de estado de las peticiones, aplicar políticas según el tipo de petición y mantener sincronizada la información dentro de todos los miembros del Cluster. Es en este momento que surgen los “balanceadores de carga”⁴ para brindar flexibilidad, escalabilidad y alta disponibilidad de aplicaciones. A partir del surgimiento de los balanceadores de carga para aplicaciones muchos otros tipos de Balanceadores han visto la luz, desde balanceadores de enlaces de red⁵ hasta mecanismos balanceadores de

4 BOURKE , Tony. Server Load Balancing. Sebastopol, CA . O'Reilly & Associates, Inc. 2001. Pag 8-9

5 NEEDHAM, Jason. Link load balancers smooth Web traffic [online]. 2002. Disponible en internet: <http://www.networkworld.com/news/tech/2002/0916tech.html>

instancias de base de datos como Oracle Rac (Real Application Cluster).

Con los primeros desarrollos surgieron también los desafíos del balanceo de aplicaciones. ¿Cómo generar políticas de balanceo? ¿Cuál es la mejor manera de balancear la aplicación? ¿Cuál es el nivel de acoplamiento que tendrán las aplicaciones para que permitan ser balanceadas?. Los primeros desarrollos presentaron solo políticas de balanceo basado en algoritmos Round Robin⁶, con lo cual todos los miembros del Cluster debían ser homogéneos en cuando a tecnología y capacidad; rápidamente esto se volvió un impedimento debido a que la renovación tecnológica normalmente sigue la ley de Moore⁷ y la mezcla de tecnologías y capacidades forma parte esencial del sostenimiento tecnológico. En respuesta se desarrollaron Balanceadores capaces de aceptar más de una política basadas en información propias de las aplicaciones y de las plataformas donde esta se ejecutaban; esta nueva etapa exigió de componentes más especializados capaces de mantener información histórica, interactuar con sistemas operativos, dispositivos de red, aplicaciones de bases de datos y servidores de aplicaciones.

Surgen entonces dos grandes divisiones dentro de los balanceadores de carga de aplicaciones:

El primer enfoque requiere que las decisiones de aplicación de políticas de balanceo sea dinámico y estén basadas en el aprendizaje del. A este tipo de

6 <http://choices.cs.uiuc.edu/~f-kon/RoundRobin/node2.html>

7 http://en.wikipedia.org/wiki/Moore%27s_law

balanceo se le denominó “Balanceo adaptativo basado en aprendizaje” y los elementos principales estaban determinados por el tipo de algoritmo de aprendizaje y los tipos de algoritmos de distribución soportados; este método es especialmente útil en ambientes donde las condiciones no pueden ser conocidas de antemano y la criticidad del sistema impide que se reprogramen las políticas con afectación de la disponibilidad del sistema, sin embargo la aplicación de este tipo de método castiga los tiempos de respuestas de las aplicaciones ya que los procesos de aprendizaje toman ciclos de CPU y además cuando se hablan de cientos de millones de peticiones esta contención en la respuesta puede resultar catastrófica para la aplicación y para el usuario de la misma.

El segundo enfoque por su parte, supone un ambiente controlado donde la aplicación de políticas no requieren adaptación dinámica y donde los posibles valores que se establezcan son resultado de un proceso de ajuste de desempeño o “performance tuning”. Esta ambiente es el más utilizado en ambientes altamente transaccionales, donde la concurrencia y el tiempo de respuesta son los factores más críticos. Este enfoque es el más utilizado en las aplicaciones web, cualquiera que sea la naturaleza de las mismas.

Los algoritmos de distribución utilizados en el segundo enfoque pueden tomar decisiones basados en información que reciben de las peticiones del usuario o en información que reciben de las aplicaciones servidor.

La información del cliente en que ser tomada para aplicación de reglas puede ser:

- Dirección o rango de direcciones de origen
- Tipo de petición realizada
- Protocolo y puerto solicitado (HTTP, HTTPS, SOAP, propietario)
- Patrón en una cadena de consulta.
- Tipo de aplicación cliente utilizado (navegador, cliente ligero, cliente robusto)
- La información del las aplicaciones servidor que se puede utilizar para aplicar reglas puede ser:
 - Número de peticiones servidas, procesadas o en cola.
 - Cantidad de bytes servidos.
 - Velocidad de conexión disponible.
 - Carga actual del servidor o grado de ocupación.
 - Respuesta de estado de una aplicación.
 - Tipo de aplicación que sirve.
 - Prioridad o peso del servidor en el Cluster.
 - Número de sesiones manejadas.

A pesar que esta cantidad de parámetros pareciera ser suficiente para ayudar a aplicar políticas de balanceo, las actuales demandas de procesamiento junto con

el advenimiento de tecnologías de virtualización, computación en malla y computación en nube requieren que continuamente se examinen nuevas métricas para que la distribución de las cargas cumpla con las premisas de disponibilidad, flexibilidad y tolerancia a fallos.

De acuerdo al grado de criticidad de las aplicaciones los balanceadores de carga pueden ser implementados como soluciones de software o como soluciones de hardware.

Las soluciones de hardware se brindan preconfiguradas en equipos especializados o “appliance” que contienen un conjunto de elementos diseñados para lograr el mejor desempeño. Muchos de estos están dotados de sistemas operativos propietarios y poseen interfaces de administración también propietaria. Estas soluciones normalmente exponen garantías de cantidad de tráfico que puede procesar o número de sesiones concurrentes que puede mantener. El gran problema de este tipo de soluciones es su costo que en promedio supera los 20 mil dólares en los equipos de rango bajo o de entrada.

Las soluciones de software son mucho más baratas y flexibles ya que generalmente se pueden implementar en un conjunto amplio de plataformas de base, sin embargo tiene en su contra que para lograr los mejores rendimientos necesita una combinación de hardware muy bien seleccionado (que resulta

costoso) y un conjunto de ajustes que solo puede ser hecho una vez la solución esta operativa; dentro de esta categoría podríamos mencionar:

- Apache Proxy balancer
- Linux Virtual Server
- Linux HA y RedHat Cluster Suite
- Windows Cluster Services
- IBM WebSphere Application Server Edge Components

Para aplicaciones web multicapa especialmente es relevante aquellos balanceadores capaces de integrarse a los servidores Web más populares como son Apache Web Server e Internet Information Server.

Apache por ser software de fuente abierta⁸ (OpenSource), por ser el líder en el segmento de servidores web⁹ y además por estar diseñado con un núcleo reducido (especialmente a partir de su segunda versión) y un conjunto de extensiones que permiten agregar nueva funcionalidad¹⁰ de manera modular, resulta muy útil no solo desde el punto de vista del diseño sino también para la

8 Disponible en internet: <http://www.apache.org/licenses/>

9 Market Share for Top Servers Across All Domains August 1995 - June 2009. Netcraft. Disponible en internet http://news.netcraft.com/archives/web_server_survey.html

10 DRAGOI, Octavian Andrei. The Conceptual Architecture of the Apache Web Server [online]. Department of Computer Science, University of Waterloo. Disponible en internet: http://www.cs.ucsb.edu/~tve/cs290i-sp01/papers/Concept_Apache_Arch.htm

implementación en ambientes de desarrollo dispersos geográficamente. Es por esta razón que la gran cantidad de lenguajes de programación y frameworks de desarrollo módulos soportados que luego de un proceso de validación es software oficial del servidor apache.

La creación de un módulo de apache implica la interacción con varios módulos de base: el Apache Portable Runtime (APR)¹¹ permite asegurar la portabilidad dentro de los diferentes sistemas operativos en los cuales opera el servidor y el MultiProcessing Module (MPM) permite tener control de los diferentes subprocesos e hilos que se mantengan activos. Cada uno de estos por formar parte de la arquitectura base son fundamentales en el momento de implementar un nuevo módulo para el servidor¹².

Dentro de los módulos más populares se encuentra el mod_perl¹³, encargado de dar soporte al lenguaje PERL considerado uno de los más poderosos, flexibles y ampliamente utilizados dentro del sistema operativo GNU/Linux¹⁴ debido a su muy completo soporte para expresiones, a scripting e integración con funciones del sistema Operativo. Este módulo en la actualidad permite la extensión de funcionalidades de los módulos existentes o la escritura en lenguaje perl de

11 Disponible en internet: <http://apr.apache.org/>

12 WEINSTEIN, Paul. Pro Apache 2.0 Module Development: From Professional to Expert. APress,US. Diciembre 2004. pag 21-50

13 Disponible en internet: <http://perl.apache.org/>

14 Trends in Programming Languages: Disponible en internet: <http://www.sys-con.com/node/450313>

módulos nuevos.

Las razones antes expuestas en gran parte motivaron a que la empresa Maat Gknowloge¹⁵ decidiera crear una plataforma orientada completamente a Grid y Web basada en el soporte brindado por el lenguaje Perl, el cual sirve de base para que cada uno de los componentes desarrollados puedan correr como módulos autónomos soportados en el módulo de apache mod_perl.

Las tecnologías orientadas a Grid no son nuevas en el ambiente informático, han pasado por cerca de dos décadas de maduración esperando que las tecnologías de hardware y telecomunicaciones evolucionaran hacia estados de alta capacidad, confiabilidad y además velocidades de transmisión, de tal manera que los ideales de la computación distribuida en entornos colaborativos pudiese ser una realidad. A este respecto David de Roure realiza una revisión de la evolución del Grid¹⁶ en donde identifica las 3 generaciones de desarrollo de Grid mostrando de una manera práctica los diferentes entornos y plataformas que se han desarrollado en cada una de estas generaciones y los retos que para la tercera generación se avecinan ya que apenas estamos en los momentos iniciales de su desarrollo e implementación.

15 Disponible en internet: <http://www.Maat-g.com/>

16 ROURE, David De, BAKER, Mark A et al. The Evolution of the Grid. University of Southampton. 2003. Disponible en internet: <http://www.semanticGrid.org/documents/evolution/evolution.pdf>

Cabe la pena destacar que según este análisis las tecnologías Grid tendrán una clara orientación al negocio especialmente en arquitecturas orientadas en ambientes web, pues se considera la web como la mayor infraestructura de información interconectada, los metadatos son los elementos mediante los cuales se puede realizar inteligencia sobre la información y las diferentes alternativas de interacción debe estar orientadas a servicios de acceso Grid y se anticipa que la siguiente generación estará concentrada en el manejo del conocimiento para ser almacenado, consultado y catalogado especialmente cuando esté disperso a grandes organizaciones de nivel transnacional que se puedan presentar como unidades de conocimiento distribuida¹⁷.

Enmarcado dentro de este enfoque Maat-g basa su propuesta en una plataforma completamente orientada al Grid, no solamente con elementos tecnológicos (han desarrollado un lenguaje basado en grafos) sino a nivel de soluciones verticales que le permitan aprovechar al máximo la colaboración, la flexibilidad y la capacidad de la distribución de sus soluciones.

Las soluciones de T.I tiene un nivel de clasificación de acuerdo a su criticidad y pueden ser clasificadas en:

¹⁷ Ibid. Pg. 30-31.

- Disponibilidad normal.
- Alta disponibilidad.
- Disponibilidad de misión crítica.
- Disponibilidad continua.
- Disponibilidad de Transacciones en tiempo real.

Cada uno de estos niveles está asociado con la cantidad máxima de tiempo de tiempo que el sistemas podrá estar fuera de línea ya sea por eventos controlados o por errores presentados. Esta nivel de disponibilidad se presenta normalmente como el porcentaje de tiempo que el sistemas está en línea prestando servicios de manera continua y tiene los niveles indicados en la tabla No. 3. Niveles de disponibilidad por año. Estos niveles colocan las arquitectura en un nivel de criticidad expuesto de la siguiente manera.¹⁸

- Disponibilidad normal, 99%.
- Alta disponibilidad, 99.9%.
- Disponibilidad de misión crítica, 99.99%.
- Disponibilidad continua, 99.999%.
- Disponibilidad de Transacciones en tiempo real, 99.9999%.

¹⁸ Un análisis más detallado sobre el impacto de la disponibilidad en el negocio se puede encontrar en el artículo, Five Nines. Chasing the dream. Disponible en internet:

<http://www.continuitycentral.com/feature0267.htm>

El presente proyecto fija su meta en una arquitectura de alta disponibilidad con un nivel esperado de 99.96%.

2.3 DESCRIPCION DEL MÉTODO EMPLEADO

El presente proyecto buscó seguir un enfoque práctico acorde a los lineamientos enmarcados dentro de una maestría de profundización, sin embargo agregar elementos teóricos importantes que permitan desarrollar herramientas para la evaluación de la solución. Para eso se adoptó el siguiente procedimiento:

1. Revisión documental y evaluación de los modelos existentes en el software libre y en el software propietario.
2. Identificación de las variables de mayor impacto dentro de la solución.
3. Propuesta de una arquitectura base.
4. Instalación de un ambiente simulado para realizar una prueba de concepto de la arquitectura propuesta.
5. Realización de pruebas de carga para medir la disponibilidad y el comportamiento de la solución.
6. Presentación de resultados de investigación y las recomendaciones.

3 CAPITULO II

3.1 DESARROLLO DEL PROYECTO

En este apartado se realiza una descripción de cada una de las tareas realizadas:

3.1.1 Evaluación de los modelos existentes en el software libre y software propietario.

Dentro de esta etapa se realizó la investigación tanto bibliográfica como de consulta a proveedores de tecnología nacional y de talla mundial, para determinar el estado del arte.

Se realizaron visitas a instalaciones que en la actualidad emplearan este tipo de soluciones y además a empresas que realizaran instalaciones de las mismas.

Las implementaciones revisadas fueron:

1. Proyecto MUISCA, del la Dirección de Impuestos y Aduanas Nacionales DIAN.
2. Proyecto SOFIA, del Servicio Nacional de Aprendizaje Sena.
3. Sistema de consulta de los seguros sociales.
4. Sistema AFIS de la Registraduría Nacional de la República.

La empresas de implementación consultadas por su parte fueron:

- SISA -SA. Expertos en arquitecturas IBM y sistemas de Clustering.

- OpenBox Ltda. Expertos en implementación de soluciones de Software libre.
- F5 technologies. Empresa especializada en soluciones de balanceo de carga y alta disponibilidad de enlaces.
- Cisco de Colombia. Empresa de líder en soluciones de networking incluyendo soluciones de balanceo de carga, de enlaces y de HA.
- Corporación ITS. Empresa integradora especializadas en soluciones HP, de alta disponibilidad con Linux.

Una vez realizado la recopilación de las experiencias en campo de cada uno de los profesionales y empresas dedicadas a este tipo de soluciones, se identificaron algunas de las soluciones más populares e implementadas en el mercado Colombiano.

Dentro del software propietario:

- Microsoft Cluster Services¹⁹
- Sun Solaris Cluster²⁰
- IBM Power HA, GPFS, HACMP.²¹

Dentro del software libre:

- Linux HA con HeartBeat²²

19 Disponible en internet: <http://www.microsoft.com/windowsserver2003/enterprise/Clustering.mspx>

20 Disponible en internet: <http://www.sun.com/software/solaris/Cluster/>

21 Disponible en internet: <http://www-03.ibm.com/systems/power/software/availability/aix/index.html>

22 Disponible en internet: <http://www.linux-ha.org/>

- RedHat Cluster Suite²³
- Rocks Cluster ²⁴

Debido al enfoque de la presente proyecto se eligieron aquellos que pertenecen a la categoría de software libre además de los costos asociados, permitiendo que se pueda realizar un piloto sin incurrir en costos de licenciamiento.

Para las soluciones de software libre se analizaron los siguientes aspectos:

1. Tipo de licencia, arquitecturas de hardware soportadas, documentación existente, implementaciones de referencia y distribuciones Linux soportadas.

Los resultados se presentan en la siguiente tabla, junto con un análisis de resultados.

23 Disponible en internet: http://www.redhat.com/Cluster_suite/

24 Disponible en internet: <http://www.rocksClusters.org/>

Solución	Linux HA con HeartBeat	RedHat Cluster Suite	Rocks Cluster
Tipo de licencia	GPL y LGPL	RedHat suscription Support	Rocks CopyRight
Arquitecturas de procesadores soportadas.	ia32, Power PC y System Z.	ia32, ia64, PowerPC, X86_64	X86,x86_64
Documentación existente.	Un poco escasa y limitada a howtos, presentaciones y algunas referencias en libros	Amplia y mantenida oficialmente, incluyente manuales de uso, implementación y técnicos.	Amplia documentación tanto por la comunidad como por los diversos fabricantes.
Distribuciones Linux Soportadas	Suse, Redhat, TurboLinux, Debian, Gentoo.	RedHat y derivados incluyendo Centos y Suse.	Linux Kernel 2.4 y 2.6, Solaris

Tabla 1: Comparación soluciones de HA en Linux

Fuente: Autor

Los tipos de licencias y arquitecturas se explican a continuación.

Las licencia GPL (General Public License) es una de las más populares dentro del mundo del software libre y se caracteriza por su efecto “viral”, es decir incluye una clausula que obliga a quienes desarrollen o enlacen programas con software libre a que también licencien sus productos bajo los términos de la licencia GPL, actualmente se encuentra vigente la versión 3 e incluye un apartado especial para los dispositivos de hardware y los productos de “software como servicio”²⁵.

La licencia LGPL (Lesser General Public License) es un tipo de licencia menos restrictiva que la GPL pensada especialmente para librerías y enlaces dinámicos en tiempo de ejecución, no obliga que si existen un enlace blando (en tiempo de

²⁵ Disponible en internet: <http://www.gnu.org/licenses/gpl.html>

ejecución) el software que lo use tenga que ser licenciado bajo los términos de la misma licencia²⁶.

RedHat suscription Support, en realidad no es una licencia de uso de software, sino un contrato de soporte que permite el uso de marcas registradas, además de acceder a software específico, documentación y soporte sobre el mismo. Cada software que se use tiene su propia licencia, sin embargo la suscription mantiene una relación de todas ellas²⁷.

Rocks CopyRight, es un tipo de licencia BSD que obliga a quien lo use o integre mantenga los créditos de sus autores además de exigir una solicitud expresa en caso de sublicenciar un producto que incluya este software.

Las arquitecturas de procesadores pueden ser:

- IA32. Procesadores con juego de instrucciones x86 para 32 bits. Se incluyen procesadores Intel, AMD y otros.
- IA64. Procesadores Itanium con soporte de 64 bits.
- X86_64. Procesadores con juego de instrucciones x86 para 64 bits.
- Power PC. Procesadores tipo RISC de IBM con soporte para 64 Bits.
- System Z. Procesadores especializados para mainframe.

Una comparación más extensa entre los tipos de procesadores fue realizada por Chan Kit y Somasundaram Meiyappan²⁸

26 Disponible en internet: <http://www.gnu.org/copyleft/lesser.html>

27 Disponible en internet: https://www.redhat.com/licenses/Enterprise_Agr_Argentina.pdf

28KIT, Chan y Meiyappan Somasundaram. Comparative study of the Pentium and PowerPC family of micro-processors . Dispñible en internet: <http://www.webabode.com/articles/Basic%20comparison%20of%20PowerPC%20and%20Pentium%20processor%20families.pdf>

Analizados los aspectos anteriormente mencionados se eligió una solución basada en Linux HA por las siguientes razones.

- Las arquitecturas de cómputo más comunes son las basadas en x86 de 32 o 64 bits, en las cuales está ampliamente soportado esta solución.
- La licencia utilizada es la GPL, la cuál es la más ampliamente aceptada para considerar un software verdaderamente libre y además no contempla costos de uso del software, lo cual es premisa del presente proyecto.
- La documentación aunque no muy completa, es suficiente para realizar correctamente la instalación y el soporte de la comunidad en foros y listas es amplio sobre los problemas comunes que se presentan en la implementación.
- Las distribuciones empresariales y comunitarias lo incluyen dentro de sus repositorios y realizan mantenimiento de los paquetes.

Esta elección sin embargo se aclara fue hecha para el objeto del presente proyecto, sin embargo algunas empresas eligen otros tipos de soluciones de acuerdo a las variables que ellos determinan de mayor valor, como se podrá apreciar en el siguiente apartado.

El presente apartado pretende lograr el objetivo de ***“Elegir una alternativa de alta disponibilidad, dentro de la oferta de software libre que se integre con los servidores web y de aplicaciones de la plataforma Maat-g.”***

3.1.2 Identificación de las variables de mayor impacto dentro de la solución.

Con las arquitecturas y soluciones analizadas, rea realizó un breve análisis de los elementos que son de mayor importancia para las implementaciones de acuerdo al tipo de industria o sector. Para determinar los elementos mencionados se aplicaron entrevistas personales con el personal encargado quien bajo términos de confidencialidad compartieron parte de sus perspectivas, necesidades y retos. Los términos de confidencialidad impiden que en el presente documentos se muestren cada una de las entrevistas, sin embargo en el Anexo No. 2, se muestra el modelo de entrevista.

Los resultados obtenidos se describen en la siguiente tabla:

Sector de actuación	Factores	Descripción
Gobierno	Disponibilidad, Nivel de soporte y garantía, capacitación para funcionarios y gama de proveedores de la solución.	Los servicios prestados de cara al ciudadano requieren siempre estar disponibles (a toda hora, todos los días del año), con un nivel de soporte adecuado (no importa que sea software propietario) y además que existen más de un proveedor para asegurar los procesos licitatorios. La capacitación de los técnicos de las entidades que soportarán las aplicaciones es primordial.
Banca	Disponibilidad, Tiempo de Respuesta y fiabilidad de la plataforma.	Los servicios financieros necesitan estar siempre disponibles (7 x 24 x 365), garantizando tiempos de respuestas adecuados y sobre todo asegurar que la plataforma es segura tanto a nivel de acceso como a nivel operativo. Solo en algunos aspectos se acepta el software libre.
Educación	Bajo costo, Facilidad de administración, capacitación especializada.	Debido a los bajos presupuestos existentes el costo de la solución es un factor importante. Se asumen ciertos riesgos siempre que exista personal de soporte por tal razón la facilidad de administración y la capacitación son factores importantes en este tipo de soluciones. En caso de tener servicios críticos la disponibilidad entra como primer factor de importancia.
Servicios	Disponibilidad, facilidad de uso, bajos costo	Debido a que muchos de los procesos de toma de información ocurren en tiempo real, la disponibilidad de la plataforma debe estar completamente asegurada. El no tener este factor asegurado implica pérdidas importantes de dinero y problemas en los procesos de facturación.

Tabla 2: Factores de mayor impacto en una arquitectura HA según sector

Fuente: Autor

Como se puede observar en los sectores productivos el hecho que sea software libre o no realmente no juega un papel importante, incluso en el sector financiero existen algunos impedimentos para su uso, lo cual confirma que las soluciones presentadas más que ser software libre deben garantizar que los elementos de disponibilidad, soporte y fiabilidad estén asegurados.

Para la selección de la plataforma y la arquitectura propuesta la disponibilidad, junto con la documentación existentes fueron las variables de mayor peso al momento de elegir, con la premisa que debiera ser una solución dentro del modelo de software libre.

3.1.3 Cálculo del la disponibilidad del sistema

Para calcular la disponibilidad de un sistema se deben distinguir dos tipos de interrupciones.

- Interrupciones no planeadas: Son eventos aleatorios que se presentan por fallas no controladas o desconocidas y que no permiten el acceso a los sistemas o aplicaciones.
- Interrupciones planeadas: También consideradas ventanas de mantenimiento, son periodos de tiempo previamente planeados y controlados para realizar procesos de actualización o mantenimiento del sistema. Dentro de estos procesos se pueden reemplazar partes, aplicar parches u realizar procesos de optimización del sistemas o aplicaciones.

Dentro del cálculo de la disponibilidad aparecen las siguiente medidas:

- Tiempo Medio entre Fallas (MTFB): Tiempo promedio medido en horas o minutos entre dos fallas consecutivos de un elemento del sistema. Se calcula

$$MTBF = \frac{TIEMPO TOTAL DEL SISTEMA}{NUMERO DE FALLAS}$$

- Tiempo medio entre interrupciones (MTBI): Tiempo promedio entre dos interrupciones (las interrupciones son planeadas, los fallos no) y se calcula

$$MTBI = \frac{TIEMPO TOTAL DEL SISTEMA}{NUMERO DE INTERRUPCIONES}$$

- Tiempo medio de reparación (MTTR): Tiempo que se tarda en reparar una falla del sistema del sistema o de algunos de sus componentes

De estas medidas fundamentales se desprenden las siguiente:

$$Tasa\ de\ fallas = \frac{1}{MTFB}$$

Así podemos calcular la disponibilidad de un sistemas x año como:

$$Disponibilidad = \frac{MTBF}{MTFB + MTTR}$$

también suele calcularse de la siguiente manera:

$$\text{Disponibilidad} = \frac{\text{TIEMPO ARRIBA SISTEMA}}{\text{TOTAL TIEMPO OPERACION}} \times 100$$

Esta medida está ligada al al concepto de “tiempo fuera de un sistema”, que indica la cantidad de tiempo (en minutos normalmente) que un sistemas está fuera de servicio (por fallos) al año y que puede ser calculado de la siguiente manera:

$$\text{Tiempo fuera x año} = (1 - \text{Porcentaje Arriba}) \times 365 \times 24 \times 60$$

De este modo podríamos definir los siguientes niveles de disponibilidad basado en la cantidad máxima de tiempo fuera por año.

Disponibilidad %	Tiempo fuera por año
99	3 días 15 horas 36 minutos
99,9	8 horas 45 minutos
99.96	3 horas .30 minutos
99,99	52 minutos 33 segundos
99,999	5 minutos 15 segundos
99,9999	31 segundos

Tabla 3: Niveles de disponibilidad por año

Fuente: Autor

Así la arquitectura propuesta con disponibilidad de 99.96, tendría un tiempo máximo fuera de línea por año de 3 horas 30 minutos.

El cálculo de la disponibilidad propuesta se basó en los modelos de disponibilidad propuestos por las diferentes casas de software para la arquitectura planteada; un

análisis más detallado para cada parte de hardware se puede localizar en los manuales de los fabricantes y en sus propios cálculos de disponibilidad, ver a Vargas, Enrique. High Availability Fundamentals²⁹ y configuraciones de ejemplo en availability Digest³⁰ y Understanding High Availability Configuration Examples³¹

3.2 PROPUESTA DE UNA ARQUITECTURA BASE

La arquitectura propuesta se diseña para cumplir con un SLA de 99.96% de disponibilidad, con una disposición N+1 para todos los componentes dentro de la solución en combinación de hardware de comodidad y software libre. El modelo general se puede observar en el gráfico Arquitectura general de la aplicación.

La siguiente propuesta se realiza cumpliendo el objetivo de ***“Habilitar una arquitectura que le permita a la plataforma de E-elarnign Maat-g ser implementada sobre un ambiente de alta disponibilidad, brindando servicio ininterrumpido, aún en caso de fallo”***

Cada uno de sus componentes propuestos se describen en detalle a continuación:

29 Vargas, Javier. High Availability Fundamentals. Sun Enterprise Engineering. Sun Blue Prints. 2002.

30 Availability Digest. Disponible en internet: <http://www.availabilitydigest.com/>

31 Understanding High availability Configuration examples. Disponible en internet: <http://docs.sun.com/app/docs/doc/820-3746/givdo?a=view>

3.2.1.1 Cluster para balanceo de canales

La presente arquitectura contempla la instalación de un Cluster de balanceadores de tráfico, encargados de manejar 2 enlaces hacia internet de 2 proveedores distintos, logrando así tener salidas redundantes y a la vez balanceando peticiones para no tener canales inactivos. El Cluster de balanceadores se deben configurar en alta disponibilidad, es decir en takeover lo cual significa que uno de ellos de manera aleatoria se marca como principal y el otro como respaldo, cada n cantidad de segundos se envían mensajes de vida y si no se contestan por lo menos 3 de ellos, se declara caído el balanceador principal, el servidor secundario se transforma en servidor principal y el principal se marca como fuera de línea hasta que pueda ser iniciado correctamente.

Los elementos aquí utilizados por su criticidad, compatibilidad y desempeño no fueron elegidos dentro de soluciones de software libre, sino directamente de fabricantes de hardware los cuales se describen a continuación:

3.2.1.2 Soluciones de hardware:

- Cisco: Posee equipos de frontera capaz de balancear conexiones hacia los ISP dentro de los cuales se encuentran: Equipos CISCO ASA y CISCO PIX 7.0 en una configuración de backup router, también puede configurarse

equipos CISCO ASA serie 5500 en configuración Active-Active Failover.

Una guía completa es suministrada por el fabricante³².

- F5. Sus equipos proveen capacidad de balanceo de enlaces en redundancia para la serie BIG-IP link controller, y redundancia entre centro de datos con la serie BIG-IP 6800. La guía completa de productos se puede encontrar es suministrada por el fabricante³³.

- Juniper. Posee equipos capaces de balancear enlaces y mantener alta disponibilidad al ser configurados en redundancia, en los equipos de la serie WXC, información más detallada es suministrada por el fabricante³⁴.

3.2.1.3 Soluciones de software con hardware de comodidad

Para realizar balanceo de enlaces de diferentes ISP se requiere un servidor con dos tarjetas de red que permitan conectar cada una a un enlace diferente y los siguiente requisitos en cuanto a módulos de Netfilter:

- CONMARK

32 PIX/ASA: Active/Active Failover Configuration Example. Disponible en internet:

http://www.cisco.com/en/US/products/ps6120/products_configuration_example09186a0080834058.shtml

33 AVAILABILITY SOLUTIONS. Disponible en internet: <http://www.f5.com/solutions/availability/>

34 WXC series portfolio. Disponible en internet: <http://www.juniper.net/us/en/products-services/application-acceleration/wxc-series/>

- nth
- condition
- Iproute2

Con este tipo de software se puede construir un conjunto de reglas de de routing para realizan balanceo en la salida de los paquetes para que cada uno de ellos sera dirigido por una interfaz diferentes. El proceso de instalación y configuración se describe en el Anexo No 1. Sección. [5.5 INSTALACION DE BALANCEADOR DE ENLACES CON LINUX IPFILTER](#)

Para la presente propuesta se propone un balanceador de enlaces por hardware que además haga compresión y caché de contenido servido.

3.2.1.4 Cluster de servidores Linux LVS.

Este Cluster de servidores tienen como función exponer una única dirección IP capaz de balancear las peticiones hacia una granja de servidores de tal manera que cada nueva petición pueda ser balanceada y en caso de dañarse o quedar inaccesible un miembro de la granja, el Cluster lo marcará como no disponible, no enviará peticiones a él sin que el cliente perciba su caída.

Los miembros de este Cluster están configurados en alta disponibilidad con la solución de software libre HeartBeat encargada de establecer una dominio de alta disponibilidad, en donde un servidor principal continuamente es monitoreado por un servidor de respaldo que se comunican por una única interfaz virtual

denominada interfaz de servicio, asociada a interfaces de red físicas. El proceso de instalación y configuración es descrito en el Anexo 1. Sección [5.6](#)
INSTALACION CLUSTER DE SERVIDORES BALANCEADOS CON LVS.

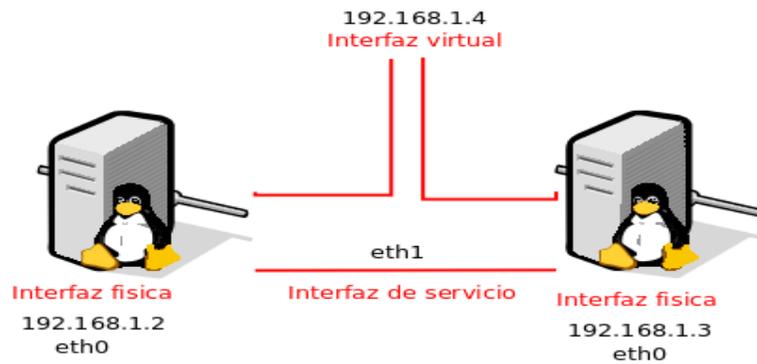


Gráfico 1: Modelo HeartBeat para Alta Disponibilidad

3.2.1.5 Arquitectura general de la solución.

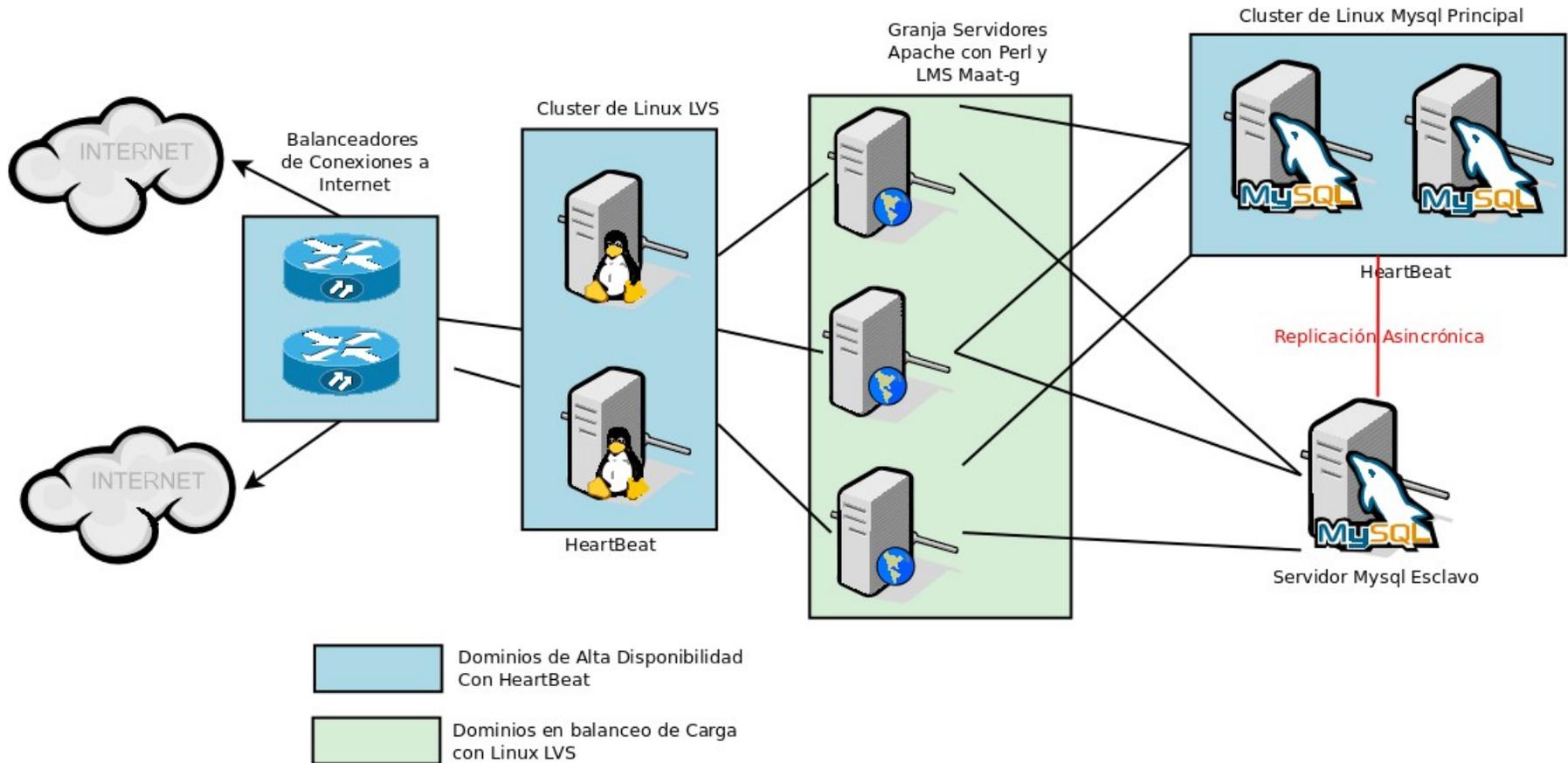


Gráfico 2: Arquitectura General de la solución

3.2.1.6 Esquema de direccionamiento de la solución

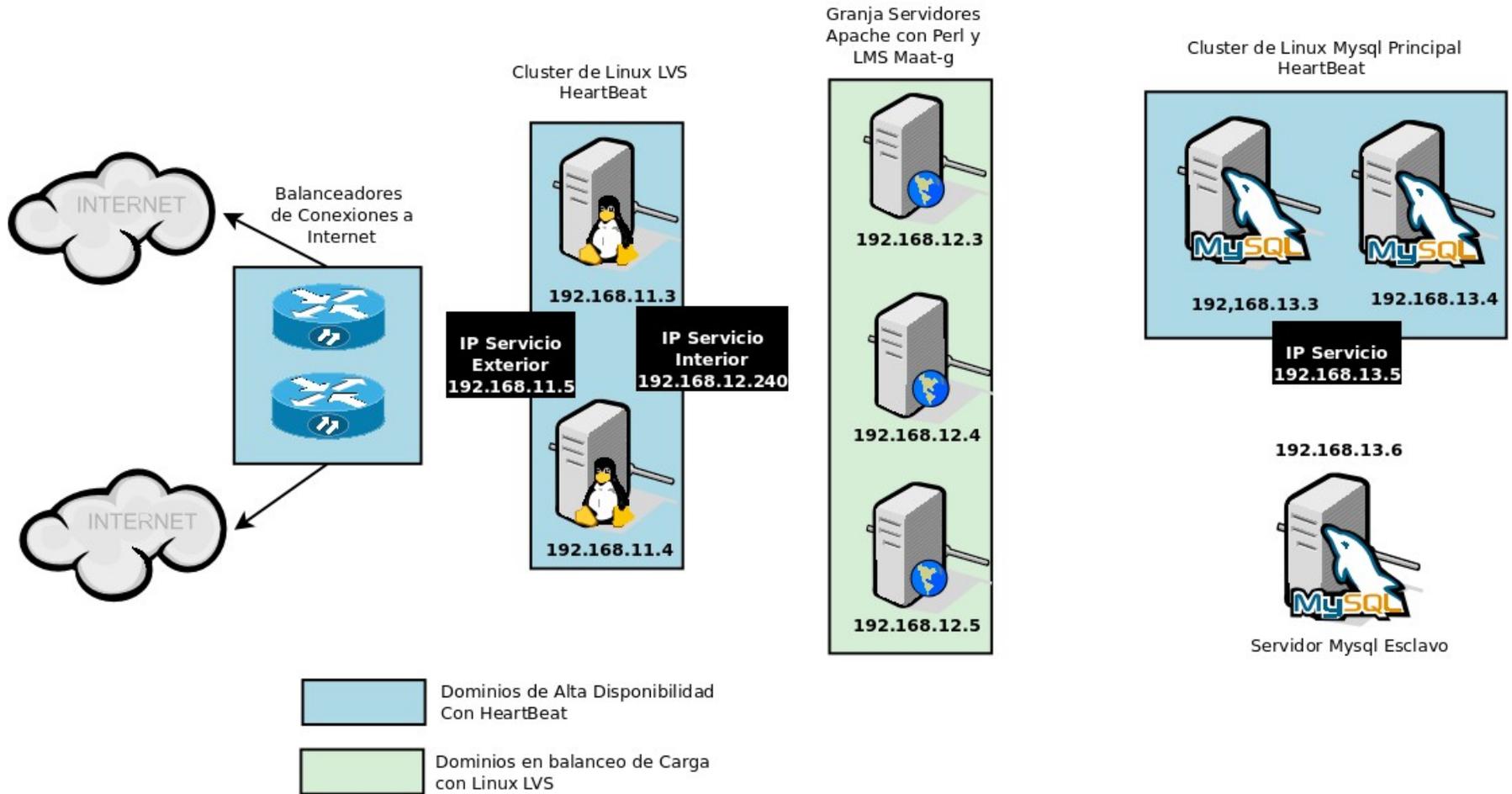


Gráfico 3: Esquema de direccionamiento de la solución

La alta disponibilidad del balanceador es brindada por el modelo HeartBeat. El procedimiento de configuración es descrito en el Anexo No 1. Sección [5.3 CONFIGURACIÓN DEL CLUSTER LVS CON HEARTBEAT](#)

3.2.1.7 Granja de Servidores Apache, Perl y LMS Maat-g.

Se configuran por lo menos 2 servidores Linux, con servidor Web Apache, módulos de perl y la aplicación LMS de Maat-g de acuerdo al Anexo No 1. Sección [5.2 INSTALACIÓN Y CONFIGURACIÓN PLATAFORMA MAAT-G SOBRE SISTEMAS OPERATIVO CENTOS OS](#)

3.2.1.8 Cluster de servidores Mysql

Se configuran 3 servidores con Mysql versión 5, de la siguiente manera.

2 servidores se configuran en alta disponibilidad con la configuración principal utilizando un HeartBeat.

Un segundo servidor se configura como esclavo y en el será activada la replicación desde el servidor principal, además tendrá la función de atender las consultas que no involucran actualización.

Su configuración es descrita en el Anexo No 1. Sección [5.4 INSTALACIÓN DE CLUSTER MYSQL CON REPLICACIÓN](#)

3.3 INSTALACIÓN DE UN AMBIENTE SIMULADO PARA REALIZAR UNA PRUEBA DE CONCEPTO DE LA ARQUITECTURA PROPUESTA.

Como se puede observar la arquitectura propuesta incluye 2 servidores por capa como mínimo, para un total de 6 servidores y 2 balanceadores de canal. El costo

de implementar una arquitectura como la propuesta supera los US \$ 150.000, según cotizaciones realizadas por proveedores de hardware, por tal motivo fue necesario crear un ambiente de simulación basado en máquinas virtuales sobre Linux.

3.4 REALIZACIÓN DE PRUEBAS DE CARGA PARA MEDIR LA DISPONIBILIDAD Y EL COMPORTAMIENTO DE LA SOLUCIÓN.

Para realizar de estas pruebas se utilizó el software de fuente abierta Jmeter³⁵, mediante el cual se construyó un conjunto de pruebas de acceso al portal de e-learning MAAT-g, instalado en la granja de servidores y se realizaron las mediciones de tiempo de respuesta, número de peticiones y número de errores.

Además se realizaron tomas de muestras de procesador, memoria y disco en cada uno de los servidores de la granja que soporta el aplicativo para determinar su comportamiento en ambientes de carga.

La realización del conjunto de pruebas y los resultados se describen en el Anexo No. 1. Sección [5.7 REALIZACIÓN DE PRUEBAS DE DISPONIBILIDAD.](#)

3.4.1 Resultados de las pruebas

Teniendo el ambiente configurado, se realizaron las siguiente pruebas para determinar el nivel de disponibilidad de la plataforma.

1. Caída de 1 Nodo del Cluster LVS.

³⁵ Disponible en internet: <http://jakarta.apache.org/jmeter/>

2. Caída de 1 Nodo de la granja de servidores apache, perl y LMS.
3. Caída de 1 Nodo del Cluster de Mysql

Los resultados se presentan a continuación.

3.4.1.1 Caída de 1 Nodo del Cluster LVS

El tiempo de migración de los servicios al nodo de respaldo fue de 45 segundos, por tanto el tiempo de indisponibilidad que se agregará a cualquier SLA propuesto, será de 1 minuto 40 segundos. Bajo este escenario para cumplir con un nivel de disponibilidad de 99.96% (3 horas 30 minutos), el número máximo de migración de servicios entre nodos del LVS teóricamente sería de 150 veces, sin embargo algunos SLA incluyen elementos como número máximo de veces que un nodo puede caerse, lo cual haría menor este límite.

3.4.1.2 Caída de 1 Nodo de la granja de servidores apache, perl y LMS.

En esta prueba no existen tiempo de indisponibilidad, pues virtualmente la IP balanceada continúa respondiendo, sin embargo aquellas transacciones que se estaban ejecutando en el nodo que se cae se pierden y el LVS tarda aproximadamente 25 segundos en darse cuenta que el nodo no responde, por tanto se toman estos 25 segundos como tiempo de indisponibilidad.

3.4.1.3 Caída de 1 Nodo del Cluster de Mysql

El tiempo de migración de los servicios de Mysql de un nodo a otro fue en total 3 minutos 20 segundos y puede aumentar con una carga de transacciones mayores, esto debido a que el nodo principal debe dejar todas las transacciones cerradas y una vez hace esta, realiza la sincronización contra el nodo esclavo. Este proceso

asegura la consistencia de los datos. Cuando el nodo se cae inesperadamente los servicios se migran, pero el nodo de respaldo debe realizar la misma comparación de consistencia antes de iniciar los servicios y dejarlos operativos. Con esta perspectiva el número máximo de migraciones de nodo de Mysql es de 60, para cumplir con los SLA propuestos, sin embargo por tratarse de un nodo transaccional lo recomendable es no exceder los 15 al año.

Es de anotar que las pruebas se realizaron con 8000 registros en la base de datos.

Los resultados obtenidos confirman que la arquitectura propuesta no solo cumple con objetivos académicos, sino también cumple con los requerimientos que la industria en la actualidad exige, cumpliendo así el objetivo de ***“Brindar una solución de grado empresarial utilizando las herramientas de software libre existentes y de esta manera crear un antecedente a futuros trabajos para que puedan generar componentes que puedan ser operativos dentro del mundo empresarial”***

3.5 CRECIMIENTO DE LA PLATAFORMA PROPUESTA.

La plataforma propuesta asume una carga de 300 usuarios concurrentes, como se observa en las pruebas de disponibilidad efectuadas. 300 usuarios concurrentes podría equivaler a 3000 usuarios del sistema utilizando la regla informal que las empresas productoras de hardware utilizan estimando un 10% del total de usuarios del sistema por unidad de tiempo, como la concurrencia esperada en un instante de tiempo.

Por tanto las proyecciones se realizaron sobre la base de 3000 usuarios/día en el sistema con una concurrencia de 300 usuarios en un momento dado.

Existen modelos más formales para estimar la concurrencia esperada como el propuesto por Wong, Eric Man³⁶, sin embargo la industria ha comprobado que solo a través de mediciones se puede lograr realizar una estimación exacta.

El aumento por tanto en la cantidad de usuarios y por ende de la concurrencia es un factor importante en las plataformas altamente disponibles, por lo cual normalmente se acompañan con planes de crecimiento siguiente algunas reglas generales como las siguiente:

- En caso que cualquier servidor mantenga un uso superior al 80% en procesador y 90% de memoria en periodos de tiempo largos (mayores a ½ Hora), es necesario agregar un equipo adicional a la solución.
- En caso que el almacenamiento del servidor llegue a una ocupación del 90% o superior, es necesario aumentar la capacidad de almacenamiento, implementar una política de backups más eficiente o integrarlo dentro de una solución de Almacenamiento centralizado (SAN, NAS o DAS)³⁷.
- En caso que el tráfico de red supere el 85% de la capacidad del canal de manera sostenida durante periodos de operación largos (½ hora o más) se debe aumentar la capacidad del mismo o realizar un análisis de tráfico para determinar si todo el tráfico que se está produciendo es legítimo o puede tener origen en malfuncionamientos o configuraciones de los equipos de red.

36 Wong, Eric Man. Method for Estimating the Number of Concurrent Users. Disponible en internet http://prathaputility.googlepages.com/Concurrent_Users_Estimation.pdf

37 SACKS, David. Demystifying Storage Networking. IBM. Disponible en internet: <http://www-03.ibm.com/industries/ca/en/education/k12/technical/whitepapers/storagenetworking.pdf>

Todos estos elementos normalmente son incluidos en un plan de capacidad que se realiza a petición de los clientes para determinar la salud de la plataforma o el momento en que se debe planear el crecimiento.

Cada fabricante tiene su propio modelo para realizar sus planes de capacidad, sin embargo un modelo general que se puede evaluar el el propuesto por TeamQuest³⁸, el cual no se centra en el hardware utilizado sino en métricas de la plataforma, lo cual permite tener una visión completa de la arquitectura y no tan solo de algunos elementos.

Con la ejecución de un plan de capacidad, se logra cumplir el objetivo de **“Desarrollar un modelo de crecimiento que permita crecimiento tanto en concurrencia como en volúmenes de datos y que pueda integrar diferentes fabricantes de hardware.”** Dicho plan debe ser ejecutado cuando las condiciones sobre las cuales el presente proyecto se ha realizado cambien considerablemente (que el incremento sea mayor al 30%) tanto en concurrencia, almacenamiento o tráfico de red.

38 How to Do Capacity Planning. White Paper. TeamQuest. Disponible en internet:

www.teamquest.com/pdfs/whitepaper/tqwp23.pdf

4 CAPITULO III

4.1 CONCLUSIONES

- La madurez que ha tenido el software libre y de fuente abierta en la última década permiten que gran parte de la industria pueda confiar su desarrollo y operación a plataformas basadas en él, logrando desempeños excepcionales en algunos casos, incluso con niveles superiores al software propietario.
- Las soluciones de alta disponibilidad que se tienen dentro del software libre y de fuente abierta pueden competir abiertamente en sectores que requieran niveles de disponibilidad de hasta 99.99%; niveles superiores requieren un delicado balance entre hardware, software y servicios especializados.
- Las plataformas de e-learning cada vez más se verán enfrentadas a ambientes con cientos de miles de usuarios, lo cual obligará a que sus arquitecturas migren a sistemas altamente disponibles capaces de atender la oferta creciente de educación virtual.
- La utilización de métodos formales para estimar la capacidad real de un sistemas es un requisitos fundamental para crear arquitecturas capaces de soportar los ambientes muchas veces agresivos de la red internet. Los planes de capacidad y crecimiento deben formar parte de cualquier diseño que sobre un servicio expuesto en la red se hagan. La creación de dichos modelos y métodos es un área de muy poco desarrollo nacional a pesar de ser una de las necesidades evidentes de la industria.

- Las posibilidades que el software libre ofrece para crear, implementar e integrar soluciones en el ámbito tecnológico son ilimitadas, por lo tanto se deben crear organizaciones académicas e industriales capaces de proponer dichos modelos y probarlos en ambientes de competencia.
- Las pruebas realizadas confirman que la arquitectura presentada cumple con los requerimientos de alta disponibilidad (99.9%) e incluso logra niveles superiores (99.96%) aún utilizando servidores y redes virtualizados.
- Cualquier arquitectura considerada de alta disponibilidad puede ser sometida al mismo conjunto de pruebas, para determinar su disponibilidad real de tal manera que los modelos y las configuraciones ofrecidas por diferentes proveedores de hardware tengan un procedimiento de medición real y no tan solo descripciones indicando que lo logran.

4.2 RECOMENDACIONES

- Realizar dentro de la maestría un esfuerzo para que se generen grupos de investigación especialmente en software libres capaces de planes de acción en la industria nacional y apoyar los procesos de implementación de tecnologías de la información. Dichos grupos deberían poseer centros de competencia en donde la industria exponga sus necesidades y la academia aporte el conocimiento generado en pro de lograr modelos exitosos a nivel científico y empresarial.
- Generar las alianzas necesarias dentro de la maestría para lograr que sus egresados y estudiantes puedan acceder a certificaciones de clase mundial con niveles preferenciales dando valor agregado a cada uno de sus egresados y además generando redes profesionales sólidas de reconocimiento nacional e internacional.
- Realizar los enlaces necesarios entre los fabricantes de hardware, empresas de integración y comunidades de software libre para generar eventos de gran impacto, semejante al generado por el 4CCC o en su defecto crear un capítulo especializado en el tema, donde los proyectos de investigación, casos de éxito y productos culminados exitosamente puedan ser expuestos a la comunidad nacional e internacional.
- Ahondar los esfuerzos para que los egresados de la maestría puedan tener espacios de participación dentro del desarrollo de la misma, realizando retroalimentación periódica de los logros obtenidos y las experiencias ganadas a fin que la comunidad de especialistas en software libre se consolide a nivel técnico y científico.

10. BIBLIOGRAFIA

BADER, David. PetaScale Computing. Chapman & Hall/CRC. 2008.

BISCHOF C y BUCKER M. Parallel Computing: Architectures, Algorithms and Applications. IOS Press 2008.

BOURKE , Tony. Server Load Balancing. Sebastopol, CA . O'Reilly & Associates, Inc. 2001.

CHAPMAN, Barbara y JOST Gabrielle. Using OpenMP. Portable Shared Memory Parallel Programming. MIT 2008.

DONGARRA, Jack; FOSTER Ian . Sourcebook of Parallel Computing. Morgan Kaufmann Publishers. 2003.

GROPP William, LUSK Ewing . Using MPI-2: Advanced Features of the Message-passing Interface.

HARIRI Salim, PARASHAR Manish. Tools and environments for parallel and distributed computing. A John Wiley & sons, inc., publication. 2004.

HERLYHY, Mauricie. The Art of Multiprocessor programming. Morgan Kaufmman. 2008

HUGHES, Cameron, HUGHES, Tracey. Professional MultiCore Programming, Design and Implementation for C++ Developers. Wrox Press. 2008.

HUGHES Cameron Hughes, HUGHES Tracey . Parallel Programming using C++. Addison-Wesley Professional. 2003.

LEVNER, Eugene. Multiprocessor Scheduling: Theory and Applications I-TECH Education and Publishing. 2007.

MEHRA Pankaj. Benjamin W. Load Balancing: An Automated Learning Approach.

Singapore. World Scientific Publishing Company. 1995.

SANDERS Beverly y MATTSON, Timothy. Pattern for Parallel programming software. Addison-Wesley, 2005

SHAMEEN Akhter, JASON Robert. MultiCore Programming, Intel Press. 2007

REINDERS, James. Intel Threading Building Blocks. O'Reilly. 2006

RAHMAN As Abdel, AJEEL Ab d Al. Simulation and Modeling: Current Technologies and Applications. IGI Publishing. 2008.

WEINSTEIN, Paul. Pro Apache 2.0 Module Development: From Professional to Expert. APress,US. Diciembre 2004.

5 ANEXO NO. 1

5.1 MANUAL DE CONFIGURACIÓN E INSTALACIÓN PLATAFORMA MAAT-G.

Convenciones:

Todos los comando que se ejecuten sobre la linea de comando se distinguen por estar en fuente Courier New 10 y por tener un fondo gris

```
ls -lta
```

Las anotaciones y advertencias se por estar fuente Times New Roman 13 y fondo de color según cada caso, con un símbolo ante puesto.

Advertencia

⚠ *Advertencia:*

Anotación

▶ *Anotación:*

5.2 INSTALACIÓN Y CONFIGURACIÓN PLATAFORMA MAAT-G SOBRE SISTEMAS OPERATIVO CENTOS OS

5.2.1 PREREQUISITOS

5.2.1.1 Requerimientos Mínimos de Hardware:

- Equipo x86, con procesador Pentium III o superior
- 512 MB de Memoria RAM
- 4GB mínimos de Espacio en Disco Duro disponible
- Dispositivo de video con resolución mínima de 1024 x 768
- conexión a Internet (512 Kbps sugerido)

5.2.1.2 Requerimientos Mínimos de Software:

- Sistema Operativo de Base
- Linux kernel 2.6 o superior
- Linux Centos O.S 5.1 o Superior (o RedHat EL 5x)
- Compilador GCC versión 3.4 o superior
- Librerías Libgc, libgcc,

► *Este manual fue elaborado y probado con la versión Linux Centos OS v. 5*

5.2.1.3 Proceso de instalación de los productos

..

La presente solución incluye los siguientes productos:

- Perl 5.1 con los siguientes módulos
 - XML::Simple
 - XML::SAX
 - URI::Escape
 - DBI
 - SOAP::Lite
 - CSS::Tiny
 - File::Spec
 - File::Temp
 - Net::Daemon
 - Term::ReadKey
- Apache 2
- Version de G más reciente.

En la versión Centos la mejor manera de realizar la instalación es utilizando el cliente yum, encargada de los procesos administrativos y para este caso de los procesos de instalación de software:

```
yum install httpd
```

Ante lo cual el sistema mostrará el conjunto de dependencias necesarias para instalar el servidor apache http, como se muestra en la siguiente figura

```
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* addons: pubmirrors.reflected.net
* base: hpc.arc.georgetown.edu
* extras: centos.mirror.nac.net
* updates: hpc.arc.georgetown.edu
Setting up Install Process
Resolving Dependencies
```

```
--> Running transaction check
---> Package httpd.i386 0:2.2.3-31.el5.centos.2 set to be updated
--> Finished Dependency Resolution
```

Dependencies Resolved

```
=====
Package      Arch      Version      Repository
Size
=====
Updating:
httpd        i386      2.2.3-31.el5.centos.2  updates
1.2 M
```

Transaction Summary

```
=====
Install      0 Package(s)
Update       1 Package(s)
Remove       0 Package(s)
```

Total download size: 1.2 M

Is this ok [y/N]:Y

Downloading Packages:

Running rpm_check_debug

Running Transaction Test

Finished Transaction Test

Transaction Test Succeeded

Running Transaction

```
  Updating      : httpd
1/2
```

```
  Cleanup       : httpd
2/2
```

Updated:

```
  httpd.i386 0:2.2.3-31.el5.centos.2
```

Complete!

Una vez instalado el servidor apache, se requiere la instalación de perl y los módulos de perl para apache, de la siguiente manera:

```
yum install -y perl
yump install -y mod_perl
```

Una vez instalados estos paquetes ya se tendrá los requisitos necesarios para realizar la configuración de módulos requeridos por Maat-g.

5.2.1.4 INSTALACION DE MODULOS DE PERL

Una vez se ha instalado correctamente el PERL, se utilizará CPAN para instalar los módulos requeridos para la plataforma G, de la siguiente manera:

1. Entrar en el shell de CPAN

```
perl -MCPAN -e shell
CPAN is the world-wide archive of perl resources. It consists of about
300 sites that all replicate the same contents around the globe. Many
countries have at least one CPAN site already. The resources found on
CPAN are easily accessible with the CPAN.pm module. If you want to use
CPAN.pm, lots of things have to be configured. Fortunately, most of
them can be determined automatically. If you prefer the automatic
configuration, answer 'yes' below.

If you prefer to enter a dialog instead, you can answer 'no' to this
question and I'll let you configure in small steps one thing after the
other. (Note: you can revisit this dialog anytime later by typing 'o
conf init' at the cpan prompt.)
Would you like me to configure as much as possible automatically? [yes]
...
Autoconfigured everything but 'urllist'.
Please call 'o conf init urllist' to configure your CPAN server(s) now!

commit: wrote '/home/cidel/.cpan/CPAN/MyConfig.pm'
Terminal does not support AddHistory.

cpan shell -- CPAN exploration and modules installation (v1.9205)
ReadLine support available (maybe install Bundle::CPAN or
Bundle::CPANxxl?)
cpan[1]>
```

Una vez el CPAN han sido configurado adecuadamente necesitamos instalar cada uno de los módulos requeridos, mediante la orden `install nombreModulo`, por ejemplo:

```
cpan[1]> install XML::Simple
CPAN: Storable loaded ok (v2.18)
CPAN: LWP::UserAgent loaded ok (v5.810)

I would like to connect to one of the following sites to get
'authors/01mailrc.txt.gz':

  http://www.perl.org/CPAN/
  ftp://ftp.perl.org/pub/CPAN/

Is it OK to try to connect to the Internet? [yes]
Fetching with LWP:
  http://www.perl.org/CPAN/authors/01mailrc.txt.gz
Going to read /home/cid1/.cpan/sources/authors/01mailrc.txt.gz
.....
...DONE
Fetching with LWP:
  http://www.perl.org/CPAN/modules/02packages.details.txt.gz
Going to read /home/cid1/.cpan/sources/modules/02packages.details.txt.gz
  Database was generated on Wed, 13 May 2009 01:29:35 GMT
.....
  New CPAN.pm version (v1.9304) available.
  [Currently running version is v1.9205]
  You might want to try
    install CPAN
    reload cpan
  to both upgrade CPAN.pm and run the new version without leaving
  the current session.

.....DONE
Fetching with LWP:
  http://www.perl.org/CPAN/modules/03modlist.data.gz
Going to read /home/cid1/.cpan/sources/modules/03modlist.data.gz
.....
...DONE
Going to write /home/cid1/.cpan/Metadata
XML::Simple is up to date (2.18).
```

Como se puede observar al final, el módulo `XML::Simple` ha sido instalado correctamente, este procedimiento debe ser repetido para cada uno de los siguiente módulos requeridos.

- XML::SAX
- URI::Escape
- DBI
- SOAP::Lite
- CSS::Tiny
- File::Spec
- File::Temp
- Net::Daemon
- Term::ReadKey

Cada instalación debe terminar con la frase

```
módulo is up to date .
```

Si no es así el modulo no se instaló correctamente y habrá que realizar un clean y make adicional.

5.2.2 INSTALACION Y CONFIGURACION DE PLATAFORMA MAAT-G SOBRE CENTOS

5.2.2.1 Pre-requisitos:

Antes de instalar la plataforma se deben realizar algunos ajustes en el sistema, que a continuación se nombra: En primer lugar se debe ajustar los siguientes parámetros del usuario dueño del proceso de apache.

Para revisar el nombre de usuario dueño del proceso, se debe editar el archivo

```
/etc/http/conf/httpd.conf.
```

En donde se puede encontrar la siguiente información por defecto

```
User  apache  
Group apache
```

Ahora el usuario apache se le debe asociar el shell de bash, de la siguiente manera:

con privilegios de root

```
usermod -s /bin/bash apache
```

y luego se puede comprobar mediante

```
grep apache /etc/passwd
```

y el resultado debe ser algo semejante a

```
apache:x:48:48:Apache:/var/www:/bin/bash
```

Una vez realizado este cambio, se procede a realizar los ajustes necesarios dentro del archivo de configuración del servidor apache, localizados en

```
/etc/http/conf/httpd.conf
```

y reemplazar la línea

```
LanguagePriority en cs de es fr it ja ko nl pl pt-br ro sv tr
```

por la línea

```
LanguagePriority es en cs de es fr it ja ko nl pl pt-br ro sv tr
```

Se debe revisar además que exista soporte para juego de caracteres latinos, mediante la siguiente línea dentro del archivo httpd.conf

```
AddCharset ISO-8859-1 .iso8859-1 .latin1
```

Una vez estos ajustes se han realizado, es necesario reiniciar los servicios de apache, con privilegios de superusuario, de la siguiente manera

```
service httpd stop  
service httpd start
```

o

```
/etc/init.d/apache2 restart
```

5.2.2.2 Instalación Base de Plataforma G.

para esto se debe crear una carpeta donde se copie los instaladores de la plataforma G.

para nuestros caso

```
~/instaladores/
```

En donde debe quedar la versión más reciente del instalador en extensión .tgz, y del archivo de instalación VGLinuxInstaller.pl

No se puede utilizar la extensión .tar.gz, el instalador exige .tgz

Una listado de la carpeta debería verse como el siguiente:

```
File Edit View Scrollback Bookmarks Settings Help
eshernan@linux-0cny:~/instaladores> ls
httpd.conf InstallInfo.lib rc.local vg_actual vg_actual.tgz VGLinuxInstaller.pl
eshernan@linux-0cny:~/instaladores> ls -lta
total 2548
-rw-r--r-- 1 eshernan users 2503379 2009-09-30 11:48 vg_actual.tgz
drwxr-xr-x 3 eshernan users 4096 2009-09-30 11:48 .
-rwxr-xr-x 1 eshernan users 22044 2009-09-30 11:41 VGLinuxInstaller.pl
drwxr-xr-x 3 eshernan users 4096 2009-09-30 11:41 vg_actual
-rwxr-xr-x 1 eshernan users 35073 2009-09-30 11:41 httpd.conf
-rwxr-xr-x 1 eshernan users 64 2009-09-30 11:41 InstallInfo.lib
-rwxr-xr-x 1 eshernan users 334 2009-09-30 11:41 rc.local
drwxr-xr-x 30 eshernan users 4096 2009-09-30 11:37 ..
eshernan@linux-0cny:~/instaladores> █
```

Gráfico 4: Contenido del instalador de Maat-g

Ahora con privilegios de root corremos el instalador de la siguiente manera:

```
perl VGLinuxInstaller.pl
```

Una vez de lanza, el instalador realiza una validación previa y luego solicita el listado de opciones

```
#####  
User is OK.  
#####  
Please, select one of the targzipped files found in l  
ted below.  
[0]: I don't know which's the correct file.  
[1]: VG (version 2.8.8.5)  
Choose a file to install: 1█
```

Gráfico 5: Elección de la versión de Maat-g

En este caso se elige la versión del instalador que se tenga, el sistema realiza una serie de validaciones, indicando si todos los módulos de perl, requeridos se encuentran presentes y en caso contrario mostrando cuál hace falta, como aparece en la siguiente figura:

```
VG LINUX INSTALLER is going to install vg_actual.tgz
#####
[Ok]:Module XML::Simple(*).
[Ok]:Module XML::SAX.
[Ok]:Module URI::Escape.
[Ok]:Module DBI.
[Ok]:Module SOAP::Lite.
[Ok]:Module CSS::Tiny.
[Ok]:Module File::Spec.
[Ok]:Module File::Temp.
[Ok]:Module Net::Daemon.
[Ok]:Module Term::ReadKey.

All required modules are installed!
#####
Which's Apache's config file? [httpd.conf]:
```

Gráfico 6: Validación de los módulos de Perl requeridos

Si el paso 2, de este manual se ha ejecutado correctamente, entonces el resultado debe ser igual al mostrado en la figura anterior. De manera seguida solicita confirmar el nombre del archivo de apache que se buscará para aplicar los cambios solicitados, que para el caso de Centos es el httpd.conf, como aparece en la siguiente figura

```
All required modules are installed!
#####
Which's Apache's config file? [httpd.conf]:uid.conf
```

Gráfico 7: Archivo en que se hacen los cambios para Maat-g

El instalador realiza una búsqueda del archivo y muestra las posibles coincidencias para que se elija una de ellas

```
Choose a file: @
Processing file: /etc/apache2/uid.conf [OK]
#####
VG LINUX INSTALLER is now ready to install in your sys
Please, check the information below:
Directory to install application (Apache's DocumentRoot):
Apache's User: apache
Apache's Group: apache
Is this information correct?: █
```

Gráfico 8: Confirmación de parámetros de actualización

Una vez se confirma esta instalación se determina que la instalación se ha realizado con éxito, como se muestra en la siguiente pantalla.

```
#####
VG (VERSION 2.8.8.5) HAS BEEN INSTALLED
#####
Now application needs to modify Apache configuration file
These changes are:
1.- Set directive FilesMatch to protect G files (those ending with '.db, .xml.cond, .gtd, .rls
wfl, .prv') from being browseable.
2.- Set directive ScriptAlias or AddHandler to allow Apache run .cgi scripts.
3.- Configure directory /vg_actual/gestion/apps/ to allow run .cgi script from there
4.- Set directive DirectoryIndex to use index.htm as default index file.
What do you wish to do?
[1] Let VG LINUX INSTALLER make changes for you.
[2] Make changes yourself.
[3] Apache is configurated correctly.
Your selection:2
Remember to make this changes to /etc/apache2/uid.conf:
1.- Use directive FilesMatch to protect G files (those ending with '.db, .xml.cond, .gtd, .rls
wfl, .prv') from being browseable.
2.- Use directive ScriptAlias or AddHandler to allow Apache run .cgi scripts.
3.- Configure directory //gestion/apps/ to allow run .cgi script from it
4.- Use directive DirectoryIndex to use index.htm as default index file.
and restart Apache
```

Gráfico 9: Confirmación de la instalación de Maat-g Base

Una vez terminada esta fase se procede a la fase de configuración inicial de apache con soporte para G.

5.2.2.3 Post-instalación de la plataforma g.

Una vez realizada la instalación se debe revisar la configuración de la carpeta raíz, en donde se colocaran todos los archivos de la configuración (DocumentRoot). En el caso de Suse esta información se encuentra localizada en

```
/etc/httpd/conf/http.conf
```

y por defecto se encuentra dirigido a

```
DocumentRoot "/srv/www/htdocs"
```

Como la ruta solicitada por g está definida al directorio /var/www/html, se debe crear mediante el siguiente comando con privilegios de root

```
$> sudo mkdir -p /var/www/html
```

Ahora se debe cambiar la configuración en /etc/apache2/default-server.conf por la requerida por la plataforma G,

```
DocumentRoot "/var/www/html"
```

Una vez creado se deben asignar al dueño de la carpeta y los permisos de escritura y ejecución de la misma mediante el siguiente comando

```
$> sudo chmod -R 755 /var/www/html/*  
$> sudo chown -R apache:apache /var/www/html/
```

Se debe cambiar dos `parametros` en el archivo `gService`, debido a que la forma como se invoca el servicio de apache es diferente.

Para este caso se debe editar el archivo

```
/var/www/html/vg/admin/gService
```

y reemplazar las líneas

por las líneas

```
function fApacheStart()
{
    /usr/sbin/apachectl start
}

function fApacheStop()
{
    /usr/sbin/apachectl stop
}
```

Ilustración 1: Reemplazo de código para arranque Maat-g

Por último se debe lanzar el servidor de G, de la siguiente manera

```
/var/www/html/vg/admin/gService start
```

Si todos los pasos anteriores han sido correctamente ejecutados, la respuesta mostrada por el sistema debe ser similar a la siguiente figura.

```
[ Services summary ]
```

port	up/down	pid		service
2104	up	9366		gajax.pl
2343	up	9344		gvdb.pl
2345	up	9348		gcache.pl
2346	up	9342		ginsert.pl
2367	up	9460		gcluster.pl
2368	up	9351		gindexsrv.pl
2504	up	9339		gwflw.pl
3456	up	9346		gcachedg.pl
3457	up	9338		gvos.pl

```
[G!] Services connection test done.  
    Enjoy working with G.
```

Gráfico 10: Menú de Inicio del servicio Maat-g

5.3 CONFIGURACIÓN DEL CLUSTER LVS CON HEARTBEAT

Para activar la funcionalidad de HeartBeat, se deben instalar los siguiente paquetes:

```
HeartBeat-2.08
HeartBeat-pils-2.08
HeartBeat-stonith-2.08
```

los cuales pueden ser descargados con sus dependencias de la siguiente manera:

```
yum install HeartBeat
```

el sistema lanzará un listado de dependencias que serán resueltas como muestra a continuación:

```
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * addons: mirror.anl.gov
 * base: mirror.steadfast.net
 * extras: mirror.steadfast.net
 * updates: mirrors.combinetworks.com
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package HeartBeat.i386 0:2.1.3-3.el5.centos set to be updated
--> Processing Dependency: HeartBeat-stonith = 2.1.3-3.el5.centos for
package: HeartBeat
--> Processing Dependency: HeartBeat-pils = 2.1.3-3.el5.centos for
package: HeartBeat
--> Processing Dependency: libpils.so.1 for package: HeartBeat
--> Processing Dependency: libstonith.so.1 for package: HeartBeat
--> Running transaction check
---> Package HeartBeat-pils.i386 0:2.1.3-3.el5.centos set to be updated
---> Package HeartBeat-stonith.i386 0:2.1.3-3.el5.centos set to be
updated
--> Finished Dependency Resolution

Dependencies Resolved
```

```

=====
Package                               Arch      Version                               Repository
Size
=====
Installing:
HeartBeat                               i386      2.1.3-3.el5.centos                   extras
1.7 M
Installing for dependencies:
HeartBeat-pils                          i386      2.1.3-3.el5.centos                   extras
213 k
HeartBeat-stonith                       i386      2.1.3-3.el5.centos                   extras
311 k

Transaction Summary
=====
Install      3 Package(s)
Update      0 Package(s)
Remove      0 Package(s)

Total download size: 2.2 M
Is this ok [y/N]: y

```

ante lo cual se debe responder con Y para aceptar el listado de paquetes que serán instalados y entonces se realizará la descarga e instalación de los paquetes como se muestra a continuación:

```

Downloading Packages:
(1/3): HeartBeat-pils-2.1.3-3.el5.centos.i386.rpm | 213 kB
00:01
(2/3): HeartBeat-stonith-2.1.3-3.el5.centos.i386.rpm | 311 kB
00:01
(3/3): HeartBeat-2.1.3-3.el5.centos.i386.rpm | 1.7 MB
00:06
-----
Total                               220 kB/s | 2.2 MB
00:10
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing      : HeartBeat-pils
1/3

```

```
Installing      : HeartBeat-stonith
2/3
useradd: user haCluster exists
error: %pre(HeartBeat-2.1.3-3.el5.centos.i386) scriptlet failed, exit
status 9
error:  install: %pre scriptlet failed (2), skipping HeartBeat-2.1.3-
3.el5.centos

Installed:
  HeartBeat.i386 0:2.1.3-3.el5.centos

Dependency Installed:
  HeartBeat-pils.i386 0:2.1.3-3.el5.centos
  HeartBeat-stonith.i386 0:2.1.3-3.el5.centos

Complete!
```

Una vez instalados, el directorio de configuración principal se localiza en la ruta

```
/etc/ha.d/
```

Dentro de ellos encontramos los archivos

```
ha.cf
haresources
authkeys.
```

En primer lugar se debe configurar el archivo ha.cf, en donde se debe indicar los siguientes elementos:

```
logfile      /var/log/ha-log
```

indica el lugar donde se escribirán los logs,

```
keepalive 1
```

Indica el periodo de tiempo en el cual se realizará la revisión del nodo. Cada cuánto se revisará el estado del nodo.

```
Deadtime 3
```

Indica el periodo después del cual se declara que el nodo está abajo.

```
initdead 4
```

Indica el tiempo después de declarar el nodo abajo en que se iniciará el traslado del servicio al nodo de respaldo.

```
udpport 694
```

Puerto por el el cual escuchará el HeartBeat

```
bcast eth1
```

Indica la interface de red por la cual todos los mensajes de broadcast serán enviados entre los nodos del HeartBeat.

```
auto_failback on
```

Indica que una vez el nodo principal se recupera de la falla, el servicio vuelve a pasar del nodo secundario al primario de manera automática.

```
node lvs1  
node lvs2
```

Indica cada uno de los nodos que componen el dominio de HeartBeat. En esta configuración se relaciona a los nombres, los cuales deben ser resueltas ya sea por el archivo /etc/host o por el DNS.

El otro archivo que se debe configurar es el haresources, en donde se indica los servicios que estarán bajo HeartBeat y además se determina el nodo principal.

```
lvs1 10.20.1.5 lvs
```

En esta línea de indica que el nodo principal será el lvs1, que la dirección virtual por la cual se brindará el servicio será la 10.20.1.5 y que el servicio a respaldar será lvs.

En caso de requerir más de un servicio estos se debe colocar separados por espacios.

El servicio al que hace referencia el HeartBeat no se debe sumir en los nodos, ya que es el HeartBeat el encargado de manejarlos y exponerlos en la interfaz virtual indicada.

En el archivo authkeys, se debe colocar el método de autenticación entre nodos, por recomendación de seguridad se debe elegir el método sha1.

```
auth 2
#1 crc
2 sha1 test-HA
#3 md5 Hello!
```

Esta configuración debe ser igual en ambos nodos, para permitir que la comunicación se establezca.

Para que el servicio se active automáticamente una suba cada uno de los nodos, se debe activar en cada uno de los nodos mediante la orden

```
chkconfig --level 345 HeartBeat on
```

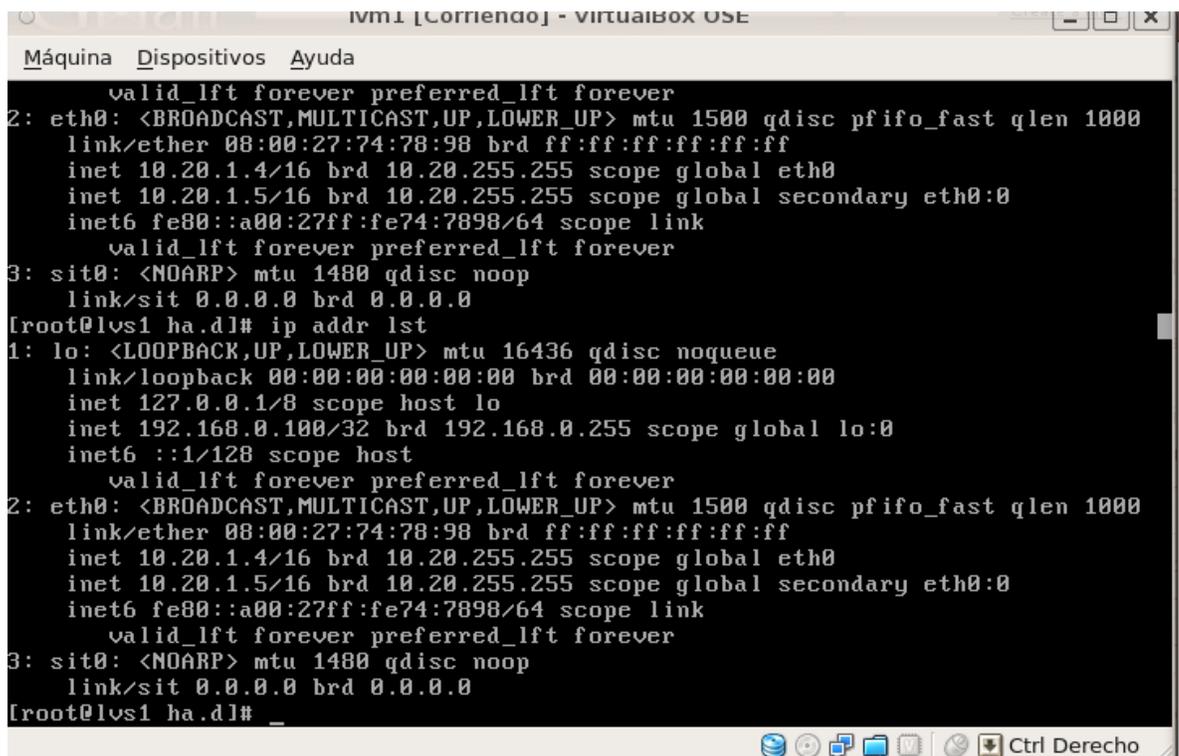
Se administra el servicio con la orden

```
service HeartBeat stop
service HeartBeat start.
```

0

```
/etc/init.d/HeartBeat stop
/etc/init.d/HeartBeat start
```

Una vez el HeartBeat se establezca se podrá apreciar la dirección virtual en el nodo establecido como primario, anexa al interfase de red indicada .



```
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
   link/ether 08:00:27:74:78:98 brd ff:ff:ff:ff:ff:ff
   inet 10.20.1.4/16 brd 10.20.255.255 scope global eth0
   inet 10.20.1.5/16 brd 10.20.255.255 scope global secondary eth0:0
   inet6 fe80::a00:27ff:fe74:7898/64 scope link
   valid_lft forever preferred_lft forever
3: sit0: <NOARP> mtu 1480 qdisc noop
   link/sit 0.0.0.0 brd 0.0.0.0
[root@lvs1 ha.d]# ip addr lst
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
   inet 192.168.0.100/32 brd 192.168.0.255 scope global lo:0
   inet6 ::1/128 scope host
   valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
   link/ether 08:00:27:74:78:98 brd ff:ff:ff:ff:ff:ff
   inet 10.20.1.4/16 brd 10.20.255.255 scope global eth0
   inet 10.20.1.5/16 brd 10.20.255.255 scope global secondary eth0:0
   inet6 fe80::a00:27ff:fe74:7898/64 scope link
   valid_lft forever preferred_lft forever
3: sit0: <NOARP> mtu 1480 qdisc noop
   link/sit 0.0.0.0 brd 0.0.0.0
[root@lvs1 ha.d]#
```

Gráfico 11: Heartbeat establecido en el nodo lvs1

Mientras que el segundo nodo mantendrá una única dirección como se muestra en el siguiente gráfico.

```
lv2 [Corriendo] - VirtualBox OSE
Máquina Dispositivos Ayuda
Jan 14 11:42:16 lvs2 heartbeat: [2196]: info: acquire local
y).
Jan 14 11:42:16 lvs2 heartbeat: [2196]: info: local HA resou
leted (standby).
Jan 14 11:42:16 lvs2 heartbeat: [2169]: info: Standby resour
[foreign].
Jan 14 11:42:16 lvs2 heartbeat: [2169]: info: Initial resour
ete (auto_failback)
Jan 14 11:42:17 lvs2 heartbeat: [2169]: info: remote resourc
ed.

[root@lvs2 ~]# ip addr lst
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pf
    link/ether 08:00:27:fe:82:a6 brd ff:ff:ff:ff:ff:ff
    inet 10.20.1.3/16 brd 10.20.255.255 scope global eth0
    inet6 fe80::a00:27ff:fefe:82a6/64 scope link
        valid_lft forever preferred_lft forever
3: sit0: <NOARP> mtu 1480 qdisc noop
    link/sit 0.0.0.0 brd 0.0.0.0
[root@lvs2 ~]#
```

Gráfico 12: Nodo de respaldo del HeartBeat

5.4 INSTALACIÓN DE CLUSTER MYSQL CON REPLICACIÓN

Para la base de datos MySQL se propone un esquema master-slave replicate, en donde toda la transaccionalidad está en la base de datos principal la cual después de un proceso de copia inicial, se replica de manera asincrónica hacia una base de datos esclava. La Base de datos esclava aplica todas las transacciones y mantiene una base de datos Standby lista para ser usada en caso de Failover sobre el nodo secundario, en donde

5.4.1 Instalación en cada Nodo

Una vez se tengan los binarios en rpm propios de la instalación, se debe correr el siguiente comando.

```
rpm -ivh rpm -ivh mysql-server-5.0.45-7.el5.i386.rpm
```

si el archivo es instalado correctamente debe aparecer el siguiente mensaje

```
Preparing... #####
[100%]
 1:mysql-server #####
[100%]
```

Después se debe subir el servicio y revisar que la base de datos resida en este filesystem.

```
service mysqld start
```

luego comprobar con la orden

```
ps aux | grep mysqld
```

```
10:22  0:00 /bin/sh /usr/bin/mysqld_safe --datadir=/var/lib/mysql
--socket=/var/lib/mysql/mysql.sock --log-error=/var/log/mysqld.log --pid-
file=/var/run/mysqld/mysqld.pid
mysql    31013  0.1  0.0 157664 31024 pts/2    S1   10:22   0:00
/usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=mysql
--pid-file=/var/run/mysqld/mysqld.pid --skip-external-locking -
socket=/var/lib/mysql/mysql.sock
```

Como se puede comprobar la directiva datadir debe estar apuntando al filesystem montado /var/lib/mysql.

Ahora se debe programar para que el servicio de mysql suba automáticamente con el sistemas en los niveles de ejecución 3 al 5.

```
chkconfig --level 35 mysqld on
```

Después de este procedimiento se debe instalar los demás rpms, correspondiente a un cliente mysql, los conectores odbc, un suite de pruebas y una benchmark.

```
-r--r--r-- 1 mysql mysql 4685645 Nov 24 14:17 mysql-5.0.45-
7.el5.ia64.rpm
-r--r--r-- 1 mysql mysql 525464 Nov 24 14:17 mysql-bench-5.0.45-
7.el5.ia64.rpm
-r--r--r-- 1 mysql mysql 199141 Nov 24 14:17 mysql-connector-odbc-
3.51.12-2.2.ia64.rpm
-r--r--r-- 1 mysql mysql 3625160 Nov 24 14:16 mysql-test-5.0.45-
7.el5.ia64.rpm
```

La Base de datos master queda configurado en 10.20.1.4 y la slave en la 10.20.1.5

5.4.2 Creación de usuarios para replicación

Se debe crear un usuario para replicación en la base de datos master y en la base de datos slave, con las siguientes órdenes dentro del ambiente de mysql.

```
mysql
CREATE USER replica identified by 'password',
```

el password está en control de ingeniero Alfredo.

Después a este usuario se le debe dar privilegios para que ejecute la replicación hacia la base de datos master y desde la base de datos slave.

```
GRANT REPLICATION SLAVE,REPLICATION CLIENT ON *.* TO replica@10.20.1.5
IDENTIFIED BY 'Re_Sof75&';
```

Después de ejecutada la anterior orden se debe aplicar nuevamente los privilegios con

```
FLUSH PRIVILEGES;
```

5.4.3 Creación de archivo de configuración de replicación.

Se crea un archivo en /etc/my.cnf en caso que no exista. Por defecto el instalador del rpm los crear con valores por defecto.

Antes de iniciar el proceso de configuración se debe detener los servicios de mysql con la orden:

```
service mysqld stop
```

En el servidor Master, la configuración del archivo my.cnf debe quedar de la siguiente manera:

```
[mysqld]
server-id=1
relay-log =/var/lib/mysql/log/mysql-relay-bin
relay-log-index =/var/lib/mysql/log/mysql-relay-bin.index
log-error=/var/lib/mysql/log/mysql.err
datadir=/var/lib/mysql
log-bin=/var/lib/mysql/mysql-bin
master-info-file = /var/lib/mysql/mysql-master.info
relay-log-info-file = /var/lib/mysql-relay-log.info
```

El archivo -/etc/my.cnf en el servidor slave debe quedar de la siguiente manera

```
[mysqld]
server-id=2
relay-log =/var/lib/mysql/log/mysql-relay-bin
relay-log-index =/var/lib/mysql/log/mysql-relay-bin.index
log-error=/var/lib/mysql/log/mysql.err
datadir=/var/lib/mysql
master-info-file = /var/lib/mysql/mysql-master.info
relay-log-info-file = /var/lib/mysql/relay-log.info
```

Como se puede notar es muy importante que el id del servidor master sea diferente del servidor slave.

Una vez realizado estos cambios se debe iniciar el servidor master y se debe revisar la configuración que ha sido tomada por este:

```
mysql
mysql> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000003 |      393 |              |                   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

En el cual se indica el nombre del log binario y la posición en la cual se encuentra. Si ya se tiene una base de datos, se debe realizar una copia de seguridad con mysqldump para luego aplicar estos cambios en la base de datos slave.

1. Elegir la base de datos
2. Bloquear las tablas.
3. Realizar copia de los datos
4. copiar backup a maquina slave
5. Aplicar backup a servidor slave,
6. Desbloquear las tablas en servidor Master.

Como se muestra en las siguiente ordenes.

En Master, tomamos como ejemplo la base de datos test.

```
USE test;
FLUSH TABLES WITH READ LOCK;
quit
en linea de comandos
mysqldum -u root test > /tmp/backup.sql
```

Copiar el archivo a el servidor slave, por ejemplo en la ruta /tmp/backup.sql y luego aplicar los cambios

```
mysql -u root < /tmp/backup.sql
```

Una vez aplicados los cambios, en la base Master se deben desbloquear las tablas y comprobar el estado de la replicación.

```
mysql> unlock tables
mysql> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
```

```

+-----+-----+-----+-----+
| mysql-bin.000003 |      393 |           |           |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

En la base de datos slave, después de iniciar los servicios de mysql, se debe activar el modo de replicación de la siguiente manera:

```

CHANGE MASTER TO MASTER_HOST='lmsdb1', MASTER_USER='replica',
MASTER_PASSWORD='Re_sof75&', MASTER_LOG_FILE='mysql-bin.000003',
MASTER_LOG_POS=98;

```

Donde el password el asignado al usuario de replica, el Master_log_File, y el Master_log_pos son los indicados por la salida del comando show master status en la base Master.

Se comprueba luego el estado de la base de datos slave con la orden

```

mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: lmsdb1
      Master_User: replica
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000003
  Read_Master_Log_Pos: 393
      Relay_Log_File: mysql-relay-bin.000003
      Relay_Log_Pos: 530
  Relay_Master_Log_File: mysql-bin.000003
  Slave_IO_Running: no
  Slave_SQL_Running: no
  Replicate_Do_DB:
  Replicate_Ignore_DB:
  Replicate_Do_Table:
  Replicate_Ignore_Table:
  Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
      Last_Errno: 0
      Last_Error:
      Skip_Counter: 0
  Exec_Master_Log_Pos: 393
      Relay_Log_Space: 530
      Until_Condition: None

```

```

      Until_Log_File:
      Until_Log_Pos: 0
      Master_SSL_Allowed: No
      Master_SSL_CA_File:
      Master_SSL_CA_Path:
      Master_SSL_Cert:
      Master_SSL_Cipher:
      Master_SSL_Key:
      Seconds_Behind_Master: 0
1 row in set (0.00 sec)

```

Como se puede observar los threads que corren los servicios de replicación están deshabilitados

```

Slave_IO_Running: no
Slave_SQL_Running: no

```

Por lo que tenemos que iniciar el slave.

Con la orden

```

slave start;

```

y al correr de nuevo el comando `show slave status`, se puede comprobar que los threads se ha iniciado correctamente.

```

mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: lmsdb1
      Master_User: replica
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000003
      Read_Master_Log_Pos: 393
      Relay_Log_File: mysql-relay-bin.000003
      Relay_Log_Pos: 530
      Relay_Master_Log_File: mysql-bin.000003
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes

```

También se pueden observar en el archivo de log que los servicios se hayan podido iniciar correctamente, en el servidor slave.

```
tail -f /var/lib/mysql/log/mysql.err
```

y se podrá observar una salida como esta (esta ruta como se observa se ha configurado dentro del archivo /etc/my.cnf).

```
091124 16:11:09 [Note] Slave SQL thread initialized, starting replication
in log 'mysql-bin.000003' at position 98, relay log
'/var/lib/mysql/log/mysql-relay-bin.000002' position: 235
091124 16:11:09 [Note] Slave I/O thread: connected to master
'replica@lmsdb1:3306', replication started in log 'mysql-bin.000003' at
position 98
```

5.5 INSTALACION DE BALANCEADOR DE ENLACES CON LINUX IPFILTER

Para la instalación se requiere un sistema Linux cualquier configurado con 2 tarjetas de red, que sean conectadas a cada uno de los enlaces existente.

Para verificar el estado de los enlaces se puede utilizar la herramienta ethtool de la siguiente manera:

```
ethtool eth0 o eth1
```

que tendrá una salida similar a la siguiente

```
Settings for eth0:
  Supported ports: [ TP MII ]
  Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
  Supports auto-negotiation: Yes
  Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
  Advertised auto-negotiation: Yes
  Speed: 10Mb/s
  Duplex: Half
  Port: MII
  PHYAD: 0
  Transceiver: internal
  Auto-negotiation: on
  Supports Wake-on: pumbg
  Wake-on: g
  Current message level: 0x00000033 (51)
  Link detected: no
```

En la sección Link detected, indica si se estable o no conexión.

Una vez se tengan las dos tarjetas de red configuradas se debe revisar que los siguiente módulos de netfilter existan o hayan sido configurados

- CONMARK
- nth
- condition
- Iproute2

La revisión de puede realizar de la siguiente manera

```
ls /lib/modules/$(uname -r)/kernel/net/netfilter | grep -i "modulo"
```

en donde módulo es el nombre específico de un módulo sin comillas.

Como se puede ver estos módulos pertenecen al kernel y por tanto con la compilación del mismo se debieron incluir.

En caso de no tener los módulos cargados se puede utilizar la orden

```
modprobe "nombre del modulo "
```

y luego revisarlo con

```
modinfo "nombre del modulo"
```

ahora marcamos los paquetes con CONMARK de la siguiente manera

```
iptables -A PREROUTING -t mangle -j CONNMARK --restore-mark
iptables -A PREROUTING -t mangle -m mark --mark 0x0 -m nth --counter 1 \
--every 4 --packet 1 -j MARK --set-mark 1
iptables -A PREROUTING -t mangle -m mark --mark 0x0 -m nth --counter 1 \
--every 4 --packet 2 -j MARK --set-mark 2
iptables -A PREROUTING -t mangle -m mark --mark 0x0 -m nth --counter 1 \
--every 4 --packet 3 -j MARK --set-mark 1
iptables -A PREROUTING -t mangle -m mark --mark 0x0 -m nth --counter 1 \
```

```
--every 4 --packet 4 -j MARK --set-mark 1
iptables -A POSTROUTING -j CONNMARK --save-mark
```

Ahora ponemos marcas a los paquetes de acuerdo a la interface por la cual pase ya sea la 1 o la 2.

```
iptables -N MARKING
iptables -A PREROUTING -t mangle -j CONNMARK --restore-mark
iptables -A PREROUTING -t mangle -m mark --mark 0x0 -j MARKING

iptables -A MARKING -t mangle -m condition --condition link1_up \
-m nth --counter 1 --every 4 --packet 1 -j MARK --set-mark 1
iptables -A MARKING -t mangle -m condition ! --condition link1_up \
-m nth --counter 1 --every 4 --packet 1 -j MARK --set-mark 1

iptables -A MARKING -t mangle -m condition --condition link2_up
-m nth --counter 1 --every 4 --packet 2 -j MARK --set-mark 2
iptables -A MARKING -t mangle -m condition ! --condition link2_up
-m nth --counter 1 --every 4 --packet 2 -j MARK --set-mark 1

iptables -A MARKING -t mangle -m condition --condition link1_up \
-m nth --counter 1 --every 4 --packet 3 -j MARK --set-mark 1
iptables -A MARKING -t mangle -m condition ! --condition link1_up \
-m nth --counter 1 --every 4 --packet 3 -j MARK --set-mark 2

iptables -A MARKING -t mangle -m condition --condition link1_up \
-m nth --counter 1 --every 4 --packet 4 -j MARK --set-mark 1
iptables -A MARKING -t mangle -m condition ! --condition link1_up \
-m nth --counter 1 --every 4 --packet 4 -j MARK --set-mark 2

iptables -A POSTROUTING -j CONNMARK --save-mark
```

ahora realizamos el balanceo para que los paquetes marcados con uno tomen el enlace 1 y los marcados con 2 tomen en enlace 2

```
ip route add default gw GW_LINK1 table LINK1
ip route add default gw GW_LINK2 table LINK2
ip rule add fwmark 1 lookup table LINK1
ip rule add fwmark 2 lookup table LINK2
```


5.6 INSTALACION CLUSTER DE SERVIDORES BALANCEADOS CON LVS.

5.6.1 Instalación de paquetes

Para realizar el balanceo de los servidores de la granja en cada uno de los equipos encargados de balanceo se debe instalar los siguiente paquetes:

```
HeartBeat
HeartBeat-pils
HeartBeat-stonith
HeartBeat-ldirectord
```

Para esto se utilizará la herramienta yum encargada de instalar y resolver las dependencias necesarias de cada uno de los paquetes de la siguiente manera:

```
[root@localhost ~]# yum install HeartBeat,
```

ante lo cual se tendrá una respuesta semejante a la siguiente:

```
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * addons: mirror.anl.gov
 * base: mirror.steadfast.net
 * extras: mirror.steadfast.net
 * updates: mirrors.combinetworks.com
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package HeartBeat.i386 0:2.1.3-3.el5.centos set to be updated
--> Processing Dependency: HeartBeat-stonith = 2.1.3-3.el5.centos for
package: HeartBeat
--> Processing Dependency: HeartBeat-pils = 2.1.3-3.el5.centos for
package: HeartBeat
--> Processing Dependency: libpils.so.1 for package: HeartBeat
--> Processing Dependency: libstonith.so.1 for package: HeartBeat
--> Running transaction check
---> Package HeartBeat-pils.i386 0:2.1.3-3.el5.centos set to be updated
---> Package HeartBeat-stonith.i386 0:2.1.3-3.el5.centos set to be
```

```
updated
--> Finished Dependency Resolution
```

Dependencies Resolved

Package Size	Arch	Version	Repository
-----------------	------	---------	------------

```
Installing:
HeartBeat          i386          2.1.3-3.el5.centos      extras
1.7 M
Installing for dependencies:
HeartBeat-pils     i386          2.1.3-3.el5.centos      extras
213 k
HeartBeat-stonith  i386          2.1.3-3.el5.centos      extras
311 k
```

Transaction Summary

```
Install      3 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 2.2 M
Is this ok [y/N]: y
```

A esta pregunta se debe responder Y, para indicar que se está de acuerdo con la instalación de los paquetes mencionados y sus respectivas dependencias,.

El sistemas entonces realizará la descarga y la instalación como se muestra a continuación:

```
Downloading Packages:
(1/3): HeartBeat-pils-2.1.3-3.el5.centos.i386.rpm | 213 kB
00:01
(2/3): HeartBeat-stonith-2.1.3-3.el5.centos.i386.rpm | 311 kB
00:01
(3/3): HeartBeat-2.1.3-3.el5.centos.i386.rpm | 1.7 MB
00:06
-----
Total                               220 kB/s | 2.2 MB
00:10
```

```

Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing      : HeartBeat-pils
1/3
  Installing      : HeartBeat-stonith
2/3
useradd: user haCluster exists
error: %pre(HeartBeat-2.1.3-3.el5.centos.i386) scriptlet failed, exit
status 9
error:  install: %pre scriptlet failed (2), skipping HeartBeat-2.1.3-
3.el5.centos

Installed:
  HeartBeat.i386 0:2.1.3-3.el5.centos

Dependency Installed:
  HeartBeat-pils.i386 0:2.1.3-3.el5.centos
  HeartBeat-stonith.i386 0:2.1.3-3.el5.centos

Complete!

```

Si la instalación se realizó satisfactoriamente, se verá el mensaje de complete.

Ahora se procede con la instalación del segundo paquete requerido:

```
[root@localhost ~]# yum install heartbeat-ldirectord
```

Este producto tiene varias dependencias de librerías de PERL, por tanto el listado será grande y debe ser semejante al siguiente:

```

Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
* addons: mirror.linux.duke.edu
* base: mirror.steadfast.net
* extras: mirrors.netdna.com
* updates: mirrors.liquidweb.com
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package HeartBeat-ldirectord.i386 0:2.1.3-3.el5.centos set to be
updated

```

```

--> Processing Dependency: perl-HTML-Parser for package: HeartBeat-ldirectord
--> Processing Dependency: perl-Authen-Radius for package: HeartBeat-ldirectord
--> Processing Dependency: perl-MailTools for package: HeartBeat-ldirectord
--> Processing Dependency: perl-Net-IMAP-Simple for package: HeartBeat-ldirectord
--> Processing Dependency: perl-POP3Client for package: HeartBeat-ldirectord
--> Processing Dependency: perl-Net-DNS for package: HeartBeat-ldirectord
--> Processing Dependency: ipvsadm for package: HeartBeat-ldirectord
--> Processing Dependency: perl-libwww-perl for package: HeartBeat-ldirectord
--> Processing Dependency: perl-Crypt-SSLeay for package: HeartBeat-ldirectord
--> Processing Dependency: perl(LWP::UserAgent) for package: HeartBeat-ldirectord
--> Processing Dependency: perl-LDAP for package: HeartBeat-ldirectord
--> Processing Dependency: perl(Mail::Send) for package: HeartBeat-ldirectord
--> Processing Dependency: perl-Net-IMAP-Simple-SSL for package: HeartBeat-ldirectord
--> Processing Dependency: perl(LWP::Debug) for package: HeartBeat-ldirectord
--> Running transaction check
---> Package ipvsadm.i386 0:1.24-10 set to be updated
---> Package perl-Authen-Radius.noarch 0:0.13-1.e15.centos set to be updated
--> Processing Dependency: perl(Data::HexDump) for package: perl-Authen-Radius
---> Package perl-Crypt-SSLeay.i386 0:0.51-11.e15 set to be updated
---> Package perl-HTML-Parser.i386 0:3.55-1.fc6 set to be updated
--> Processing Dependency: perl-HTML-Tagset >= 3.03 for package: perl-HTML-Parser
--> Processing Dependency: perl(HTML::Tagset) for package: perl-HTML-Parser
---> Package perl-LDAP.noarch 1:0.33-3.fc6 set to be updated
--> Processing Dependency: perl(Convert::ASN1) for package: perl-LDAP
--> Processing Dependency: perl(XML::SAX::Base) for package: perl-LDAP
--> Processing Dependency: perl(IO::Socket::SSL) for package: perl-LDAP
---> Package perl-Mail-POP3Client.noarch 0:2.17-1.e15.centos set to be updated
---> Package perl-MailTools.noarch 0:1.77-1.e15.centos set to be updated
--> Processing Dependency: perl(Date::Parse) for package: perl-MailTools
--> Processing Dependency: perl(Date::Format) for package: perl-MailTools
---> Package perl-Net-DNS.i386 0:0.59-3.e15 set to be updated
--> Processing Dependency: perl(Digest::HMAC_MD5) for package: perl-Net-DNS
--> Processing Dependency: perl(Net::IP) for package: perl-Net-DNS
---> Package perl-Net-IMAP-Simple.noarch 0:1.17-1.e15.centos set to be updated
---> Package perl-Net-IMAP-Simple-SSL.noarch 0:1.3-1.e15.centos set to be

```

```

updated
---> Package perl-libwww-perl.noarch 0:5.805-1.1.1 set to be updated
--> Processing Dependency: perl(URI::URL) for package: perl-libwww-perl
--> Processing Dependency: perl(URI) for package: perl-libwww-perl
--> Processing Dependency: perl(Compress::Zlib) for package: perl-libwww-
perl
--> Processing Dependency: perl(URI::Heuristic) for package: perl-libwww-
perl
--> Processing Dependency: perl(URI::Escape) for package: perl-libwww-
perl
--> Running transaction check
---> Package perl-Compress-Zlib.i386 0:1.42-1.fc6 set to be updated
---> Package perl-Convert-ASN1.noarch 0:0.20-1.1 set to be updated
---> Package perl-Data-HexDump.noarch 0:0.02-1.el5.centos set to be
updated
---> Package perl-Digest-HMAC.noarch 0:1.01-15 set to be updated
--> Processing Dependency: perl(Digest::SHA1) for package: perl-Digest-
HMAC
---> Package perl-HTML-Tagset.noarch 0:3.10-2.1.1 set to be updated
---> Package perl-IO-Socket-SSL.noarch 0:1.01-1.fc6 set to be updated
--> Processing Dependency: perl(Net::SSLeay) >= 1.21 for package: perl-
IO-Socket-SSL
---> Package perl-Net-IP.noarch 0:1.25-2.fc6 set to be updated
---> Package perl-TimeDate.noarch 1:1.16-5.el5 set to be updated
---> Package perl-URI.noarch 0:1.35-3 set to be updated
---> Package perl-XML-SAX.noarch 0:0.14-5 set to be updated
--> Processing Dependency: perl(XML::NamespaceSupport) for package: perl-
XML-SAX
--> Running transaction check
---> Package perl-Digest-SHA1.i386 0:2.11-1.2.1 set to be updated
---> Package perl-Net-SSLeay.i386 0:1.30-4.fc6 set to be updated
---> Package perl-XML-NamespaceSupport.noarch 0:1.09-1.2.1 set to be
updated
--> Finished Dependency Resolution

```

Dependencies Resolved

Package Repository	Size	Arch	Version
Installing:			
HeartBeat-ldirectord extras	199 k	i386	2.1.3-3.el5.centos
Installing for dependencies:			
ipvsadm base	31 k	i386	1.24-10
perl-Authen-Radius extras	16 k	noarch	0.13-1.el5.centos
perl-Compress-Zlib base	52 k	i386	1.42-1.fc6

perl-Convert-ASN1		noarch	0.20-1.1
base	42 k		
perl-Crypt-SSLeay		i386	0.51-11.e15
base	45 k		
perl-Data-HexDump		noarch	0.02-1.e15.centos
extras	8.4 k		
perl-Digest-HMAC		noarch	1.01-15
base	12 k		
perl-Digest-SHA1		i386	2.11-1.2.1
base	48 k		
perl-HTML-Parser		i386	3.55-1.fc6
base	92 k		
perl-HTML-Tagset		noarch	3.10-2.1.1
base	15 k		
perl-IO-Socket-SSL		noarch	1.01-1.fc6
base	49 k		
perl-LDAP		noarch	1:0.33-3.fc6
base	316 k		
perl-Mail-POP3Client		noarch	2.17-1.e15.centos
extras	21 k		
perl-MailTools		noarch	1.77-1.e15.centos
extras	91 k		
perl-Net-DNS		i386	0.59-3.e15
base	215 k		
perl-Net-IMAP-Simple		noarch	1.17-1.e15.centos
extras	19 k		
perl-Net-IMAP-Simple-SSL		noarch	1.3-1.e15.centos
extras	6.5 k		
perl-Net-IP		noarch	1.25-2.fc6
base	31 k		
perl-Net-SSLeay		i386	1.30-4.fc6
base	195 k		
perl-TimeDate		noarch	1:1.16-5.e15
base	32 k		
perl-URI		noarch	1.35-3
base	116 k		
perl-XML-Namespacesupport		noarch	1.09-1.2.1
base	15 k		
perl-XML-SAX		noarch	0.14-5
base	75 k		
perl-libwww-perl		noarch	5.805-1.1.1
base	376 k		

Transaction Summary

```

=====
Install      25 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

```

Total download size: 2.1 M

Downloading Packages:

(1/25): perl-Net-IMAP-Simple-SSL-1.3-1.e15.centos.noarch.rpm

```

| 6.5 kB    00:00
(2/25): perl-Data-HexDump-0.02-1.el5.centos.noarch.rpm
| 8.4 kB    00:00
(3/25): perl-Digest-HMAC-1.01-15.noarch.rpm
| 12 kB     00:00
(4/25): perl-HTML-Tagset-3.10-2.1.1.noarch.rpm
| 15 kB     00:00
(5/25): perl-XML-Namespacesupport-1.09-1.2.1.noarch.rpm
| 15 kB     00:00
(6/25): perl-Authen-Radius-0.13-1.el5.centos.noarch.rpm
| 16 kB     00:00
(7/25): perl-Net-IMAP-Simple-1.17-1.el5.centos.noarch.rpm
| 19 kB     00:00
(8/25): perl-Mail-POP3Client-2.17-1.el5.centos.noarch.rpm
| 21 kB     00:00
(9/25): perl-Net-IP-1.25-2.fc6.noarch.rpm
| 31 kB     00:00
(10/25): ipvsadm-1.24-10.i386.rpm
| 31 kB     00:00
(11/25): perl-TimeDate-1.16-5.el5.noarch.rpm
| 32 kB     00:00
(12/25): perl-Convert-ASN1-0.20-1.1.noarch.rpm
| 42 kB     00:00
(13/25): perl-Crypt-SSLeay-0.51-11.el5.i386.rpm
| 45 kB     00:00
(14/25): perl-Digest-SHA1-2.11-1.2.1.i386.rpm
| 48 kB     00:00
(15/25): perl-IO-Socket-SSL-1.01-1.fc6.noarch.rpm
| 49 kB     00:00
(16/25): perl-Compress-Zlib-1.42-1.fc6.i386.rpm
| 52 kB     00:00
(17/25): perl-XML-SAX-0.14-5.noarch.rpm
| 75 kB     00:00
(18/25): perl-MailTools-1.77-1.el5.centos.noarch.rpm
| 91 kB     00:00
(19/25): perl-HTML-Parser-3.55-1.fc6.i386.rpm
| 92 kB     00:00
(20/25): perl-URI-1.35-3.noarch.rpm
| 116 kB    00:00
(21/25): perl-Net-SSLeay-1.30-4.fc6.i386.rpm
| 195 kB    00:00
(22/25): HeartBeat-ldirectord-2.1.3-3.el5.centos.i386.rpm
| 199 kB    00:00
(23/25): perl-Net-DNS-0.59-3.el5.i386.rpm
| 215 kB    00:00
(24/25): perl-LDAP-0.33-3.fc6.noarch.rpm
| 316 kB    00:01
(25/25): perl-libwww-perl-5.805-1.1.1.noarch.rpm
| 376 kB    00:01
-----
-----
Total                                     162 kB/s
| 2.1 MB    00:13

```

```
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing      : perl-Data-HexDump
1/25
  Installing      : perl-Authen-Radius
2/25
  Installing      : perl-Net-IP
3/25
  Installing      : perl-HTML-Tagset
4/25
  Installing      : perl-URI
5/25
  Installing      : perl-Convert-ASN1
6/25
  Installing      : perl-Net-IMAP-Simple
7/25
  Installing      : perl-XML-Namespacesupport
8/25
  Installing      : perl-XML-SAX
9/25
  Installing      : perl-TimeDate
10/25
  Installing      : perl-MailTools
11/25
  Installing      : perl-Mail-POP3Client
12/25
  Installing      : perl-HTML-Parser
13/25
  Installing      : perl-Net-SSLeay
14/25
  Installing      : perl-Digest-SHA1
15/25
  Installing      : perl-Crypt-SSLeay
16/25
  Installing      : perl-Compress-Zlib
17/25
  Installing      : ipvsadm
18/25
  Installing      : perl-libwww-perl
19/25
  Installing      : perl-IO-Socket-SSL
20/25
  Installing      : perl-Net-IMAP-Simple-SSL
21/25
  Installing      : perl-LDAP
22/25
  Installing      : perl-Digest-HMAC
23/25
  Installing      : perl-Net-DNS
24/25
```

```

Installing      : HeartBeat-ldirectord
25/25

Installed:
  HeartBeat-ldirectord.i386 0:2.1.3-3.el5.centos

Dependency Installed:
  ipvsadm.i386 0:1.24-10
  perl-Authen-Radius.noarch 0:0.13-1.el5.centos
  perl-Compress-Zlib.i386 0:1.42-1.fc6
  perl-Convert-ASN1.noarch 0:0.20-1.1
  perl-Crypt-SSLeay.i386 0:0.51-11.el5
  perl-Data-HexDump.noarch 0:0.02-1.el5.centos
  perl-Digest-HMAC.noarch 0:1.01-15
  perl-Digest-SHA1.i386 0:2.11-1.2.1
  perl-HTML-Parser.i386 0:3.55-1.fc6
  perl-HTML-Tagset.noarch 0:3.10-2.1.1
  perl-IO-Socket-SSL.noarch 0:1.01-1.fc6
  perl-LDAP.noarch 1:0.33-3.fc6
  perl-Mail-POP3Client.noarch 0:2.17-1.el5.centos
  perl-MailTools.noarch 0:1.77-1.el5.centos
  perl-Net-DNS.i386 0:0.59-3.el5
  perl-Net-IMAP-Simple.noarch 0:1.17-1.el5.centos
  perl-Net-IMAP-Simple-SSL.noarch 0:1.3-1.el5.centos
  perl-Net-IP.noarch 0:1.25-2.fc6
  perl-Net-SSLeay.i386 0:1.30-4.fc6
  perl-TimeDate.noarch 1:1.16-5.el5
  perl-URI.noarch 0:1.35-3
  perl-XML-NamespaceSupport.noarch 0:1.09-1.2.1
  perl-XML-SAX.noarch 0:0.14-5
  perl-libwww-perl.noarch 0:5.805-1.1.1

Complete!

```

Con los productos instalados correctamente en cada uno de los balanceadores, se procederá a la configuración de los mismos:

Los paquetes instalados crean el directorio `/etc/ha.d/` y dentro del él debemos crear los siguientes archivos.

```

ha.cf
haresources
authkeys
ldirectord.cf

```

Y dentro del directorio `/etc/ha.d/resources.d/` encontramos el archivo

```
ldirectord
```

Dentro de los tres primeros archivos se configura la alta disponibilidad de los dos nodos balanceadores y en último se configura el balanceo hacía la granja de servidores.

5.6.2 Activación de las reglas de routing

En primer lugar se debe revisar y ajustar algunos parámetros del kernel los cuales son:

```
net.ipv4.conf.all.arp_ignore = 1
net.ipv4.conf.eth0.arp_ignore = 1
net.ipv4.conf.all.arp_announce = 2
net.ipv4.ip_forward = 1
```

Estos parámetros se debido a que quien el balanceador será el punto final para el cliente, pero a la vez tendrá que hacer reenvío de peticiones hacia cada miembro de la granja.

Para hacer que estos cambios queden permanente dentro del kernel de linux, se deben colocar en el archivo `/etc/sysctl.conf`

Para que estos cambios se tomen sin necesidad de reiniciar el servidor se debe ejecutar el comando

```
sysctl -p
```

Una vez aplicados estos parámetros, se debe configurar el soporte para realizar reenvío de paquetes por medio de una regla de iptables, de la siguiente manera:

dentro del directorio /etc/sysconfig/ crear un archivo llamado iptables, con el siguiente contenido

```
# Generated by iptables-save v1.3.5 on Sat Jan 16 07:07:52 2010
*nat
:PREROUTING ACCEPT [1:230]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [1:234]
-A POSTROUTING -s 192.168.12.0/255.255.255.0 -j MASQUERADE
-A POSTROUTING -j LOG
COMMIT
# Completed on Sat Jan 16 07:07:52 2010
```

Esto permite que se puede realizar una regla de NAT con destino a la red 192.168.12.0 (red de la granja de servidores).

Una vez realizado el cambio es necesario reiniciar la el servicio de iptables, de la siguiente manera:

```
service iptables restart
```

y se debería tener una salida como la siguiente

```
[root@lvs1 ha.d]# service iptables restart
Flushing firewall rules: [ OK ]
Setting chains to policy ACCEPT: nat [ OK ]
Unloading iptables modules: Removing netfilter NETLINK layer.
[ OK ]
Applying iptables firewall rules: ip_tables: (C) 2000-2006 Netfilter Core Team
Netfilter messages via NETLINK v0.30.
ip_conntrack version 2.4 (2047 buckets, 16376 max) - 228 bytes per conntrack
[ OK ]
Loading additional iptables modules: ip_conntrack_netbios_n[ OK ]
[root@lvs1 ha.d]# IN= OUT=eth0 SRC=192.168.11.3 DST=192.168.11.255 LEN=234 TOS=0
x00 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP SPT=37262 DPT=694 LEN=214
```

Gráfico 13: Activación de los módulos de netfilter para realizar reenvío de paquetes

Se deben configurar además 2 interfaces de red en cada uno de los balanceadores, en donde la primera interfase de red debe pertenecer a la red

192.168.11.0 (red de lvs) y la segunda a la red 192.168.12.0 (red de granja de servidores).

Esto se puede comprobar con el comando `ip addr list` y el resultado debe ser semejante al siguiente:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen
1000
    link/ether 08:00:27:74:78:98 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.3/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::a00:27ff:fe74:7898/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen
1000
    link/ether 08:00:27:07:7d:3d brd ff:ff:ff:ff:ff:ff
    inet 192.168.12.243/24 brd 192.168.12.255 scope global eth1
    inet6 fe80::a00:27ff:fe07:7d3d/64 scope link
        valid_lft forever preferred_lft forever
4: sit0: <NOARP> mtu 1480 qdisc noop
    link/sit 0.0.0.0 brd 0.0.0.0
```

Este direccionamiento se realiza con bases en el gráfico 3. Esquema de direccionamiento de la solución.

Para realizar la configuración de las interfaces de red, se debe editar el archivo correspondiente a la interface en `/etc/sysconfig/network-scripts/ifcfg-x`, donde `x` es la interface ya sea `eth0`, `eth1`, etc.

Dentro de este se debe configurar similar a la siguiente ejemplo:

```
DEVICE=eth0
IPADDR=192.168.11.3
NETMASK=255.255.255.0
```

```
NETWORK=192.168.11.0
BROADCAST=192.168.11.255
ONBOOT=yes
NAME=loopback
```

esto se debe hacer para cada interfase a configurar. Después de esto se debe reiniciar las interfaces de red con el siguiente comando

```
service network restart
```

Ahora asignar en el archivo `/etc/sysconfig/network` los parametros de nombre de servidor.

```
NETWORKING=yes
HOSTNAME=lvs1
```

5.6.3 Configuración del servicio de balanceo y alta disponibilidad

Para el servicio de balanceo es necesario agregar las siguiente líneas al archivo `ldirectord.cf` ubicado en `/etc/ha.d/`

```
# Global Directives
checktimeout=3
checkinterval=1
autoreload=yes
logfile="/var/log/ldirectord.log"
quiescent=yes

# http virtual service
virtual=192.168.11.5:80
    real=192.168.12.3:80 masq
    real=192.168.12.4:80 masq
    fallback=127.0.0.1:80
    service=http
    request="index.html"
    receive="Test Page"
    scheduler=rr
```

```
protocol=tcp
checktype=negotiate
checkport=80
request="index.html"
receive="Test Page for balancing"
```

En estos parámetros se indica, cuál será la dirección virtual que será compartida por todos los servicios, además el servicio que será respaldado y los miembros con las direcciones reales en donde será balanceadas las peticiones. Para este caso se indica que el balanceo será realizado al servicio de http por el puerto 80.

También se debe configurar las opciones de alta disponibilidad, para que la dirección virtual activada por los nodos, evitando único punto de falla

En el archivo `/etc/ha.d/ha.cf`

```
logfacility    local0
keepalive     1
deadtime      10
warntime      5
initdead      120
udpport       694
mcast eth0
auto_failback on
node          lvs1
node          lvs2
ping          192.168.0.254
```

En el archivo `/etc/ha.d/authkeys`

```
auth 2
2 sha1 ThisIsMyPassword
```

y en el archivo `/etc/ha.d/haresources`

```
lvs1 \
LVSSyncDaemonSwap::master \
IPaddr::192.168.11.5/24/eth0 \
```

```
IPaddr::192.168.12.240/24/eth1 \  
ldirectord
```

Estos archivos de configuración debe ser copiados en los equipos configurados dentro de los balanceadores.

Una vez realizados los cambios, se debe reiniciar el servicio de heartbeat de la siguiente manera:

```
service heartbeat restart
```

El servicio de heartbeat debe ser reiniciado de manera escalonada, el nodo principal primero y luego el nodo secundario, si se realiza de manera simultánea puede provocar la caída del servicio de alta disponibilidad.

Al listar nuevamente las interfaces se debe tener una salida como la siguiente:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen  
1000  
    link/ether 08:00:27:74:78:98 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.11.3/24 brd 192.168.11.255 scope global eth0  
    inet 192.168.11.5/24 brd 192.168.11.255 scope global secondary eth0:0  
    inet6 fe80::a00:27ff:fe74:7898/64 scope link  
        valid_lft forever preferred_lft forever  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen  
1000  
    link/ether 08:00:27:07:7d:3d brd ff:ff:ff:ff:ff:ff  
    inet 192.168.12.243/24 brd 192.168.12.255 scope global eth1  
    inet 192.168.12.240/24 brd 192.168.12.255 scope global secondary  
eth1:0  
    inet6 fe80::a00:27ff:fe07:7d3d/64 scope link  
        valid_lft forever preferred_lft forever  
4: sit0: <NOARP> mtu 1480 qdisc noop  
    link/sit 0.0.0.0 brd 0.0.0.0
```

Donde se pueden observar las dos nuevas direcciones virtuales 192.168.11.5 correspondiente a la interfase de servicio externa y la dirección 192.168.12.240 correspondiente a la interfase de servicio interna.

Para comprobar que el balanceo ha quedado correctamente configurado, se puede ejecutar el siguiente comando:

```
ipvsadm -Ln
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port      Forward Weight ActiveConn InActConn
TCP  192.168.11.5:80 rr
  -> 192.168.12.3:80          Masq    1      0      0
  -> 192.168.12.4:80          Masq    0      0      0
```

La salida muestra el nodo con la dirección virtual y los servidores reales a los cuales apunta, además indica el peso para la realización del balanceo. En caso de estar caído el nodo coloca un 0 y en el caso de estar activo coloca un 1 o el valor que se especifique al nodo en el archivo `ldirectord.cf`.

Configuración de granja de servidores apache, LMS y perl.

En estos equipos se debe realizar la configuración de la plataforma Maat-g tal como se indica en la sección [5.2 INSTALACIÓN Y CONFIGURACIÓN PLATAFORMA MAAT-G SOBRE SISTEMAS OPERATIVO CENTOS OS](#) de este anexo.

Además se debe indicar que el gateway por defecto para estos servidores es la dirección de servicio interna 192.168.12.240, para esto se debe editar el archivo `/etc/sysconfig/network` y adicionar la siguiente línea:

```
GATEWAY=192.168.12.240
```

5.7 REALIZACIÓN DE PRUEBAS DE DISPONIBILIDAD.

Descargar el software desde el sitio <http://jakarta.apache.org/jmeter/> en la versión más reciente en su revisión estable.

Para este manual, hemos utilizado la versión 2.3.4 del producto

Requisitos: Tener instalado una versión de la máquina virtual de java 1.4 o superior.

Para comprobar la versión de java, basta con ejecutar el comando.

Java -version

el cual debe tener una salida semejante a la siguiente:

```
eshernan@eshernan-laptop:~/instaladores$ java -version
java version "1.6.0_16"
Java(TM) SE Runtime Environment (build 1.6.0_16-b01)
Java HotSpot(TM) Server VM (build 14.2-b01, mixed mode)
eshernan@eshernan-laptop:~/instaladores$
```

con este requisito, se descomprime el instalador y dentro de la estructura de directorios encontramos la carpeta bin.

```
eshernan@eshernan-laptop:~/programas/jakarta-jmeter-2.3.4$ ls -l
total 120
drwxr-xr-x 3 eshernan eshernan 4096 2009-12-29 19:36 bin
drwxr-xr-x 5 eshernan eshernan 4096 2009-12-29 19:33 docs
drwxr-xr-x 2 eshernan eshernan 4096 2009-12-29 19:33 extras
drwxr-xr-x 4 eshernan eshernan 4096 2009-12-29 19:33 lib
-rw-r--r-- 1 eshernan eshernan 85433 2009-06-17 10:23 LICENSE
-rw-r--r-- 1 eshernan eshernan 1536 2009-06-17 10:23 NOTICE
drwxr-xr-x 6 eshernan eshernan 4096 2009-12-29 19:33 printable_docs
-rw-r--r-- 1 eshernan eshernan 5071 2009-06-17 10:23 README
```

Dentro de ella encontramos 3 archivos que serán los de mayor interés,

```
-rw-r--r-- 1 eshernan eshernan 28333 2009-06-17 10:23 jmeter.properties
-rwxr-xr-x 1 eshernan eshernan 1358 2009-06-17 10:11 jmeter-server
-rwxr-xr-x 1 eshernan eshernan 1372 2009-06-17 10:11 jmeter.sh
```

En el primero de ellos jmeter.sh se crean los planes de pruebas y los recolectores de información, en el jmeter.properties se inscriben los agentes remotos para la ejecución de pruebas distribuidas y el jmeter-server se ejecuta como un agente remoto.

5.7.1 Creación del plan de pruebas

El plan de pruebas se realiza con la ejecución del archivo jmeter.sh, el cual lanza un interfase como la siguiente:

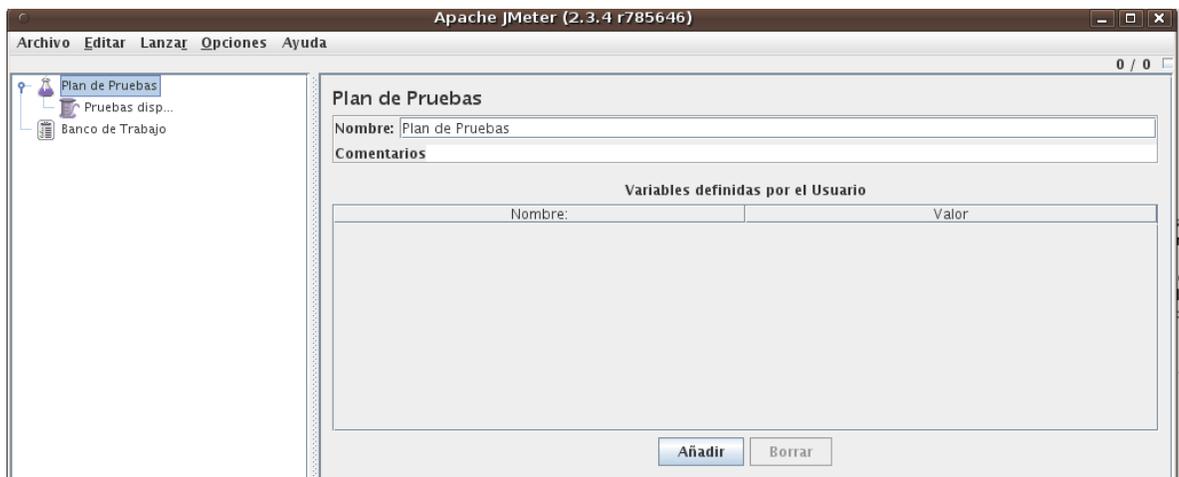


Ilustración 2: Pantalla de inicio de Jmeter

Dentro del plan se crea un conjunto de hilos o thread, que serán los encargados de generar las peticiones hacia el servicio indicado. En la elaboración del conjunto

de hilos se debe tener en cuenta la cantidad, el tiempo entre creación de cada uno de ellos y la cantidad de memoria asignada al cliente de jmeter para poder analizar los resultados de estas peticiones:



Ilustración 3: Generación de grupos de hilos en Jmeter

La realización de las pruebas se montó sobre un ambiente virtualizado siguiendo el esquema de direccionamiento y configuración propuesto.

1. Creación de las peticiones al portal

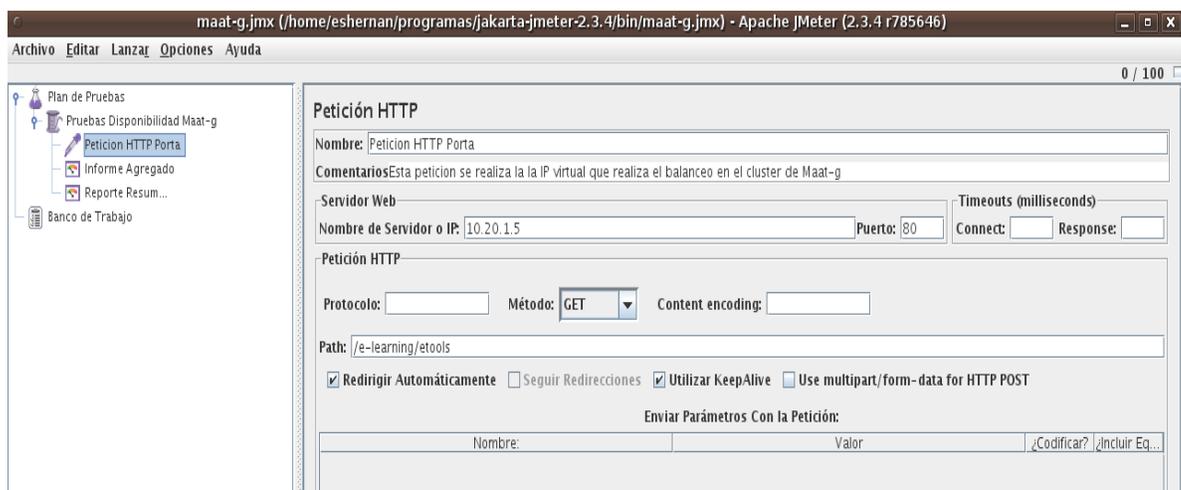


Gráfico 14: Generación de petición http al sistema balanceado

Las peticiones fueron realizadas a la dirección de servicio (ip de servicio 10.20.1.5), la cual estará flotando entre los dos servidores linux respaldados en HeartBeat y que tienen su propia dirección de red (10.20.1.4 y 10.20.1.3).

Esta dirección será mantenida por el HeartBeat y será migrada del servidor que falle, al servidor que se encuentre de respaldo.

Después de esto se deben crear los listener, encargados de almacenar el resultado de las pruebas realizadas y mostrarla de diferentes maneras (en detalle o en resumen, con gráficas o texto, etc).

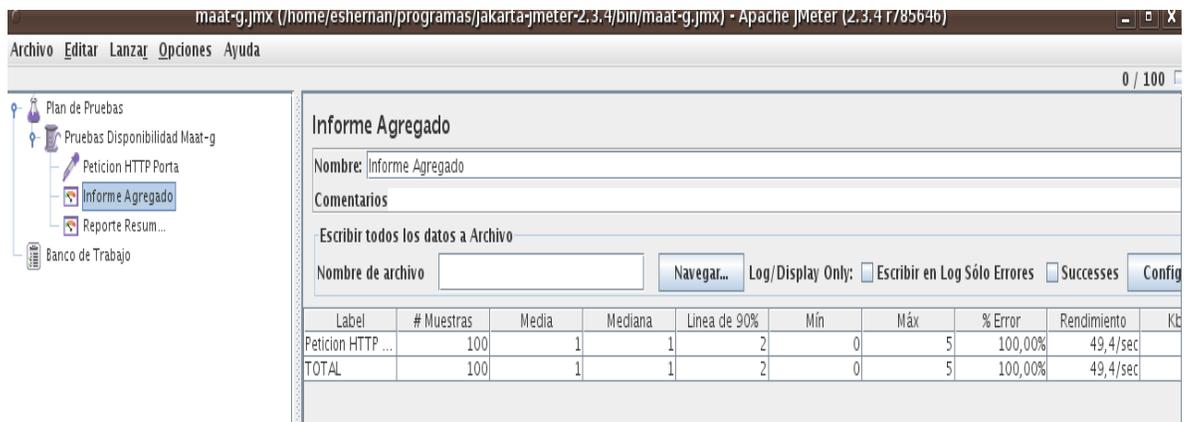


Gráfico 15: Informe agregado de pruebas de disponibilidad

El informe agrega, nos permite obtener un resumen del conjunto de peticiones realizadas, el porcentaje de fallos y la cantidad de muestras.

Con este informe podremos determinar si los SLA propuestos en el presente proyecto se han cumplido o no con la arquitectura propuesta.

Una el test de pruebas y los informes se encuentran completos se debe guardar con una extensión .jmx, como se muestra en el siguiente gráfico. Este archivo será utilizado para lanzar los cliente remotos a fin de realizar una prueba distribuida.



Gráfico 16: Almacenamiento del test de pruebas en archivo.

5.7.2 Instalación de agentes remotos.

Una vez almacenado el plan de pruebas, se debe copiar en los equipos remotos, para la realización de la prueba distribuida, en cada uno de los equipos remotos también se debe instalar el jakarta jmeter con los mismos requisitos ya explicados en el presente manual.

Una vez instalado, se indica que será un agente y que correrá en modo desatendido. Para esto se debe correr el programa jmeter-server.sh de la siguiente manera:

```
./jmeter-server -n -t Maat-g.jmx
```

El archivo Maat-g.jmx debe haber sido copiado con anterioridad del archivo principal donde esté corriendo el jakarta jmeter.

El servidor lanzará una salida semejante a la siguiente, indicando que está tomando una dirección de tipo unicast para la IP que actualmente tiene el nodo. El parámetro -n indica que no se debe lanzar una interfaz gráfica.

```
eshernan@eshernan-laptop:~/programas/jakarta-jmeter-2.3.4/bin$ ./jmeter-server -n -t Maat-g.jmx
Created remote object: UnicastServerRef [liveRef: [endpoint: [192.168.1.9:50252] (local) ,objID: [271cf182:12628fedde7:-7fff,6242031197124628363]]]
```

Después de iniciar los agentes, en el servidor que realizará la consolidación, en el cual se iniciará la interfaz gráfico, es necesario realizar las siguientes modificaciones:

En el archivo jmeter.properties, anexar la lista de ip o nombres de equipos en los cuales ser está corriendo el jmeter-server.sh (agentes), como se indica.

```
# Remote Hosts - comma delimited
remote_hosts=127.0.0.1,192.168.1.9
#remote_hosts=localhost:1099,localhost:2010
```

Los equipos inscritos se deben separar por comas. Después de guardar los cambios se debe reiniciar la interfase gráfica en la cual en la sección de lanzar en la opción de arrancar remoto, aparecen todas los agentes actualmente configurados, como se muestra en la siguiente figura.

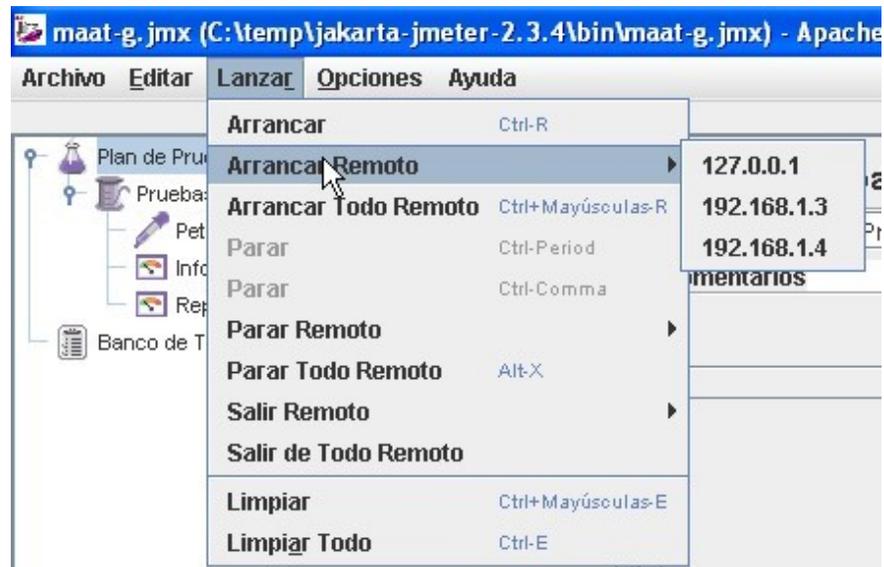


Gráfico 17: Configuración de agentes remotos, para prueba distribuida.

Una vez se tengan los agentes iniciados con el procedimiento anterior, se corre la prueba general que tendrá los resultados totales.

5.7.3 Realización de la prueba.

Para la realización de la prueba se configuró con una interfase master y con 2 agentes remotos corriendo en mismo set de pruebas. Se crearon 100 peticiones por cada agente, incluyendo el servidor principal para un total de 300 peticiones.

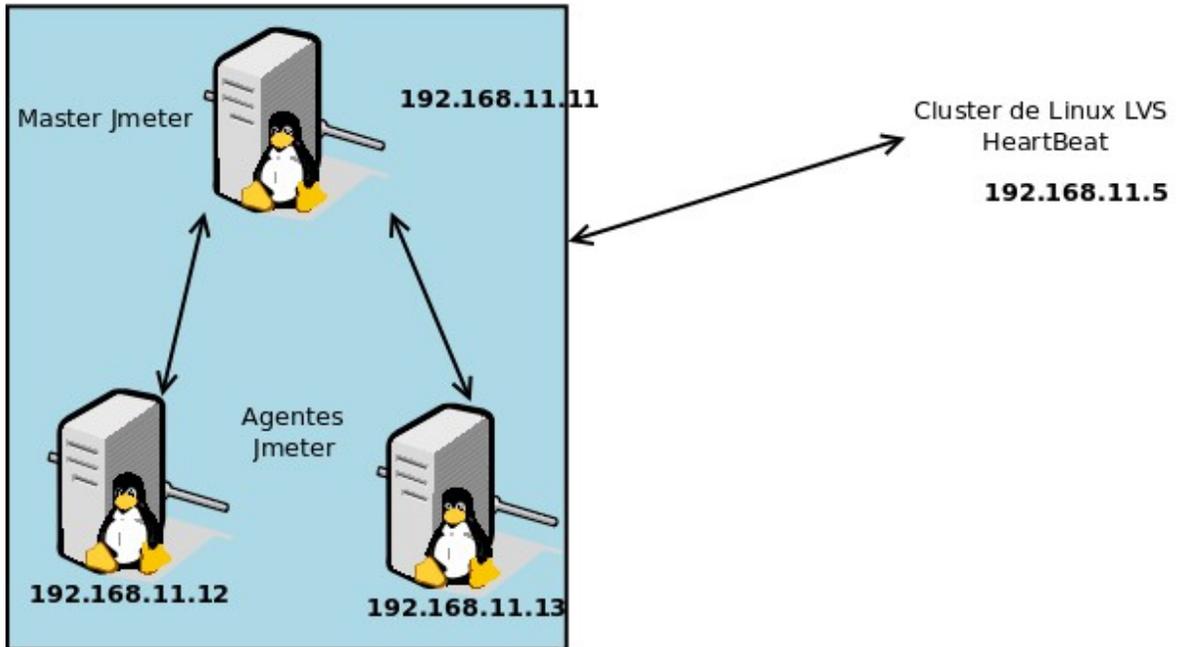


Gráfico 18: Arquitectura para las pruebas de disponibilidad

Una vez configurado de esta manera, lanza la prueba y se toman los resultados para su análisis.

5.7.4 Resultado de las pruebas

En el caso que ninguno de los nodos falle, el resultado es que el 100% de las peticiones efectuadas se pueden procesar, con un tiempo promedio de respuesta de 720 milisegundos, como se puede observar en el siguiente gráfico.

Label	# Muestras	Media	Mín	Máx	Std. Dev.	% Error
Peticion HTTP ...	300	72.1	1	1794	802,53	0,00%
TOTAL	300	72.1	1	1794	802,53	0,00%

Gráfico 19: Resultados de las pruebas sin elementos de falla

5.7.4.1 Caída de 1 nodo del Cluster LVS.

En el caso que uno de los nodos falle, resulta en un porcentaje de 66% de peticiones falladas y el tiempo en que tarda el segundo nodo de estar totalmente operativo con la dirección IP de servicio es de 45 segundos, como se puede observar en el siguiente gráfico.

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento
Peticion HTTP ...	300	65	101	102	3	108	61,33%	19,7/se
TOTAL	300	65	101	102	3	108	61,33%	19,7/se

Gráfico 20: Peticiones erróneas por falla en un nodo del lvs

Los siguiente gráficos muestra el tiempo de transacción del servicio, esto dado por el mensaje

```
info: lvs1:eth0 up
```

Lo cual indica que la interfase es operativo y comienza el proceso de movimiento de recursos.

```
dttime.
heartbeat[6137]: 2010/01/14_14:07:57 info: URL: http://linux-ha.org/FAQ#heavy_load
heartbeat[6137]: 2010/01/14_14:07:57 info: Link lvs1:eth0 up.
heartbeat[6137]: 2010/01/14_14:07:57 WARN: Late heartbeat: Node lvs1: interval 213030 ms
heartbeat[6137]: 2010/01/14_14:07:57 info: Status update for node lvs1: status active
harc[6521]: 2010/01/14_14:07:57 info: Running /etc/ha.d/rc.d/status status
heartbeat[6137]: 2010/01/14_14:07:59 info: Heartbeat shutdown in progress. (6137)
heartbeat[6538]: 2010/01/14_14:07:59 info: Giving up all HA resources.
ResourceManager[6551]: 2010/01/14_14:07:59 info: Releasing resource group: lvs1
10.20.1.5 httpd
ResourceManager[6551]: 2010/01/14_14:08:00 info: Running /etc/init.d/httpd stop
ResourceManager[6551]: 2010/01/14_14:08:00 info: Running /etc/ha.d/resource.d/IPaddr 10.20.1.5 stop
IPaddr[6633]: 2010/01/14_14:08:00 INFO: ifconfig eth0:0 down
IPaddr[6616]: 2010/01/14_14:08:00 INFO: Success
heartbeat[6538]: 2010/01/14_14:08:00 info: All HA resources relinquished.
heartbeat[6137]: 2010/01/14_14:08:02 info: Received shutdown notice from 'lvs1'.
heartbeat[6137]: 2010/01/14_14:08:02 info: Resource takeover cancelled - shutdown in progress.
```

Gráfico 21: Inicio del traslado de recursos al nodo secundario lvs

Y cuando el segundo nodo es declarado completamente operativo, cuando aparece el mensaje

```
remote resource transition complete.
```

```

IPaddr 10.20.1.5 start
Jan 14 14:10:43 lvs1 IPaddr[10948]: INFO: Using calculated nic for 10.20.1.5: eth0
Jan 14 14:10:43 lvs1 IPaddr[10948]: INFO: Using calculated netmask for 10.20.1.5: 255.255.0.0
Jan 14 14:10:43 lvs1 IPaddr[10948]: INFO: eval ifconfig eth0:0 10.20.1.5 netmask 255.255.0.0 broadcast 10.20.255.255
Jan 14 14:10:43 lvs1 avahi-daemon[2117]: Registering new address record for 10.20.1.5 on eth0.
Jan 14 14:10:43 lvs1 avahi-daemon[2117]: Withdrawing address record for 10.20.1.5 on eth0.
Jan 14 14:10:43 lvs1 avahi-daemon[2117]: Registering new address record for 10.20.1.5 on eth0.
Jan 14 14:10:44 lvs1 IPaddr[10931]: INFO: Success
Jan 14 14:10:44 lvs1 ResourceManager[10845]: info: Running /etc/init.d/httpd start
Jan 14 14:10:44 lvs1 heartbeat: [10706]: info: remote resource transition completed.

```

Gráfico 22: Traslado completo de servicios al nodo de respaldo lvs

5.7.4.2 Caída de 1 Nodo de la granja de servidores apache, perl y LMS.

Este tipo de caída es la que menos impacta la disponibilidad de la aplicación, ya que por tratarse de una granja de servidores que balancean cada nueva petición mediante un algoritmo roun-drobin, las fallas se generan mientras el balanceador marca el nodo como no disponible y le deja de enviar peticiones, para nuestro conjunto de pruebas el porcentaje de error fue del 11%, con un tiempo máximo de indisponibilidad de 25 segundos, por cada nodo que se caiga.

Label	# Muestras	Media	Min	Máx	Std. Dev.	% Error	Rendimiento	Kb/sec	Avg. Bytes
Peticion HTTP ...	300	7	1	71	9,17	11,00%	7,5/sec	34,10	4657,2
TOTAL	300	7	1	71	9,17	11,00%	7,5/sec	34,10	4657,2

Gráfico 23: Peticiones fallidas por nodo caído en la granja de apache

5.7.4.3 Caída de 1 Nodo del Cluster de mysql.

En este caso, el porcentaje de error en las peticiones aumentó al 78%, debido al tiempo de subida del nodo de respaldo de mysql y a la necesidad de sincronización de los logs binarios para replicación que posee, como se puede observar en la siguiente figura:

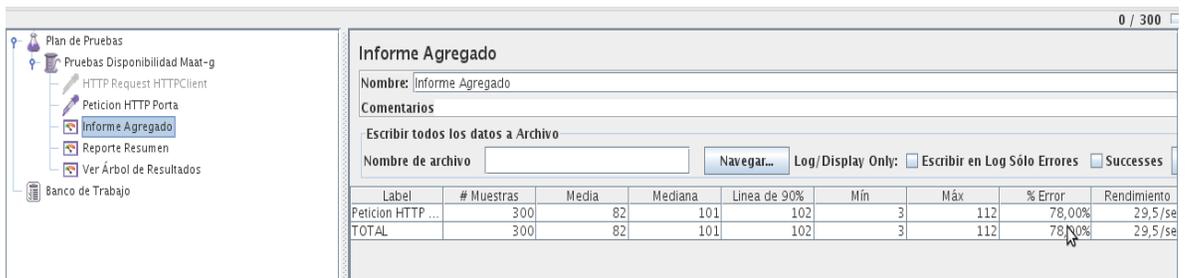


Gráfico 24: Fallo de peticiones con caída de nodo mysql

6 ANEXO 2.

6.1 MODELO DE ENTREVISTA REALIZADA

Fecha: _____

Institución, empresa o proyecto: _____

Descripción de la plataforma de hardware utilizada:

Descripción de la plataforma de software implementada.

Descripción de los SLA que aseguran la plataforma.

Nivel de disponibilidad esperada de la plataforma:

Resuman cuáles son los aspectos más importantes que el equipo técnico ha tenido en cuenta la ahora de elegir la plataforma.

Algún o algunos de los elementos de la plataforma está soportados en software libre o de fuente abierta?, en caso afirmativo por favor descríbalos.

Describas la razones por las cuales se a elegido o no implementar software libre dentro de la actual plataforma:
