



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES**  
**Y REDES**

**ESTUDIO COMPARATIVO DE SOLUCIONES DE BALANCEO DE**  
**CARGA A NIVEL SOFTWARE Y HARDWARE PARA EL**  
**SISTEMA ACADEMICO ESPOCH**

Trabajo de titulación presentado para optar al grado académico de

**INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y**  
**REDES**

**AUTOR:** RICARDO RAÚL AGUILAR LEÓN

**TUTOR:** ING. ANDRES SANTIAGO CISNEROS BARAHONA

Riobamba-Ecuador

2016

©2016, Ricardo Raúl Aguilar León.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

# **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

## **FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

### **ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES**

El Tribunal de Trabajo de Titulación certifica que: El trabajo de Titulación: “ESTUDIO COMPARATIVO DE SOLUCIONES DE BALANCEO DE CARGA A NIVEL SOFTWARE Y HARDWARE PARA EL SISTEMA ACADEMICO ESPOCH”, de responsabilidad del señor Ricardo Raúl Aguilar León, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

Ing. Washington Luna

**DECANO FACULTAD DE  
INFORMÁTICA Y  
ELECTRÓNICA**

\_\_\_\_\_

Ing. Franklin Moreno

**DIRECTOR DE ESCUELA**

\_\_\_\_\_

Ing. Santiago Cisneros.

**DIRECTOR DE TESIS**

\_\_\_\_\_

Ing. Edwin Altamirano.

**MIEMBRO DEL TRIBUNAL**

\_\_\_\_\_

Yo, Ricardo Raúl Aguilar León, soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del Trabajo de Titulación pertenece a la Escuela Superior Politécnica De Chimborazo.

RICARDO RAÚL AGUILAR LEÓN

## **DEDICATORIA**

Dios el responsable directo de permitirme culminar una etapa en mi vida donde el sacrificio y esfuerzo se vio plantado en la educación que me brindaron, los docentes que se volvieron amigos brindando su tiempo en valores y enseñanzas de vida en toda mi carrera.

Así mismo a mis papas Gladis Carmita, Ricardo Aguilar por ser el pilar de todo esto, ya que ellos fueron mis representantes en todo momento de mi carrera estudiantil, enseñándome los valores y responsabilidades que uno puede brindar, sin dejar a lado el apoyo total que me brindaron todo este tiempo les quedo eternamente agradecido.

**Ricardo.**

## AGRADECIMIENTO

De la mano de Dios y mis padres que todo este tiempo me brindaron salud y se esforzaron por que nada me haga falta así mismo enseñándome a valorar las cosas que me otorgaban se las obtiene con esfuerzo que en esta vida nada es fácil, mama y papa aquí me tienen terminando una etapa más en vida juntos a ustedes, les quedo agradecido por tanto en esta vida.

Al Departamento de Tecnologías de Información y Comunicación de la ESPOCH por brindarme el apoyo y confianza por ser partícipe de un estudio en su establecimiento donde nada me hizo falta ya que me brindaron toda su ayuda en especial a los Ing. David Garcés por el apoyo y supervisión que me brindo en este tiempo que duro el trabajo de titulación.

A mi Director de Tesis Ing. Santiago Cisneros por su apoyo incondicional que me brindo en un momento tan esperado para mí y toda mi familia, me brindo sus conocimientos y guías de enseñanza, el Ing. Edwin Altamirano que me colaboro con la supervisión académica que necesitaba para poder culminar este gran paso en mi vida.

Ricardo.

## TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE ILUSTRACIONES.....	xi
RESUMEN	
SUMMARY .....	xiv
INTRODUCCIÓN .....	1

## CAPÍTULO I

<b>1.</b>	<b>MARCO TEÓRICO.....</b>	<b>7</b>
<b>1.1.</b>	<b>Introducción.....</b>	<b>7</b>
<b>1.2.</b>	<b>Servidor web .....</b>	<b>7</b>
<i>1.2.1.</i>	<i>Características del servidor web.....</i>	<i>8</i>
<i>1.2.2.</i>	<i>Componentes de un servidor web.....</i>	<i>8</i>
<b>1.3.</b>	<b>Servicios web.....</b>	<b>9</b>
<b>1.4.</b>	<b>Ataque DoS.....</b>	<b>10</b>
<b>1.5.</b>	<b>Clustering .....</b>	<b>11</b>
<i>1.5.1.</i>	<i>Definición.....</i>	<i>11</i>
<i>1.5.2.</i>	<i>Arquitectura.....</i>	<i>12</i>
<i>1.5.3.</i>	<i>Clasificación .....</i>	<i>13</i>
<i>1.5.4.</i>	<i>Características .....</i>	<i>13</i>
<i>1.5.4.1.</i>	<i>Escalamiento.....</i>	<i>13</i>
<i>1.5.4.2.</i>	<i>Disponibilidad.....</i>	<i>15</i>
<i>1.5.4.3.</i>	<i>Rendimiento.....</i>	<i>16</i>
<b>1.5.5.</b>	<b>Balaceo de carga .....</b>	<b>16</b>
<i>1.5.5.1.</i>	<i>Funcionamiento.....</i>	<i>16</i>
<i>1.5.5.2.</i>	<i>Métodos de conexiones .....</i>	<i>17</i>
<i>1.5.5.3.</i>	<i>Balaceo de carga por hardware.....</i>	<i>17</i>
<i>1.5.5.4.</i>	<i>Algoritmos de balaceo de carga que oferta Citrix Netscaler.....</i>	<i>18</i>
<i>1.5.5.5.</i>	<i>Balaceo de carga por software .....</i>	<i>34</i>

1.5.5.6.	<i>Opciones de afinidad múltiple host</i> .....	36
1.5.5.7.	<i>Modo de operación del clúster NLB</i> .....	37
1.5.6.	<i>Alta disponibilidad con heartbeat</i> .....	38
1.5.7.	<i>Propiedades del equilibrio de carga de red</i> .....	39
1.5.8.	<i>Control de servidores en NLB</i> .....	39
1.5.9.	<i>Mecanismos de balanceo</i> .....	41
1.5.10.	<i>Tipos de balanceo de carga</i> .....	42
1.6.	<b>Estructura la red del DTIC</b> .....	44
1.6.1.	<i>Topología de la red</i> .....	44
1.6.2.	<i>Número de usuarios aproximados</i> .....	44
1.6.3.	<i>Recursos del clúster</i> .....	44
1.7.	<b>Soluciones implementadas</b> .....	44
1.7.1.	<i>Solución con software</i> .....	44
1.7.2.	<i>Solución con hardware</i> .....	45
1.7.2.1.	<i>Selección y configuración de persistencia</i> .....	46
1.7.2.2.	<i>Configuración de la persistencia basada en cookies</i> .....	47
1.7.2.3.	<i>Configuración de la persistencia basada en los ids de los servidores en URL</i> .....	49

## **CAPITULO II**

2.	<b>MARCO METODOLÓGICO</b> .....	50
2.1.	<b>Introducción</b> .....	50
2.2.	<b>Escenario de pruebas</b> .....	50
2.3.	<b>Selección de las soluciones por hardware y por software a través del método cualitativo por puntos</b> .....	53
2.3.1.	<i>Selección de hardware</i> .....	54
2.3.2.	<i>Selección de software</i> .....	57
2.4.	<b>Configuración de las soluciones</b> .....	59
2.4.1.	<i>Configuración solución por software</i> .....	59
2.4.2.	<i>Configuración solución por hardware</i> .....	65



## **CAPITULO III**

<b>3.</b>	<b>MARCO DE RESULTADOS .....</b>	<b>69</b>
<b>3.1.</b>	<b>Evaluación del funcionamiento por software.....</b>	<b>69</b>
<b>3.2.</b>	<b>Evaluación del funcionamiento por hardware.....</b>	<b>71</b>
<b>3.3.</b>	<b>Análisis comparativo de las soluciones utilizadas.....</b>	<b>73</b>
	<b>CONCLUSIONES.....</b>	<b>76</b>
	<b>RECOMENDACIONES.....</b>	<b>78</b>
	<b>BIBLIOGRAFÍA</b>	
	<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

Tabla 1-1. Ejemplo de transacciones método least connection Citrix Netscaler.

Tabla 2-1. Ejemplo de transacciones método least response time.

Tabla 3-1. Resumen cálculo de tiempos de respuesta de los distintos monitores.

Tabla 4-1. Resumen cálculo de tiempos de respuesta de los distintos monitores.

Tabla 5-1. Resumen cálculo de tiempos de respuesta de los distintos monitores.

Tabla 6-1. Parámetro de máscara de red.

Tabla 7-1. Solución elegida en software.

Tabla 8-1. Solución elegida para hardware.

Tabla 9-1. Tipos de persistencia para cada tipo de servidor virtual.

Tabla 1-2. Tabla de datos técnicos de cada solución por hardware.

Tabla 2-2. Tabla de ponderaciones relativas de soluciones por hardware.

Tabla 3-2. Determinación de ponderaciones de cada parámetro solución por hardware.

Tabla 4-2. Cuadro de calificaciones solución por hardware.

Tabla 5-2. Cuadro de calificaciones solución por hardware.

Tabla 6-2. Tabla de datos técnicos de cada solución por software.

Tabla 7-2. Tabla de ponderaciones relativas de soluciones por software.

Tabla 8-2. Cuadro de calificaciones solución por software.

Tabla 9-2. Cuadro de calificaciones solución por software.

Tabla 1-3. Datos estadísticos comprendidos desde el día uno hasta el día cinco de solución NLB.

Tabla 2-3. Datos estadísticos comprendidos desde el día seis hasta el día diez de solución NLB.

Tabla 3-3. Datos estadísticos comprendidos desde el día uno hasta el día cinco de solución Citrix Netscaler.

Tabla 4-3. Datos estadísticos comprendidos desde el día seis hasta el día diez de solución Citrix Netscaler.

Tabla 5-3. Cuadro comparativo NLB vs Citrix NetScaler.

## ÍNDICE DE FIGURAS

- Figura 1-1. Número total de estudiantes por período académico
- Figura 2-1. Componentes de un servidor web.
- Figura 3-1. Escenario ideal de un servicio web.
- Figura 4-1. Clúster de alta disponibilidad y balanceo de carga
- Figura 5-1. Arquitectura de un clúster
- Figura 6-1. Escalabilidad vertical
- Figura 7-1. Escalabilidad horizontal
- Figura 8-1. Esquema de funcionamiento balanceo de carga
- Figura 9-1. Método de menor conexión (Least Connection)
- Figura 10-1. Método Round Robin
- Figura 11-1. Método Round Robin con pesos
- Figura 12-1. Método de menor tiempo de respuesta (Least response time)
- Figura 13-1. Método de menor tiempo de respuesta (Least response time Monitors)
- Figura 14-1. Diagrama de flujo del trabajo de los métodos hash
- Figura 15-1. Método URL hash
- Figura 16-1. Logo Citrix NetScaler
- Figura 17-1. Dashboard
- Figura 18-1. Configuration/Traffic Management/Load Balancing/Servers
- Figura 19-1. Configuration/Traffic Management/Load Balancing/Virtual Servers
- Figura 20-1. Reporting/System/CPUvsMemory Usage and HTTP Requests Rate
- Figura 21-1. Logo AVANU WebMux
- Figura 22-1. Logo Barracuda
- Figura 23-1. Posible configuración con NLB
- Figura 24-1. Panel de configuración de parámetros del clúster NLB
- Figura 25-1. Servidores virtuales linux
- Figura 26-1. Perlbal balanceador de carga
- Figura 27-1. Arquitectura del balanceo de carga dinámico centralizado
- Figura 28-1. Arquitectura del balanceo de carga dinámico distribuido
- Figura 29-1. Arquitectura red DTIC con solución de balanceo por software
- Figura 30-1. Arquitectura red DTIC con solución de balanceo por hardware
- Figura 31-1. Comandos de configuración de persistencia por cookies.
- Figura 32-1. Comandos de configuración de persistencia por URL PASSIVE.
- Figura 1-2. Número máximo de conexiones.
- Figura 2-2. Escenario de balanceo por software

Figura 3-2. Escenario de balanceo por hardware

Figura 4-2. Especificación del servidor.

Figura 5-2. Conexión del Citrix con el Data Center ESPOCH

Figura 6-2. Server manager >> Features >> Add features.

Figura 7-2. Pantalla Select Features del Server Manager.

Figura 8-2. Crear nuevo clúster.

Figura 9-2. Parámetros clúster.

Figura 10-2. Direcciones ip del clúster

Figura 11-2. Reglas de puertos de clúster

Figura 12-2. Conexión al servidor.

Figura 13-2. Parámetros de host.

Figura 14-2. Pantalla principal con un clúster creado y un único host

Figura 15-2. Pantalla principal los hosts añadido al clúster.

Figura 16-2. Pantalla de autenticación Citrix.

Figura 17-2. Pantalla de configuración del direccionamiento Citrix

Figura 18-2. Pantalla de configuración de servidores.

Figura 19-2. Pantalla de configuración de seguridad por autenticación

Figura 20-2. Pantalla de interfaz de monitoreo NetScaler.

Figura 1-3. Servidor caído por saturación.

Figura 2-3. Servidor activo.

Figura 3-3. Gráfica de rendimiento de Citrix NetScaler

Figura 4-3 Porcentaje uso de recurso durante trabajo Citrix Netscaler

## **RESUMEN**

El objetivo fue crear una apreciación completa del escenario en el cual se desenvuelve la red de Departamento de Tecnologías de Información y Comunicación (DTIC), de esta manera se planteó dos tipos de soluciones de balanceo de carga; la primera comprendida por el balanceo por software y la segunda a través de balanceo por hardware. Se incluyó en el trabajo descripciones que pueden servir de guía para la elección o implementación de tecnologías de clustering de servidores, una breve descripción, su arquitectura, características notables y clasificación, todo ello con el propósito de tutelar al lector de mejor manera a lo largo del estudio despejando las dudas acerca de los servicios web. Las soluciones elegidas para el estudio fueron el software Network Load Balancing (NLB) y el dispositivo Citrix NetScaler. Para ambas soluciones se realizó una cobertura exhaustiva con respecto a sus características y parámetros técnicos, se identificó los algoritmos de distribución de carga que cada uno puede solventar con una concreta definición. Se llevó a cabo además la implementación de ambas herramientas con el propósito de generar de esta forma datos que a su vez permitan a posterior crear una etapa de análisis comparativo entre ambas soluciones. El estudio demostró que la solución por hardware con respecto al número de conexiones establecidas es 30% más eficiente que el método por software. Se concluyó que la solución por hardware es la que mejor se adapta a la red, siendo un sistema robusto, con una adquisición que se convierte en una inversión que permite ganar o ganar. Se recomienda una vez encontrado el problema con respecto al sistema de bases de datos se tomen los correctivos apropiados.

**PALABRAS CLAVE:** <TECNOLGIA Y CIENCIAS DE LA INGENERIA>, <INGENIERÍA EN TELECOMNCACIONES Y REDES>, < BALANCEADOR DE CARGA >, < SERVIDOR WEB >, < CLUSTER DE SERVIDORES >, < SOFTWARE (NLB) >, < SISTEMA ACADEMICO OASIS >, < CITRIX NETSCALER (HARDWARE)>

## SUMMARY

The objective was to create a complete appreciation of the scenario in which the department of Information and Communication Technologies (DTIC); in this way, proposing two types of load balancing solutions; the first comprised by software balancing and the second through hardware balancing. It is included in the job descriptions that can serve as a guide for the election or implementation of clustering technologies of servers, a brief description, its architecture, notable features and classification, all with the purpose of protecting the reader in a better way throughout the study clearing the doubts about the web services. The solutions chosen for the study were the Network software Load Balancing (NLB) and Citrix Netscaler device. For both solutions are conducted a comprehensive coverage with respect to its characteristics and technical parameters, identified the load distribution algorithms that each one can solve with a concrete definition. It was carried out the implementation of both tools with the purpose of generating this way data that in turn allow to later create a phase of comparative analysis between the two solutions. The study showed that the hardware solution with regard to the number of connections established is 30% more efficient than the method by software. It was concluded that the solution by hardware is the one that best adapts to the network, being a robust system, with an acquisition that becomes an investment that allows win or win. It is recommended once you have found the problem with regard to the system of databases is to take the appropriate corrective measures.

**KEY WORDS:** <TECHNOLOGY AND THE ENGINEERING SCIENCES>,<ENGINEERING IN TELECOMMUNICATIONS AND NETWORKS>,<LOAD BALANCER>,<WEB SERVER>,<CLUSTER OF SERVERS>,<NETWORK LOAD BALANCING (SOFTWARE)>,<ACADEMIC SYSTEM OASIS>,<CITRIX NETSCALER (HARDWARE)>.

## INTRODUCCIÓN

Actualmente vivimos un constante cambio tecnológico, día a día se desarrollan nuevas tecnologías, dispositivos inteligentes que permiten estar conectados con el mundo en todo momento a través de Internet, ya sea por ocio, trabajo, salud, educación y un sin fin de otros motivos que hacen de nuestras vidas casi dependan por completo de ello.

La información que transmitimos desde nuestros dispositivos viaja por la red hacia grandes servidores de datos que nos brindan sus servicios, algunos de manera gratuita, otros con un costo por cada nuevo producto que nos ofrecen, pero siempre buscando garantizar la confidencialidad y disponibilidad de datos.

El mundo se está orientando a las tecnologías de almacenamiento en la nube (Cloud Computing), que va más allá de solo guardar datos, se los pueden gestionar, actualizar y modificar en tiempo real, visualizarlos desde cualquier dispositivo. Muchas instituciones vienen ya trabajando con estas tecnologías, en especial los centros de educación, que mantienen su información centralizada de los estudiantes, docentes, trabajadores, es decir, realizan una gestión eficiente sobre los servicios que brindan, los cuales deben ser transparentes para los usuarios.

Para lograr las comunicaciones y utilización de estas nuevas tecnologías es necesario la implementación de equipos adecuados al volumen de datos que van a manejar, pero en ocasiones se necesitaría una inversión extremadamente alta para lograr esto. Ventajosamente existen tecnologías como el Clustering, que nos facilitarán la implementación de grandes servicios para gestión de datos, a un relativamente bajo costo.

Cuando se trabajan con datos es imprescindible realizar un análisis minucioso sobre las proyecciones y crecimiento en nuestra red, uno de los principales problemas que presentan las instituciones educativas año tras año, el cual va de la mano con el incremento exponencial de usuarios que acceden a la información almacenada.

Si existen muchos usuarios en un momento determinado queriendo acceder a un servicio web, se generarán varias peticiones en el servidor, las cuales podrán ser atendidas dependiendo de la capacidad de los equipos para no llegar a saturarse.

Para una buena administración de peticiones será necesario balanceadores de carga, como pueden ser tanto software como hardware que se encargan de distribuir apropiadamente las peticiones a los servidores activos dentro del Clúster, evitando fallas y garantizando el acceso eficiente al servicio requerido.

Se necesita tener acceso a servidores, conocer la topografía de la red en la que se está trabajando, las horas pico donde sucede una alta demanda de peticiones hacia los servidores del sistema académico para poder identificar el algoritmo de balanceo tanto software como hardware que mejor se acople a la red.

El presente trabajo busca analizar las mejores alternativas de balanceadores de carga para mejorar el acceso y disponibilidad de los datos en una institución educativa, que servirá de guía para otros centros educativos que presenten problemas similares.

Se creará un escenario de igual similitud para ambos balanceadores para la obtención de datos, así mismo se escogerán varios balanceadores de software para determinar por medio de la validación por puntos cual es el adecuado para trabajar en la red mientras que en el balanceo de hardware será el mismo proceso además se debe verificar la garantía y soporte que esto es muy importante cuando se trata con dispositivo físico.

## **Antecedentes**

A menudo los servicios que brindan sitios web en internet presentan inconvenientes que pueden causar fallas a los sistemas o comprometer los datos almacenados. Uno de los problemas más comunes se da al gestionar grandes cantidades de solicitudes generadas por los usuarios que pueden llegar a saturar el servidor. Se evidencia como un problema de escalabilidad, ya que se debe a una falta recursos.

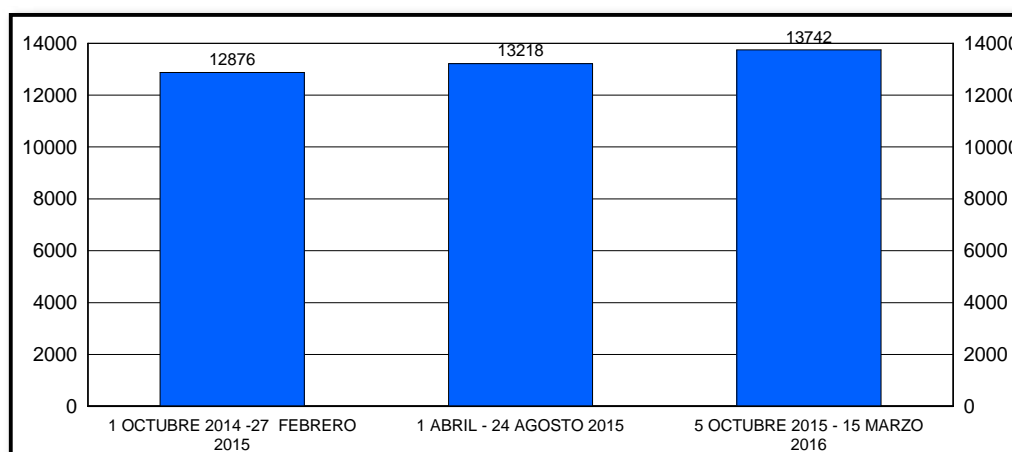
Una opción es reemplazar los sistemas por los llamados supercomputadores, que son equipos con capacidades de procesamiento extremadamente rápido, pero se encuentran ligados a un excesivo costo. Como alternativa a esto se emplea el Clustering, el cual consiste en utilizar un conjunto de equipos de computación de igual o distintas características y hacerlo funcionar como un único servidor, esto se consigue al conectar todos los equipos a una red de datos de alta velocidad. (Bernal, 2012, pp.4-10)

Los clústeres son utilizados como alternativa de escalabilidad, disponibilidad, o eficiencia de una red, principalmente nos enfocaremos en su utilización para balanceo de carga.

El sistema OASIS de la Escuela Superior Politécnica de Chimborazo (ESPOCH), es el servicio más utilizado tanto por docentes como por estudiantes, para ingreso y revisión de datos académicos (notas, matriculas, horarios, entre otras), este servicio desde hace tiempo atrás viene presentando congestión y molestia por parte de los usuarios quienes reportan dificultad al conseguir cupos para matricularse.



En los periodos académicos comprendidos entre 1 de Octubre 2014 – 27 Febrero 2015 hasta el 5 de Octubre 2015 – 15 marzo 2016, se presentó un aumento de usuarios (Figura 1-1), el cual se ve reflejado en el ancho de banda que los estudiantes disponen para ingresar al sistema académico, razón por la cual no llegan a obtener un servicio adecuado para realizar su matriculación en línea, esto sucede por la gran demanda de peticiones que realizan los usuarios al momento de intentar ingresar al sistema OASIS.



**Figura 1-1.** Número total de estudiantes por período académico  
Fuente: DTIC, 2015-2016.

En las fechas con mayor saturación de los servidores, se evidenciaba que era por una carga excesiva de denegaciones de servicio (DoS) que se recibían por las peticiones de los estudiantes al momento de gestionar sus matrículas, se notó la falencia que presentaba en el sistema académico en donde resulto trascendental dar paso a un balanceo de cargas mediante software realizado por el Ing. Alex Tacuri responsable del sistema lo cual mejoró el servicio creando también varias opciones para el ingreso de usuarios y para evitar la saturación de servicios que se ocasiona.

Con la creciente demanda de servicios de red, el DTIC ha visto la necesidad de mejorar la solución de balanceo de carga, tomando en cuenta la posibilidad de una implementación a nivel de hardware, mismo que se encargará de establecer que las solicitudes lleguen a los servidores abasteciendo el tráfico de peticiones y cubriendo la demanda de las denegaciones de servicios que se genera en la plataforma virtual de la institución.

### **Justificación del Trabajo de Titulación**

## **Justificación Teórica**

Con la aparición de nuevas tecnologías tanto de software como hardware, se crea en los administradores de red, la necesidad constante de mejorar los servicios web que ofrecen para abastecer las solicitudes de sus usuarios que crecen exponencialmente con las alternativas de acceso a internet.

Las instituciones educativas no son ajenas a esto, es así, el caso de la Escuela Superior Politécnica de Chimborazo (ESPOCH) que gestiona todos los datos académicos de los estudiantes por medio de su sistema OASIS, el cual se encarga de registrar datos informativos de cada usuario, cupos para matrículas y asignaturas, control, actualización y revisión de notas, entre otras. Mismo que es administrado por el Departamento de Tecnologías de Información y Comunicación (DTIC), el cual se encarga de mantener el correcto funcionamiento de los sistemas, generando políticas de control y mantenimiento para garantizar la integridad, disponibilidad y confidencialidad de la información.

En los últimos años la cantidad de usuarios que acceden al sistema en periodos de matrículas ha excedido la capacidad de los sistemas, esto genera una denegación de servicio (DoS) producida por las peticiones que imposibilita el acceso a los servicios y recursos del sistema académico durante un período indefinido de tiempo.

Los elementos correctamente configurados con balanceos de carga, detección de anomalías en el tráfico de red, bloqueo de fuentes de tráfico automatizadas en base a anomalías pueden ayudarnos a soportar casi cualquier tipo de DoS, aunque todo dependerá de la cantidad de recursos que se disponga.

El funcionamiento de la DTIC respecto a la infraestructura de red establece que el tráfico que viene tanto de internet como de la red interna de la institución, hacia los servidores del sistema académico, se tenga que analizar en qué momento tiene que ser balanceado o distribuido eficientemente entre los nodos del clúster que forman el sistema académico, esto se consigue gracias al Balanceador de carga instalado ya sea por Software, o con Hardware específico.

## **Justificación Aplicativa**

Al momento de configurar el nodo de balanceo se podrá elegir entre una serie de algoritmos para ver cómo se distribuirá la carga entre los servidores y cómo se elegirá el servidor al que se envía cada petición.

Un balanceo de carga por software son servidores configurados de potencia reducida, esto implica instalar en un computador, un sistema operativo y una aplicación que se encargue del balanceo de carga que tiene un bajo costo que a pesar de que cuando se quiera cambiar de balanceador este se lo podrá utilizar para otras tareas sin necesidad de que se lo descarte por completo ahora si nos vamos por un balanceador por hardware veremos que es un dispositivo específico para el balanceo de carga que presenta la mejora de escalabilidad y potencia en las infraestructuras de las aplicaciones.

Las configuraciones del balanceo de carga permiten redistribuir carga entre procesadores a tiempo de ejecución. El objetivo fundamental de estos algoritmos es mejorar el desempeño de una aplicación. Esto implica la reducción de los tiempos de respuesta, reducción del problema de escalabilidad mediante uso eficiente de recursos de cómputo.

Para estos tipos de algoritmos se deben tomar en cuenta la cantidad de carga en un procesador y/o en el sistema, capacidad de procesamiento de los equipos de cómputo, costo de procesamiento de la carga, de comunicación, entre otros. Con este fin, existen aspectos que ayudan a diseñar el algoritmo de balance de tal forma que sea eficiente y que se ajuste a la infraestructura de la DTIC junto con los servicios y software que brindan a los usuarios.

### **Formulación del problema**

¿Qué tipo de solución mediante el uso de las tecnologías de distribución de carga y el clustering es la más adecuada, apta y adaptable para mejorar el servicio de acceso por parte de los estudiantes al sistema académico de la ESPOCH?

### **Metodología**

Método de investigación documental

- ) Se investigará el funcionamiento y aplicaciones referentes a balanceo de carga por software y hardware, mediante recopilación de información en varios medios, principalmente en libros, artículos, revistas, papers.

Método de investigación descriptiva

- J) Mediante la utilización de software especializado se realizará la recopilación de datos sobre las infraestructuras y configuraciones existentes, para comprender el funcionamiento de la red y cuál será la solución más acertada para cumplir los objetivos planteados.

## **Técnicas**

Las técnicas que se utilizaran en el proyecto investigativo, son:

- Observación de funcionamiento.
- Experimentación de procedimientos.
- Análisis de los datos.
- Validación de resultados.

## **Objetivos**

### **Objetivo General:**

Estudiar soluciones de balanceo de carga a nivel software y hardware para el sistema académico ESPOCH.

### **Objetivos Específicos:**

- J) Analizar la infraestructura del DTIC sobre la cual se operan, calidad de servicio y soporte.
- J) Estudiar problemas en el acceso al Sistema Académico OASIS que determine la factibilidad del balanceo de carga a nivel de hardware y/o software.
- J) Ejecutar el estudio comparativo para destacar el balanceo de carga más adecuado.
- J) Seleccionar la solución de Balanceo de Carga en el sistema académico de la ESPOCH.

# CAPÍTULO I

## 1. MARCO TEÓRICO

### 1.1. Introducción

En la actualidad estar conectados con el mundo se ha vuelto una necesidad casi indispensable, siempre buscando nuevas formas de agilizar los procesos, gestionar datos, ofrecer y mejorar los servicios, para satisfacer la demanda de usuarios. Tales procesos se llevan a cabo de una manera transparente para los clientes, es decir, estos no necesitan conocer todo el trabajo que existe detrás de una petición web o SQL, simplemente esperan ver la respuesta a su solicitud.

Mientras mayor sea la demanda de un servicio mayor será la carga que deban procesar los servidores, que en ocasiones llegan a saturarse debido a sus propias limitaciones. Existen varias alternativas como posibles soluciones a estos tipos de problemas, una de ellas es la implementación de clústeres de servidores los cuales permiten mejorar considerablemente un servicio sin la necesidad de realizar grandes cambios de infraestructura, logrando un escalamiento óptimo de la red.

El análisis presentado en este proyecto muestra las características principales de la utilización de clústeres orientados al balanceo de carga, tanto de software como hardware, enfocándonos en las mejoras que tendrán su implementación en el sistema académico de la institución y a su vez servirá como una guía de referencia para futuras aplicaciones.

### 1.2. Servidor web

Su funcionamiento tiene como principio el permitir la interacción entre ordenadores a la espera de peticiones hechas por usuarios, una vez que las peticiones son escuchadas por el servidor este tiene la tarea de responder a ellas utilizando hipertexto. El protocolo que utilizan los servidores web es HTTP (Hypertext Transfer Protocol).

Comúnmente se suele nombrar al ordenador físico que está ejecutando este programa como servidor.

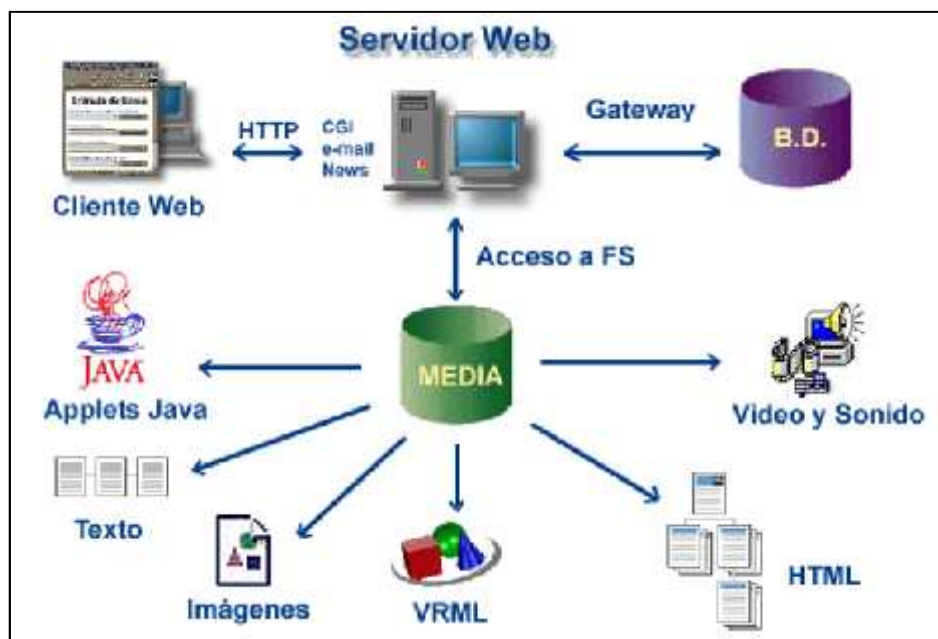
### 1.2.1. Características del servidor web

Dentro de las principales características de un servidor web se puede nombrar las siguientes:

- J Se encuentra a la espera del arribo de peticiones.
- J Envía archivos de acuerdo a la solicitud tomada.
- J Ejecuta CGIs como respuesta a las peticiones y a su vez realiza el envío de resultados.
- J Establece conexión o conexiones a las bases de datos.
- J Su configuración permite cumplir tareas de Gateway (Puerta de enlace) para otros servicios como ftp, correo, etc.

### 1.2.2. Componentes de un servidor web

En el siguiente gráfico (figura 2-1) podemos diferenciar cada uno de los elementos que componen o son parte o injerencia en un servidor web.



**Figura 2-1.** Componentes de un servidor web.  
Fuente: <http://apache.org>

En la imagen se aprecia al servidor web interactuando directamente y en doble vía con tres elementos importantes; el principal que es el cliente, quien realiza las peticiones y recibe respuestas para ser interpretadas y mostrarse como una página web, entre otros, el servidor utiliza el acceso a la multimedia de donde toma en base a lo solicitado un archivo que puede ser HTML, de video, de sonido, imágenes, y más, para reenviarlas al cliente y finalmente se observa un

sistema de base de datos que a través de la pasarela de conexión puede el servidor web tener acceso a la información organizada en estos grandes almacenes.

### **1.3. Servicios web**

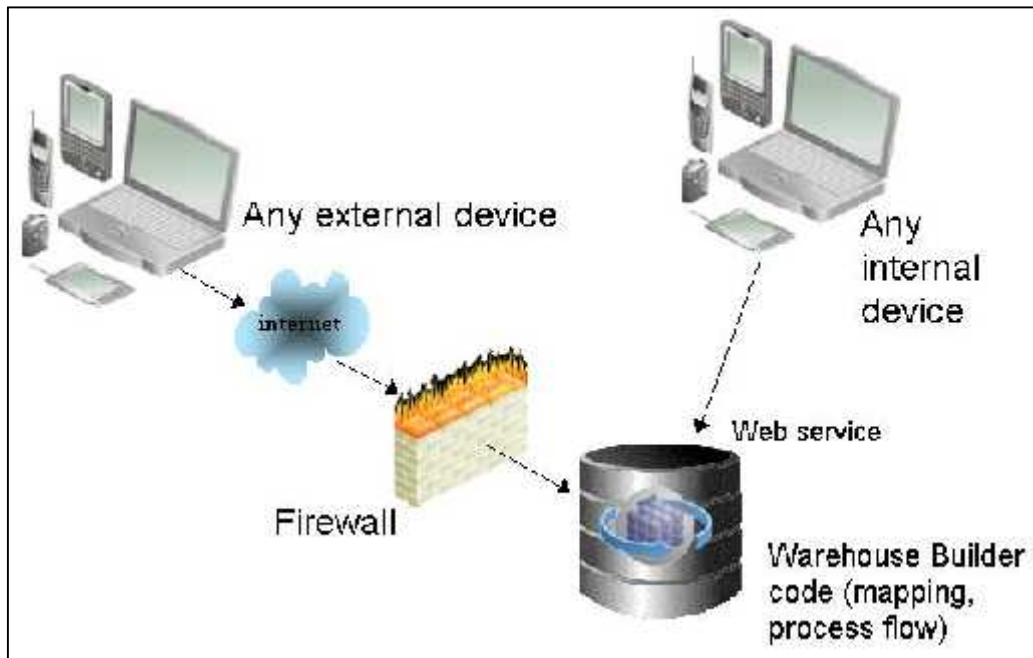
Los servicios web no son otra cosa más que el conjunto o agrupación de varios estándares y protocolos cuya finalidad es la de intercambiar datos entre distintas aplicaciones utilizando como su medio de interacción al internet, la comunicación es establecida entre diferentes plataformas, diferentes aplicaciones programadas en diferentes motores de programación.

Podemos decir que un servidor web es una herramienta o función de uso global y sin discriminación.

Dentro de las principales características que pueden ser nombradas tenemos:

- ) Facilitan el acceso a su contenido ya que utilizan tanto estándares como protocolos en formato de texto.
- ) Pueden utilizar el mismo sistema de firewall configurado para HTTP, ya que los servicios web se encuentran afirmados en este protocolo.
- ) Permiten interoperabilidad entre; aplicaciones de software independientemente de sus plataformas o propiedades, entre plataformas de distintos fabricantes a través de la utilización de estándares y protocolos abiertos.

En la figura 3-1 mostrada a continuación se encuentra establecido un escenario a modo de ejemplo de cómo se debería encontrar un servicio web y como se realiza el acceso de parte de los clientes al mismo.



**Figura 3-1.** Escenario ideal de un servicio web.  
Fuente: <http://repositorio.utn.edu.ec/bitstream/>

#### 1.4. Ataque DoS

Es uno de la variada gama de métodos que poseen los hackers para evitar o denegar básicamente el acceso de un usuario legítimo a su equipo. Los ataques consisten en el envío de muchos paquetes hacia u servidor de internet como destino que suelen ser comúnmente un servidor web, de correo o ftp. Utilizando toda esta gran cantidad de paquetes permite inundar a los recursos que posee el servidor e inutilizando completamente el sistema colocándolo en una etapa de saturación. (Barrera, 2007, pp.25-29)

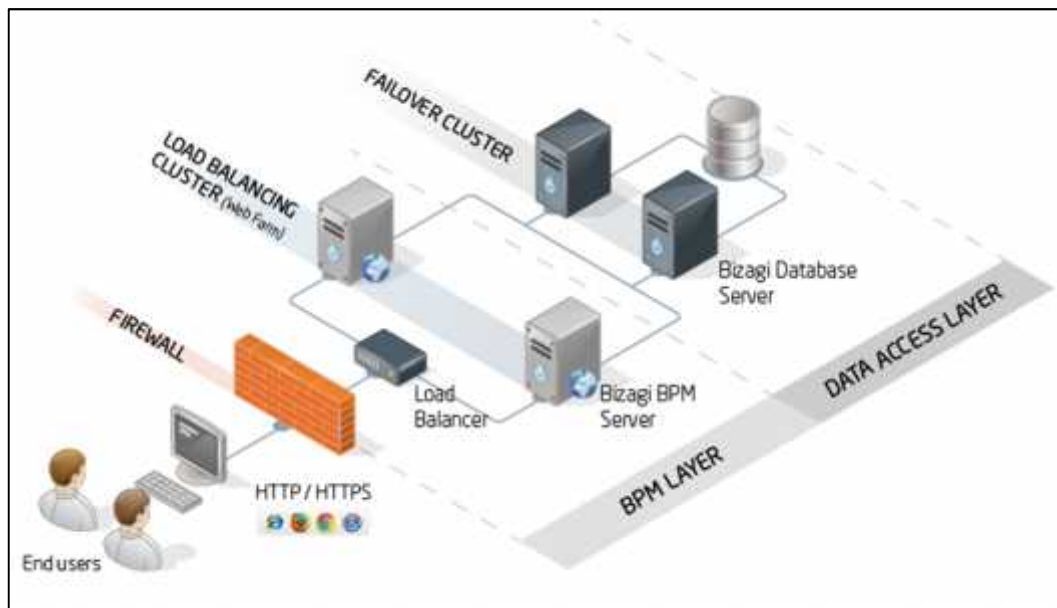
Este tipo de ataques suelen ser utilizados tanto por personas sin ética y con bajos propósitos como por los administradores de redes, puesto que es una herramienta que permite hallar el límite máximo que puede soportar un servidor en cuanto al parámetro de saturación, cabe recalcar que estas acciones son tomadas o realizadas por técnicos y sujetas a varios protocolos de seguridad con la finalidad de no comprometer la operatividad del servicio.



## 1.5. Clustering

### 1.5.1. Definición

Clúster es un grupo de equipos de cómputo que se encuentran interconectados entre sí por una red de datos de alta velocidad, estos equipos funcionan en paralelo y trabajan en conjunto comportándose como un sólo “supercomputador”. Cada equipo independiente que forma parte del clúster recibe el nombre de nodo.



**Figura 4-1.** Clúster de alta disponibilidad y balanceo de carga  
Fuente: [http://help.bizagi.com/bpmsuite/es/10.6/index.html?sysreqs\\_net\\_prod.htm](http://help.bizagi.com/bpmsuite/es/10.6/index.html?sysreqs_net_prod.htm).

La tecnología de Clustering se ha ido perfeccionando con el tiempo para satisfacer actividades que requieran poder de cómputo, servicios web que posean alta carga de tráfico, actividades de misión crítica entre las más notables. No obstante, su utilización se encuentra cada vez más generalizada y permite garantizar la confiabilidad, escalabilidad y disponibilidad de la información.

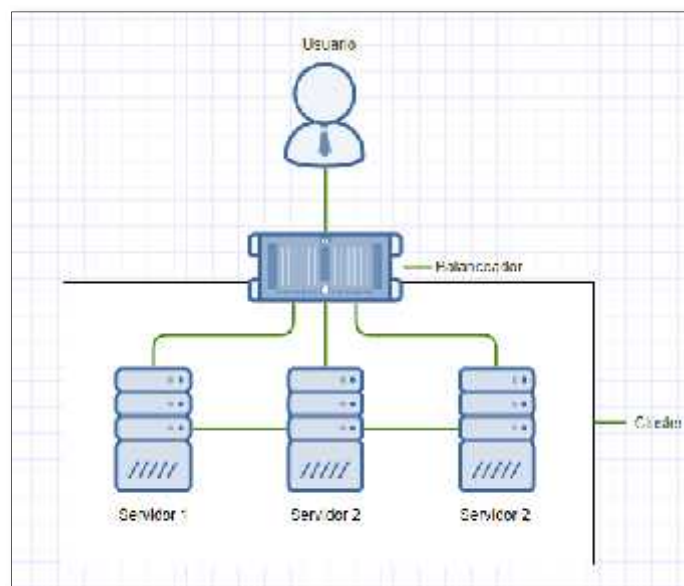
El propósito del clúster es poder integrar equipos con iguales o similares características facilitando su administración y recuperación ante posibles fallos, ya que si llegara a fallar un nodo el resto continuaría funcionando independientemente sin problema.

### 1.5.2. Arquitectura

Los clústeres disponen de características muy variadas lo cual facilita su escalabilidad y adaptación según el servicio que se requiera implementar. Sin embargo, para un funcionamiento adecuado y apropiado se requiere que el clúster esté conformado por los siguientes componentes:

- J **Nodos:** equipos o servidores interconectados en redes LAN o SAN, pueden ser de cualquier tipo.
- J **Sistemas Operativos:** deben tener características de multiusuario, preferentemente se utilizan SO. Homogéneos, pero también pueden integrarse varios tipos con particularidades propias. (Unix, Linux, Windows Server, incluso Mac OS X)
- J **Redes de datos de alta velocidad:** indispensable para la interconexión de los nodos, usualmente son de tipo: Gigabit Ethernet, Infiniband o Myrinet, entre otras.
- J **Middleware:** conocido como Single System Image (SSI), se encarga de la comunicación entre los sistemas y el usuario, dando a este la impresión de ser un único equipo con características de supercomputador.
- J **Servicios y Aplicaciones:** utilizados por los sistemas según sus requerimientos.

Cuando todos los nodos del clúster tienen las mismas características, arquitectura o recursos similares se denomina Homogéneo, por el contrario, en los sistemas Heterogéneos los nodos pueden ser distintos en diversos aspectos tales como: arquitectura, sistema operativo, rendimiento, uso eficiente de recursos, entre otros. (Indacochea, 2004, pp.230)



**Figura 5-1.** Arquitectura de un clúster  
Realizado por: Aguilar R. 2016

### 1.5.3. Clasificación

Comúnmente se puede clasificar a los clústeres por sus características propias, con respecto al trabajo que van a desempeñar. En este sentido existen 3 tipos: Alto Rendimiento, Alta disponibilidad y Balanceo de carga.

- J) **Alto Rendimiento** (HPC *High Performance Clúster*): su implementación se enfoca principalmente en aplicaciones científicas y de ingeniería donde resulta imprescindible un alto nivel de procesamiento computacional, manejo de grandes cantidades de memoria o ambas simultáneamente. (Díaz, 2015, pp.35-37)
- J) **Alta Disponibilidad** (HAC *High Availability Clúster*): este tipo de clústeres son utilizados para proveer disponibilidad y confiabilidad de la información, manteniendo activos los servicios por el mayor tiempo posible (24/7), minimizando los errores por medio de un software especializado en detección de fallas el cual permite la recuperación inmediata de los servicios, mientras que a nivel de hardware se pretende no tener un único punto de fallos. (Buyya, 199, p.999)
- J) **Balanceador de Carga** (LB *Load Balancer*): su configuración permite que el grupo de servidores implementados en el clúster compartan la carga de tráfico y trabajo proveniente de sus usuarios. Este tipo de clúster utiliza uno o más nodos que administran las peticiones hacia los servicios y las distribuyen eficientemente entre los servidores que gestionan los datos, lo que evita saturaciones del sistema y permite ampliar su capacidad fácilmente. (Cáceres, 2012, p.54)

### 1.5.4. Características

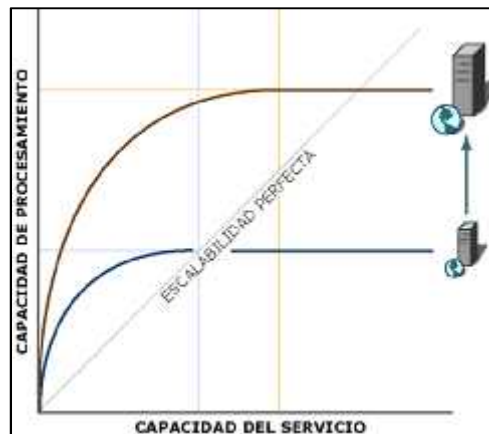
Las características notables dentro de un clúster son el escalamiento, la disponibilidad, el rendimiento y el balanceo de carga, este último atributo será abordado a mayor profundidad más adelante en el texto.

#### 1.5.4.1. Escalamiento

Es la propiedad de diseño que permite a una red la incorporación de nuevos elementos para adaptar su capacidad, nivel, rendimiento, prestaciones y demás características que conforme sean requeridas, sin la necesidad de eliminar componentes ya existentes, o alterar significativamente su topología. Podemos diferenciar dos tipos de escalabilidad:

) **Escalamiento Vertical:** también conocido como escalamiento por hardware, se basa en migrar el sistema a un equipo con mejores prestaciones (procesador, memoria), las aplicaciones, configuraciones y código fuente no sufren alteraciones mayores.

Administrativamente esto no representa un incremento de dificultad, pero si genera un impacto económico considerable, que con el progresivo avance tecnológico será una solución infructuosa, debido a que los equipos se verán limitados con los nuevos desarrollos conseguidos día a día.

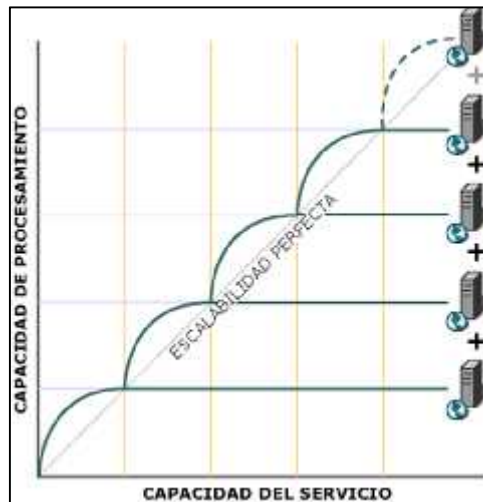


**Figura 6-1.** Escalabilidad vertical

Fuente: <https://i-msdn.sec.s-msft.com/dynimg/IC114740.gif>

) **Escalamiento Horizontal:** o escalamiento por software, consiste en incrementar el número de equipos (nodos) al clúster para distribuir las tareas de procesamiento entre todos. Aunque se utilizan varios nodos para realizar las tareas, el sistema funciona como un único equipo.

Esto requiere una mayor complejidad en la tarea administrativa, pero mejora la drásticamente la tolerancia del clúster a fallos, ya que un error en uno de sus nodos no comprometerá el funcionamiento general. (Citrix Systems, 2016)



**Figura 7-1.** Escalabilidad horizontal  
Fuente: <https://i-msdn.sec.s-msft.com/dynimg/IC47609.gif>

#### 1.5.4.2. Disponibilidad

La disponibilidad es una característica del sistema que garantiza que los datos se encontrarán siempre accesibles cuándo y cómo los usuarios los requieran. Se implementa en niveles de hardware (fuentes de poder, conexiones de red, memorias) y software (sistemas redundantes, monitoreo, heartbeat).

El sistema debe ser autosustentable, en el caso de que un nodo falle los servicios que se encuentran en este serán migrados a otro servidor y se intentará reiniciar los procesos en el equipo original. Esto solo se realizará si el clúster se encuentra configurado para distribuir las tareas entre sus nodos simultáneamente, dicho proceso debe ser transparente para el usuario y la migración resultar imperceptible para los usuarios.

Cuando el sistema tenga una estructura de master-slave (maestro-esclavo), el nodo principal o maestro ejecuta las tareas encomendadas e interactuará con los usuarios, mientras que el nodo secundario o esclavo se limitará únicamente a las tareas de respaldo, en caso de un fallo en el servidor maestro el secundario tomará su lugar hasta que el error sea corregido.

En cualquier caso, se requiere un monitoreo constante de las aplicaciones y servicios que se encuentran en ejecución. Heartbeat es el software que permite el envío constante de información, notificando entre los nodos el estado y funcionamiento de cada uno de ellos, actividad que requiere disponer de un gran ancho de banda.

Los clústeres de alta disponibilidad son muy utilizados para aplicaciones que demanden manejo de datos, sistemas bancarios, gestiones de usuarios, instituciones educativas, y demás entidades que requieran disponer de sus datos las 24 horas del día, los 365 días del año. (Pinto, 2015, pp.10-17)

#### *1.5.4.3. Rendimiento*

El rendimiento del clúster es un pilar fundamental en su funcionamiento, y, de mayor notabilidad en sistemas que requieren de una gran capacidad de: procesamiento de datos, memoria o ambos.

En el caso de los clústeres de alto rendimiento (HPPC), se focalizan en la resolución de cálculos matemáticos, renderizaciones de gráficos, compilación de programas, compresión de datos, descifrado de códigos, tareas cuyos requerimientos precisan del mejor desempeño, enfocan su labor en alcanzar grandes velocidades de procesamiento, haciendo que todos los nodos trabajen en paralelo, dividiendo el trabajo y utilizando los recursos existentes de cada nodo. (Bustos, 2007, pp.98-99)

Es indispensable en el rendimiento general del sistema, gestionar la carga de red para no entorpecer la comunicación entre procesos paralelos de los sistemas activos.

#### *1.5.5. Balanceo de carga*

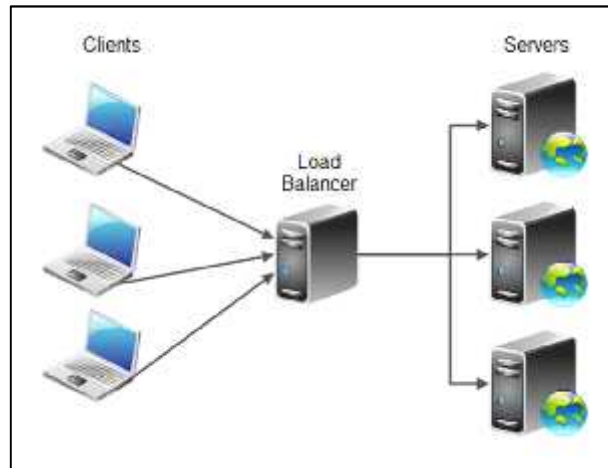
El balanceo de carga es un mecanismo de distribución de tráfico de red utilizado con el fin de alivianar la carga de recursos en un solo nodo del clúster, repartiendo las tareas entre los demás equipos.

Esta habilidad bien implementada puede garantizar la disponibilidad, incrementar la velocidad de acceso a los servicios y generar una mayor tolerancia a fallos, permite además realizar mantenimiento de los nodos en línea sin que se afecte al rendimiento general del sistema.

##### *1.5.5.1. Funcionamiento*

En esencia el balanceo de carga es el proceso de asignación de las solicitudes de una aplicación a los servidores, utilizando algoritmos sencillos como Round Robin o a su vez algunos más complejos, de tal manera de que el trabajo sea distribuido y los recursos bien aprovechados.

Existen soluciones de balanceo de carga tanto con software como con hardware dedicado, cada uno de ellos con particularidades que los distinguen y los vuelven adaptables a las necesidades de una red empresarial. Para ambas soluciones el funcionamiento es el mismo.



**Figura 8-1.** Esquema de funcionamiento balanceo de carga  
Fuente: <http://tutorials.jenkov.com/images/software-architecture/load-balancing-1.png>

#### 1.5.5.2. Métodos de conexiones

Los métodos utilizados para establecer conexiones en un clúster son:

- ) Round Robin
- ) Weighted Round Robin
- ) LeastConnection
- ) Weighted LeastConnection

#### 1.5.5.3. Balanceo de carga por hardware

Son dispositivos físicos creados únicamente con la intencionalidad de realizar el balanceo de peticiones. Las principales ventajas que proveen dichos mecanismos al proceso de balanceo son la escalabilidad, potencia y estabilidad, sin embargo, presentan podemos encontrar también algunas dificultades, como; el valor de adquisición que poseen, necesitan de un mantenimiento más especializado y por nombrarlo de cierta manera una subutilización debido a su exclusividad para tareas de balanceo de carga. Es necesario recalcar que varias de las grandes empresas a nivel internacional han optado por esta solución para contrarrestar sobrecargas en sus servidores,

obviamente son empresas que tienen la capacidad de financiar la compra de uno de estos equipos pero también conocen de los grandes beneficios que traen consigo.

Entre las soluciones que podemos mencionar tenemos a: CitrixNetscaler, WebMux Load Balancer, Barracuda Load Balancer, CISCO Arrowpoint, entre otras.

### ) **Citrix NetScaler**

Por mucho el hardware ADC (Application Delivery Controller) con mejores prestaciones que se puede encontrar en el mercado a la fecha. Es amplia y continua la creciente demanda de dispositivos de equilibrio de carga, estos dispositivos fabricados a “medida” son la navaja suiza de las redes informáticas, mejoran el rendimiento de las aplicaciones, incrementa las seguridades, realiza el balanceo de cargas desde la capa 2 hasta la 7 del modelo OSI, compresión http, aceleración con SSL, cuenta con soporte constante y actualizado, almacena contenido en cache, proporciona la flexibilidad que las aplicaciones demandan sobre la web, brinda la sostenibilidad de los servicios, incrementa las velocidades de acceso a los mismo, permite la posibilidad de dar mantenimiento a un servidor sin alterar su servicio y sobre todo hoy por hoy se ha convertido en un servicio más. (Chuquiguanca, 2015, pp.3-7)

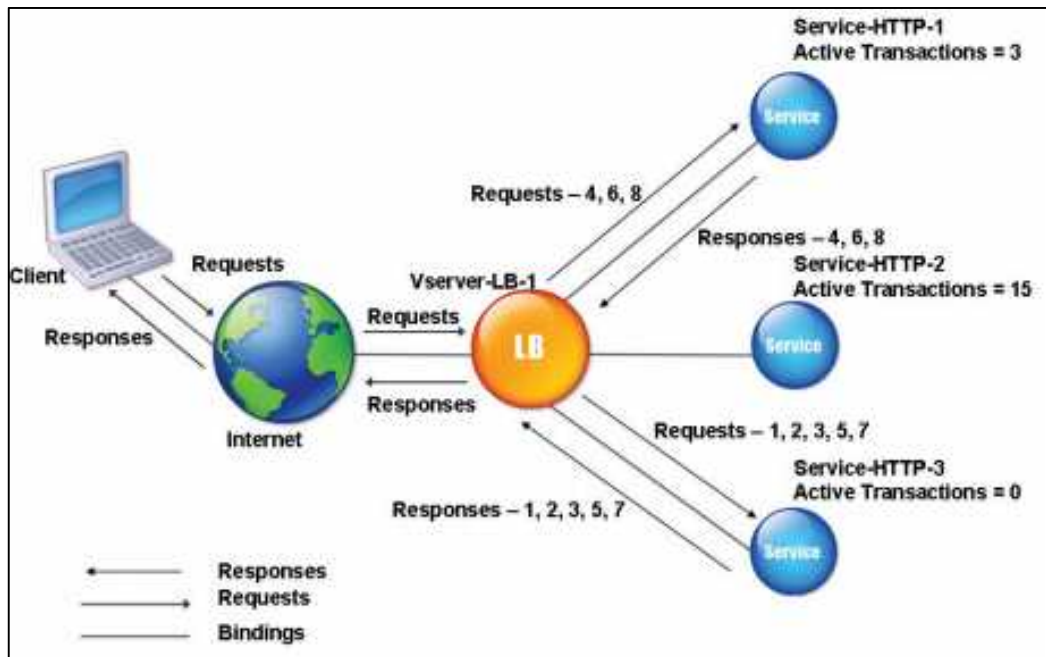
Este balanceador de carga ADC posee una interfaz de administración muy intuitiva y de sencillo uso, ofrece una variedad de métodos de conexión; desde RoundRobin hasta DomainHash o Token, siendo muy versátil a la hora de configurarlo, se encuentra perfectamente compatibilizado con redes CISCO, convirtiéndose en una importante ventaja al momento de considerar soluciones.

#### *1.5.5.4. Algoritmos de balanceo de carga que oferta Citrix Netscaler*

Dentro de las principales fortalezas de Citrix Netscaler es su adaptabilidad y versatilidad, en este punto serán mencionados algunos de los variados métodos o algoritmos que ofrece este potente hardware.

- ✓ **Least Connection.-** Cuando es seleccionado este método, Citrix selecciona el servicio con la menor cantidad de conexiones activas para asegurarse la carga de las solicitudes activas se encuentren en equilibrio sobre los servicios. Se encuentra establecido como el método por defecto.
  - Cuando se asignan pesos a los procesos Netscaler selecciona los servicios utilizando el valor (Nw) de la siguiente expresión:
    - $Nw = (\text{Número de transacciones activas}) * (10000/\text{peso})$
  - De esta manera el dispositivo de balanceo realiza el equilibrio en la carga.





**Figura 9-1.** Método de menor conexión (Least Connection)  
Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

En la figura 9-1 se puede apreciar dentro de un escenario el mecanismo que utiliza NetScaler cuando se encuentra configurado bajo el algoritmo de menor conexión o Least Connection.

N= Es el número de transacciones activas

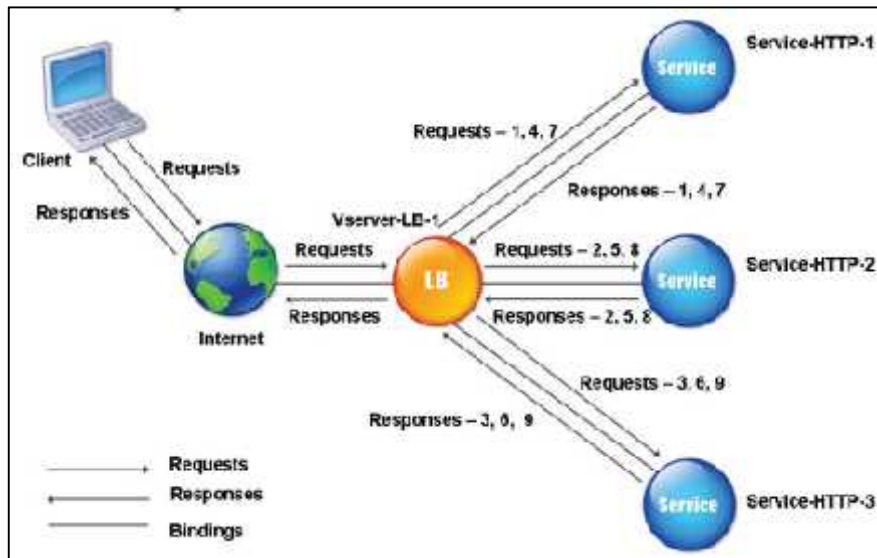
Las peticiones son entregadas como se observa en la gráfica (figura 8-1), en la siguiente tabla podemos ver de manera detallada la aparición del número de transacciones.

**Tabla 1-1. Ejemplo de transacciones método least connection Citrix Netscaler.**

Request received	Service selected	Current $N$ ( <i>Number of active transaction</i> ) value	Remarks
Request-1	Service-HTTP-3 ( $N = 0$ )	$N = 1$	Service-HTTP-3 has the least $N$ value.
Request-2	Service-HTTP-3 ( $N = 1$ )	$N = 2$	
Request-3	Service-HTTP-3 ( $N = 2$ )	$N = 3$	
Request-4	Service-HTTP-1 ( $N = 3$ )	$N = 4$	Service-HTTP-1 and Service-HTTP-3 have the same $N$ values.
Request-5	Service-HTTP-3 ( $N = 3$ )	$N = 4$	
Request-6	Service-HTTP-1 ( $N = 4$ )	$N = 5$	
Request-7	Service-HTTP-3 ( $N = 4$ )	$N = 5$	
Request-8	Service-HTTP-1 ( $N = 5$ )	$N = 6$	
Service-HTTP-2 is selected for load balancing when it completes the active transactions or when the $N$ value of other services (Service-HTTP-1 and Service-HTTP-3) is equal to 15.			

Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

- ✓ **Round Robin.**- El más conocido en forma general de todos los métodos. Netscaler circula las solicitudes entrantes a los servidores administrados independientemente de la carga, es decir, la primera solicitud se envía al Servicio-HTTP-1, la segunda al Servicio-HTTP-2, la tercera al Servicio-HTTP-3 y así sucesivamente. Cuando todas las solicitudes fueron enviadas se reinicia el ciclo.



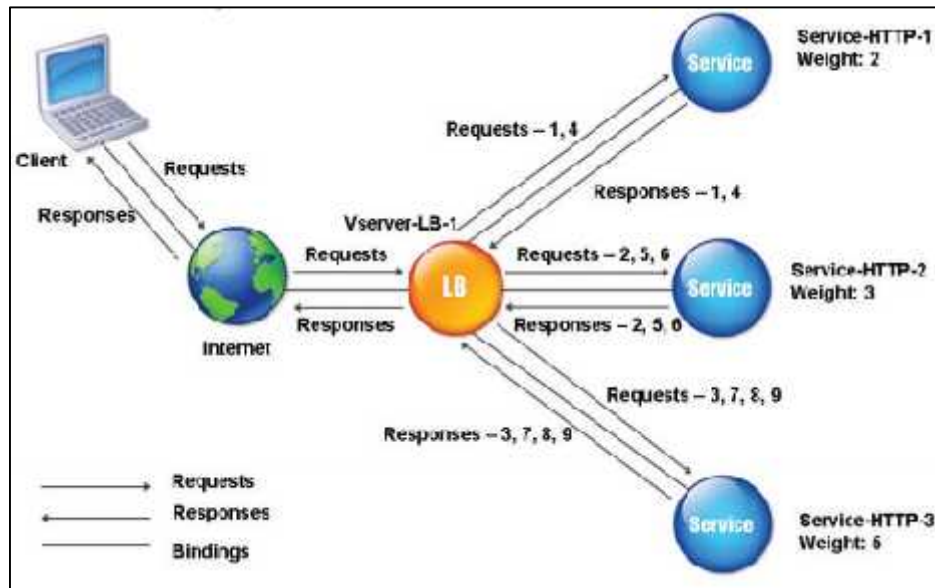
**Figura 10-1. Método Round Robin**

Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

La figura 10-1 idealiza un esquema de peticiones y respuestas a través del mecanismo de conexión round robin con tres servidores. Podemos mencionar el siguiente ejemplo para comprender un poco más cómo se comporta Citrix a través de este método. El servidor HTTP-1 es configurado con un peso 2, el servidor HTTP-2 recibe un peso 3 y el servidor HTTP-3 recibe un peso 4 y todos ellos se encuentran enlazados con el servidor de balanceo de carga, entonces las peticiones son entregadas de la siguiente manera:

- ) Servidor-HTTP-1 recibe la primera petición.
- ) Servidor-HTTP-2 recibe la segunda petición.
- ) Servidor-HTTP-3 recibe la tercera petición.
- ) Servidor-HTTP-1 recibe la cuarta petición.
- ) Servidor-HTTP-2 recibe la quinta petición.
- ) Servidor-HTTP-3 recibe la sexta petición.
- ) Servidor-HTTP-2 recibe la séptima petición.
- ) Servidor-HTTP-3 recibe la octava y novena petición.

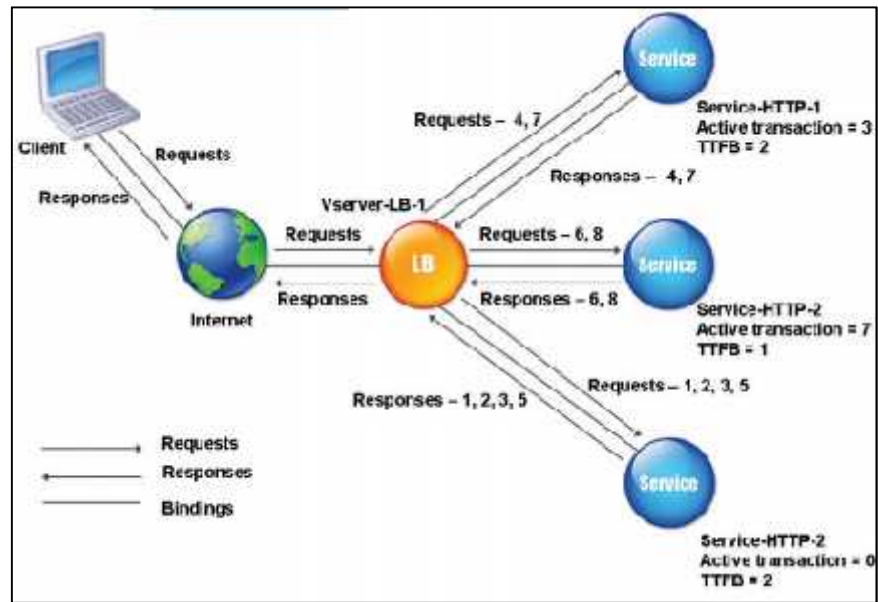
Luego de ello el ciclo vuelve a empezar usando el mismo patrón. El siguiente diagrama (figura 10-1) nos ilustra sobre el método Round Robin utilizando pesos.



**Figura 11-1.** Método Round Robin con pesos  
Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

- ✓ **Least Response Time.-** Método del menor tiempo de respuesta, Citrix además del servicio con menor cantidad de conexiones selecciona al menor tiempo promedio de respuesta. Permite ser configurado únicamente para servicios HTTP y SSL. Aquí es donde aparece por primera vez el término TTFB Time to First Byte que no es otra cosa que el intervalo de tiempo desde la petición del servicio hasta recibir el primer byte de respuesta. NetScaler utiliza el código de respuesta 200 para calcular el TTFB. En el siguiente ejemplo podemos ver como se desenvuelve el método de balanceo de menor respuesta utilizando tres servidores.
  - ) El servidor HTTP-1 está manejando tres transacciones activas y el TTFB es de dos segundos.
  - ) El servidor HTTP-2 está manejando siete transacciones activas y el TTFB es de un segundo.
  - ) El servidor HTTP-3 no está manejando ninguna transacción activa y el TTFB es de dos segundos.

A continuación, la figura 11-1 muestra como NetScaler utiliza el método de menor tiempo de respuesta y envía las solicitudes a los tres servidores.



**Figura 12-1.** Método de menor tiempo de respuesta (Least response time)  
 Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

En éste método  $N$  ya no es únicamente el número de transacciones activas, ahora es el producto de estas con el TTFB

$$N = \text{Número de transacciones} * TTFB$$

La entrega de peticiones se realiza de la siguiente manera:

- J El servidor HTTP-3 recibe la primera petición porque el servidor no se encuentra manejando ninguna transacción activa. Cuando un servidor no se encuentra manejando transacciones es elegido siempre primero.
- J El servidor HTTP-3 recibe la segunda y tercera petición porque el servidor posee el menor valor  $N$ .
- J EL servidor HTTP-1 recibe la cuarta petición porque tanto el servidor HTTP-1 y HTTP-3 poseen el mismo valor  $N$ , Citrix ejecuta el balanceo de carga en la manera de un Round Robin. Por lo tanto el servidor HTTP-3 recibe la quinta petición.
- J El servidor HTTP-2 recibe la sexta petición porque el servidor ahora tiene el menor valor  $N$ .
- J El servidor HTTP-1 recibe la séptima petición porque el servidor HTTP-1, HTTP-2, y HTTP-3 tienen el mismo valor  $N$  y realiza nuevamente el balanceo de forma de Round Robin, por lo tanto el servidor HTTP-2 recibe la octava petición.

En la siguiente tabla (tabla 2-1)

**Tabla 2-1. Ejemplo de transacciones método least response time.**

Request received	Service selected	Current <i>N</i> ( <i>Number of active transaction * TTFB</i> ) value	Remarks
Request-1	Service-HTTP-3 ( <i>N</i> = 0)	<i>N</i> = 2	Service-HTTP-3 has the least <i>N</i> value.
Request-2	Service-HTTP-3 ( <i>N</i> = 2)	<i>N</i> = 4	
Request-3	Service-HTTP-3 ( <i>N</i> = 3)	<i>N</i> = 6	
Request-4	Service-HTTP-1 ( <i>N</i> = 6)	<i>N</i> = 8	Service-HTTP-1 and Service-HTTP-3 have the same <i>N</i> values.
Request-5	Service-HTTP-3 ( <i>N</i> = 6)	<i>N</i> = 8	
Request-6	Service-HTTP-2 ( <i>N</i> = 7)	<i>N</i> = 8	Service-HTTP-2 has the least <i>N</i> value.
Request-7	Service-HTTP-1 ( <i>N</i> = 8)	<i>N</i> = 15	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same <i>N</i> values.
Request-8	Service-HTTP-2 ( <i>N</i> = 8)	<i>N</i> = 9	

Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

- ✓ **Using Weights with the Least Response Time Method.-** Como su nombre lo indica este proceso implica el uso de un cálculo realizado a través del número de conexiones, TTFB por sus siglas en inglés Time to First Byte que es el intervalo de tiempo del envío de una petición al servicio y la recepción del primer paquete de la respuesta, y los pesos si son asignados de forma distinta entre los servidores. Se calcula con la siguiente fórmula.  $Nw = (N) * (10000 / \text{weight})$
- ✓ **Least Response Time Method with Monitors.-** En esta variante del método least response tenemos que cuando NetScaler posee esta configuración éste selecciona el servicio con el último número de transacciones activas y tiempo promedio de rapidez de respuesta del monitor. Con este método Citrix utiliza toda la infraestructura de monitoreo.

En la siguiente tabla (tabla 3-1) se muestra un resumen de cómo son calculados los tiempos de respuesta de varios monitores.

**Tabla 3-1. Resumen cálculo de tiempos de respuesta de los distintos monitores.**

<b>Monitor</b>	<b>Response time calculation</b>
PING	Time difference between the ICMP ECHO request and the ICMP ECHO response.
TCP	Time difference between the SYN request and the SYN+ACK response.
HTTP	Time difference between the HTTP request (after the TCP connection is established) and the HTTP response.
TCP-ENV	Time difference between the time the data send string is sent and the data receive string is returned. A tcp-ecv monitor without the send and receive strings is considered to have an incorrect configuration.
HTTP-ECV	Time difference between the HTTP request and the HTTP response.
UDP-ECV	Time difference between the UDP send string and the UDP receive string. A udp-ecv monitor without the receive string is considered to have an incorrect configuration.
DNS	Time difference between a DNS query and the DNS response.
TCPS	Time difference between a SYN request and the SSL handshake completion.

<b>Monitor</b>	<b>Response time calculation</b>
FTP	Time difference between the sending of the user name and the completion of user authentication.
HTTPS (monitors HTTPS requests)	Time difference is same as the HTTP monitor.
HTTPS-ENV (monitors HTTPS requests)	Time difference is same as the HTTP-ECV monitor.
USER	Time difference between the time when a request is sent to the dispatcher and the time when the dispatcher responds.

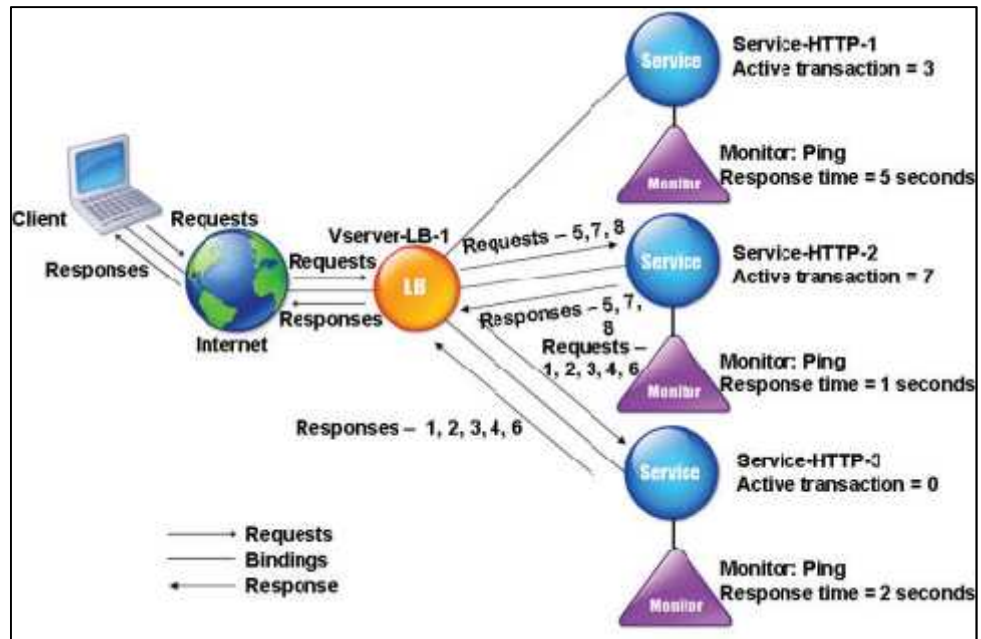
**Fuente:** [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

Veamos en el siguiente ejemplo cómo NetScaler selecciona el servidor a través del método de menor tiempo de respuesta con sus monitores configurados.

- J El servidor HTTP-1 está manejando tres transacciones activas y el tiempo de respuesta es de cinco segundos.
- J El servidor HTTP-2 está manejando siete transacciones activas y el tiempo de respuesta es de un segundo.

- El servidor HTTP-3 no está manejando ninguna transacción activa y el tiempo de respuesta es de dos segundos.

La figura 13-1 presenta el diagrama del funcionamiento de Citrix al usar el método de menor tiempo de respuesta a través de sus monitores.



**Figura 13-1.** Método de menor tiempo de respuesta (Least response time Monitors)  
Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

En la siguiente expresión se indica cuáles son los parámetros utilizados para realizar el cálculo del valor  $N$ .

$$N = \text{Número de transacciones activas} * \text{tiempo de respuesta que es determinado por el monitor.}$$

La entrega de las peticiones es ejecutada de la siguiente manera:

- El servidor HTTP-3 recibe la primera petición porque el servidor no se encuentra manejando transacciones.
- El servidor HTTP-3 recibe la segunda, tercera y cuarta petición porque éste tiene el menor valor  $N$ .
- El servidor HTTP-2 recibe la quinta petición porque el servidor tiene el menor valor  $N$ .



- J) Ahora ambos servidores HTTP-2 y HTTP-3 tienen el mismo valor  $N$ , así que NetScaler ejecuta el balanceo de carga como lo hace Round Robin. De esta manera la seta petición es recibida por el servidor HTTP-3.
- J) El servidor HTTP-2 recibe la séptima y octava petición porque ahora este servidor tiene el menor valor  $N$ .

El servidor HTTP-1 no es considerado para el balanceo de carga porque su valor  $N$  es mayor con respecto a los de los otros servidores. Sin embargo si el servidor HTTP-1 completa las transacciones activas NetScaler lo considerará para el balanceo de carga.

Se muestra en la tabla de resumen (tabla 4-1) la forma en la que las peticiones son recibidas por los servidores utilizando el valor  $N$ .

**Tabla 4-1. Resumen cálculo de tiempos de respuesta de los distintos monitores.**

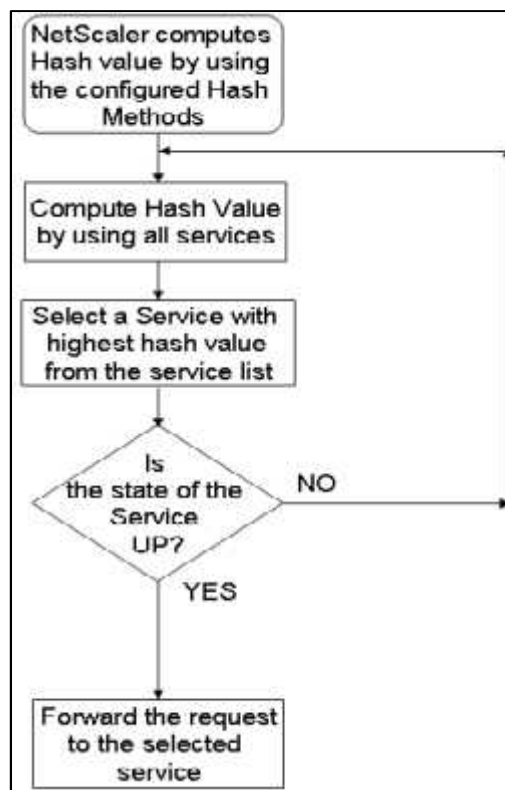
Request received	Service selected	Current N (Number of active transaction) value	Remarks
Request-1	Service-HTTP-3 ( $N = 0$ )	$N = 2$	Service-HTTP-3 has the least $N$ value.
Request-2	Service-HTTP-3 ( $N = 2$ )	$N = 4$	
Request-3	Service-HTTP-3 ( $N = 4$ )	$N = 6$	
Request-4	Service-HTTP-3 ( $N = 6$ )	$N = 8$	
Request-5	Service-HTTP-2 ( $N = 7$ )	$N = 8$	Service-HTTP-1 and Service-HTTP-3 have the same $N$ values.
Request-6	Service-HTTP-3 ( $N = 8$ )	$N = 10$	
Request-7	Service-HTTP-2 ( $N = 8$ )	$N = 9$	Service-HTTP-2 has the least $N$ value.
Request-8	Service-HTTP-1 ( $N = 9$ )	$N = 10$	
Service-HTTP-1 is selected for load balancing when it completes the active transactions or when the $N$ value of other services (Service-HTTP-2 and Service-HTTP-3) is equal to 15.			

Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

- ✓ **Using Weights with the Least Response Time.-** NetScaler siempre mejora su rendimiento utilizando el número de transacciones activas, el tiempo de respuesta y los pesos.

\* Los métodos hash son utilizados en escenarios donde el rango de contenido a ser cacheado es amplio. El cacheo de peticiones reduce considerablemente la latencia de petición y respuesta mejorando el uso de los recursos (CPU).

La figura 14-1 ilustra el diagrama de flujo que utilizan en esencia los métodos hash.

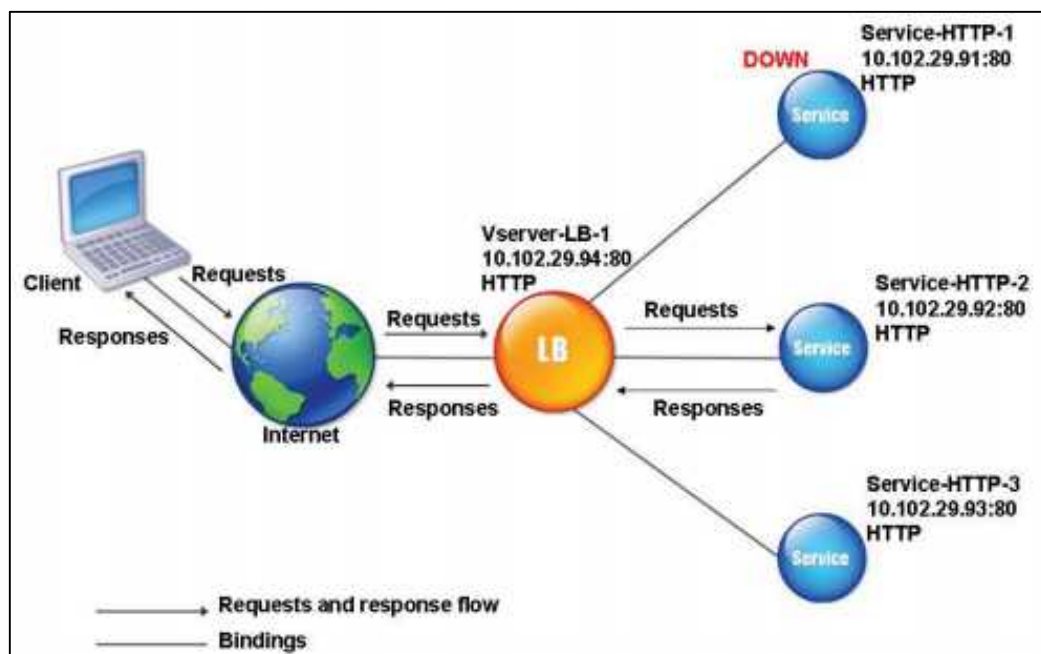


**Figura 14-1.** Diagrama de flujo del trabajo de los métodos hash  
Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

NetScaler calcula el valor Hash a través de todos los servidores. Luego se selecciona un servidor con el mayor valor Hash desde la lista de servidores. Se llega al cuestionamiento; ¿Se encuentra el estado del servidor en alto (UP)?, si la respuesta es no entonces se regresa al cálculo del valor hash, si la respuesta es afirmativa entonces se continúa al envío de la petición al servidor seleccionado.

- ✓ **URL hash.-** Cuando este método es usado Citrix selecciona el servidor basado en el valor hash que obtiene de la URL HTTP, dicho valor es obtenido a través del uso del hit de cacheo. Por defecto Netscaler calcula el valor utilizando los primeros 80 bytes de la URL. Es posible la configuración del parámetro de longitud del valor Hash para calcular un valor URL diferente. Si NetScaler no puede analizar con precisión la petición entrante, éste utiliza el método Round Robin para el balanceo de carga.

Consideremos un escenario con tres servidores enlazados a un servidor virtual y el método configurado es el URL Hash. Los servidores son: servidor HTTP-1, servidor HTTP-2 y servidor HTTP-3, y el valor hash es URL1. Cuando los servidores están levantados (UP), URL1 es enviado al servidor HTTP-1 utilizando el resultado hashing. Si el servidor HTTP-1 se encuentra caído (DOWN), el URL1 es enviado al servidor HTTP-2 utilizando el resultado Hash secundario como se indica en el siguiente diagrama (figura 14-1).



**Figura 15-1.** Método URL hash

Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

Si el servidor HTTP-1 y el servidor HTTP-2 se encuentran DOWN. URL1 es enviado al servidor HTTP-3. Si los servidores se encuentran UP entonces URL1 se envía en las siguientes maneras:

- ⌋ Si el servidor HTTP-2 está UP, el URL1 se envía al servidor HTTP-2.
- ⌋ Si el servidor HTTP-1 está UP, el URL1 se envía al servidor HTTP-1.

) Si el servidor HTTP-1 y HTTP-2 están UP al mismo tiempo entonces URL1 es enviado al servidor HTTP-1.

Para configura el método URL Hash, se usa el parámetro de longitud del valor Hash como se describe en la tabla 5-1.

**Tabla 5-1. Resumen cálculo de tiempos de respuesta de los distintos monitores.**

Parameter	Specifies
Hash Length (hashLength)	The number of bytes that are hashed in the URL or domain name. Valid values are from 1 to 4K bytes. The default is 80 bytes. It must not exceed 4096 bytes.

Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

- ✓ **Domain Hash.-** Esta vez el valor hash es obtenido del nombre del dominio encontrado en la solicitud HTTP, específicamente de la cabecera de la petición. En caso de no encontrarse el nombre del dominio automáticamente Citrix utiliza el método Round Robin para contestar estas solicitudes.
- ✓ **Destination IP Hash.-** Cuando Citrix se configura para utilizar este método, éste selecciona el servicio basando su decisión el valor hash de la dirección IP de destino. Para calcular el valor hash se puede utilizar el parámetro de la máscara de la subred para enmascarar la dirección IP de destino. Cuando las peticiones llegan a NetScaler se identifican la subred utilizando la máscara y se envía la respuesta a través del mismo servidor usando el valor hash.  
Éste método es apropiado para ser utilizado con la característica de redireccionamiento de cacheo de NetScaler. Para configurar este método se utiliza el parámetro de máscara de red descrito en la tabla 6-1.

**Tabla 6-1. Parámetro de máscara de red.**

Parameter	Specifies
Netmask (netmask)	Mask that the destination IP address before calculating the hash value so that all IP addresses belonging to a particular network are directed to the same server.

Fuente: [https://kb.wisc.edu/images/group9/13201/NS-Load\\_Balance\\_Methods.pdf](https://kb.wisc.edu/images/group9/13201/NS-Load_Balance_Methods.pdf)

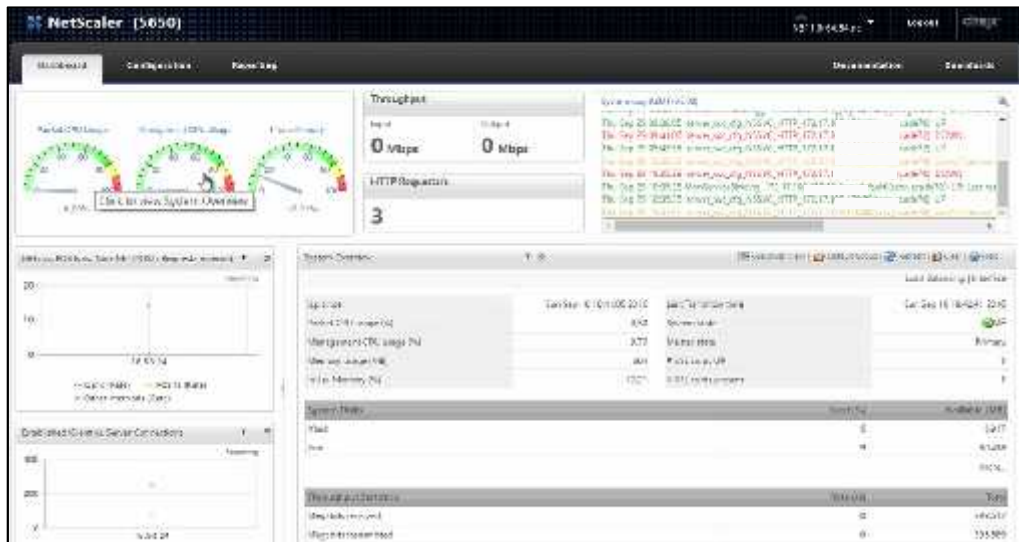
- ✓ **Source IP hash.-** La variación en este método es que el valor hash es obtenido de la dirección IP de origen, con dicho valor escoge el servidor. El método que incluye hashing es simétrico por lo tanto se obtiene el mismo valor si las direcciones IP de origen y destino se invierten. Esto asegura que todas las peticiones lleguen al mismo servidor.
- ✓ **Token.-** Citrix selecciona el servicio basándose en un token extraído desde la petición del cliente. Se puede editar tanto el lugar y el tamaño de este token, consecuentemente las peticiones con el mismo token son enviadas al mismo servidor que atendió la primera solicitud. Sin embargo ellos toman al servicio por fuera de la decisión del balanceo.



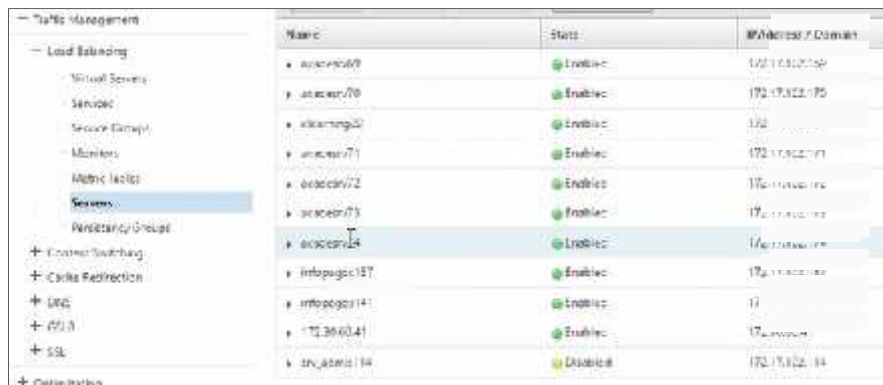
**Figura 16-1.** Logo Citrix NetScaler  
Fuente: <https://lac.citrix.com/>

Citrix NetScaler proporciona el uso de herramientas a través de una interfaz gráfica intuitiva y muy sencilla desarrollada sobre la web; dentro de las principales opciones que podemos mencionar tenemos: Dashboard, dentro del cual vamos a encontrar paneles y gráficas de rendimiento del equipo Citrix (figura 16-1), las sesiones concretadas, entre otras, la siguiente pantalla Configuration, en este apartado vamos a observar todas las opciones de configuración que ofrece Citrix, entre ellas la más destacada y que será usada en nuestros procesos es la administración del tráfico con su variante balanceo de carga, en ella podemos ver cada servidor con su IP asignada y el estado del mismo, si se encuentra habilitado o caído (figura 17-1), además se tiene una vista completa de los servicios y de los servidores virtuales con sus respectivos algoritmos o métodos de balanceo configurados (figura 18-1), creando una alta gama de posibilidades frente a la administración de este núcleo. (Citrix Systems, 2016)

Y finalmente podemos mencionar el apartado de reporte (Reporting), en el cual hallaremos toda la información estadística sobre nuestra red, servidor, rendimiento, conexiones, y demás. (figura 19-1) (figura 20-1)



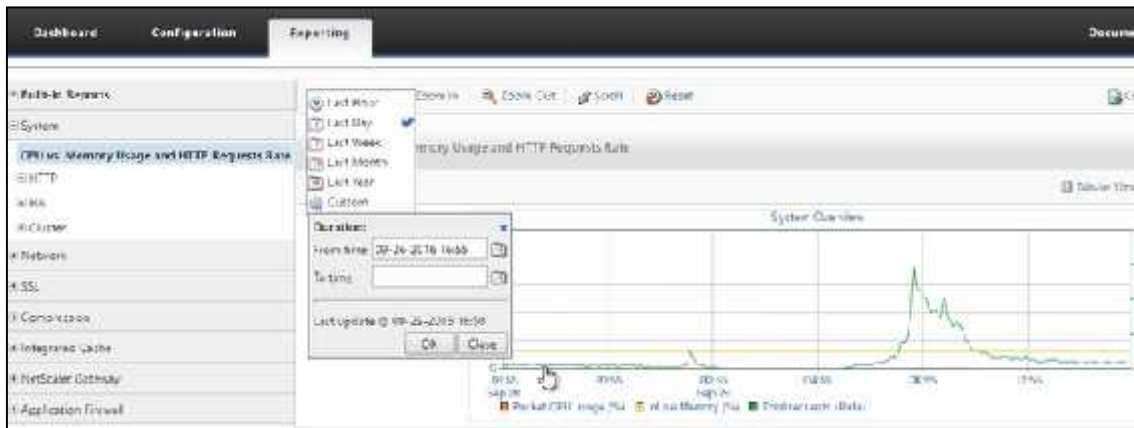
**Figura 17-1. Dashboard**  
Realizado por: Aguilar R. 2016



**Figura 18-1. Configuration/Traffic Management/Load Balancing/Servers**  
Realizado por: Aguilar R. 2016



**Figura 19-1. Configuration/Traffic Management/Load Balancing/Virtual Servers**  
Realizado por: Aguilar R. 2016



**Figura 20-1.** Reporting/System/CPUvsMemory Usage and HTTP Requests Rate  
Realizado por: Aguilar R. 2016

### ) **WebMux Load Balancer**

Este dispositivo presenta los atributos que caracterizan a un ADC, provee balanceo de carga desde capa 2 hasta capa 7 del modelo OSI, alto rendimiento, compresión de HTTP, ensancha la seguridad, es escalable y flexible, su desventaja principal es que sus equipos son desactualizados y no cuentan con el soporte requerido, son de bajo costo de acuerdo al mercado.



**Figura 21-1.** Logo AVANU WebMux  
Fuente: [www.cainetworks.com](http://www.cainetworks.com)

### ) **Barracuda Load Balancer**

Estos dispositivos presentan un gran desempeño, se encuentran certificados por Microsoft contando con un gran respaldo, son escalables, de alto rendimiento, flexibles al entorno de la red, aceleración con SSL, robustecimiento de la seguridad, cacheo de contenido HTTP, compresión de datos, actualizaciones, su principal desventaja es que en el país se dificulta la obtención de soporte.



**Figura 22-1.** Logo Barracuda  
Fuente: <https://www.barracuda.com>

#### *1.5.5.5. Balanceo de carga por software*

Son balanceadores configurados en servidores, versátiles, presentan una considerable reducción de costos frente al hardware de balanceo, puede ser instalado en servidores reciclados y realizar otra tarea a la vez que realiza la de balanceo, las desventajas que tenemos en este apartado son: menor potencia y mayor tiempo en la ejecución de las tareas de mantenimiento del mismo.

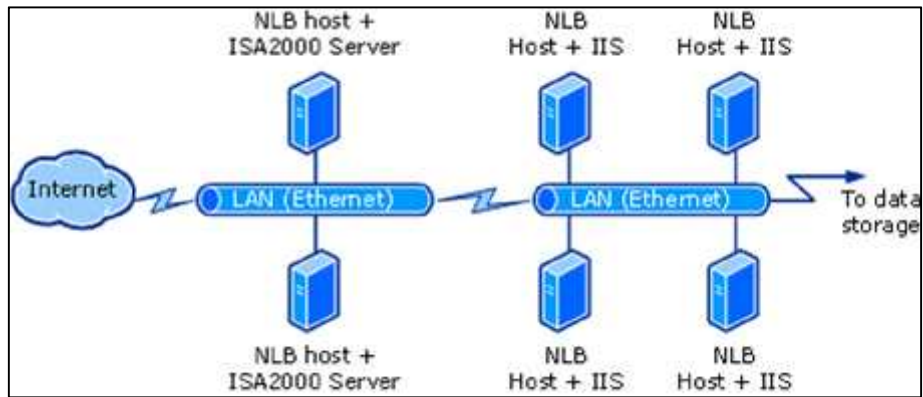
Podemos citar entre estas alternativas a; Linux Virtual Server, Perlbal, Pound, entre otras.

#### **) Network Load Balancing**

Conocido por sus siglas NLB, es una de las características con las que viene implementado el sistema operativo de administración de servidores propiedad de Microsoft “Windows Server 2008 R2”. Presenta características de escalabilidad y disponibilidad para los clientes, usado frente a servidores Web, Ftp, VPN, proxy, firewall. Puede ser configurado sobre un ordenador común, además ofrece la posibilidad de incluirlo dentro un clúster virtual combinado con varios servidores, brinda a los servidores de misión crítica fiabilidad y mejora su rendimiento.

NLB permite que todos los servidores tengan la misma dirección IP que se configuró para el clúster, distribuye las peticiones de los clientes y el tamaño de la carga que manejará cada host del clúster puede ser establecido acorde a las necesidades.





**Figura 23-1.** Posible configuración con NLB

Fuente: <https://technet.microsoft.com/>

Cuando un host del clúster falla o queda fuera de línea inmediatamente la carga es redistribuida sobre los servidores que funcionan correctamente, todo este proceso que realiza NLB es ejecutado de forma transparente al cliente. NLB emplea notificaciones de intercambio Heartbeat para mantener un informe detallado y controlado de cada situación en los servidores. Cuando un servidor no envía una notificación dentro de cinco segundos el resto de servidores realizan lo siguiente:

- a. Establecer que servidores se encuentran activos dentro del clúster.
- b. Elegir al servidor con más alta prioridad como el nuevo host por defecto.
- c. Asegurarse de que todas las peticiones de los clientes hayan sido manejadas por los hosts activos.

Las principales características que podemos mencionar son:

- ✓ Soporta la conexión de hasta 32 computadores en un clúster sencillo.
- ✓ Realiza el balance de varias solicitudes al servidor ya sea desde el mismo cliente o de varios clientes.
- ✓ Posee la capacidad de agregar nuevos hosts a medida que la carga aumente.
- ✓ Así mismo posee la capacidad de retirar hosts cuando la carga disminuye.
- ✓ Posee un alto rendimiento y un bajo costo operativo.
- ✓ Es fácil de usar. (Flavio, 2011, pp.3-10)

#### 1.5.5.6. Opciones de afinidad múltiple host

La afinidad de un clúster hace referencia a la manera en la que se realiza el balanceo de carga entre los nodos. Esta no se considera una propiedad sino más bien es denominada una regla de puertos del clúster en la modalidad de filtrado múltiple host (modelo habitual)

Nuestro clúster puede recibir varios tipos de solicitudes, en base a ellas se debe realizar la elección de la afinidad a ser configurada, entonces resulta imprescindible conocer si las sesiones se establecerán o no a través de múltiples puertos, si los clientes de la red pasan a través de algún arreglo de proxys, saber si realizan NAT o utilizan varias direcciones externas, si los clientes pueden realizar su acceso a través de diferentes direcciones, es decir, si poseen activo su DHCP, en el caso de aplicaciones WEB saber si es necesario mantener una sesión activa o esta es capaz de mantener información compartida entre todos los nodos.

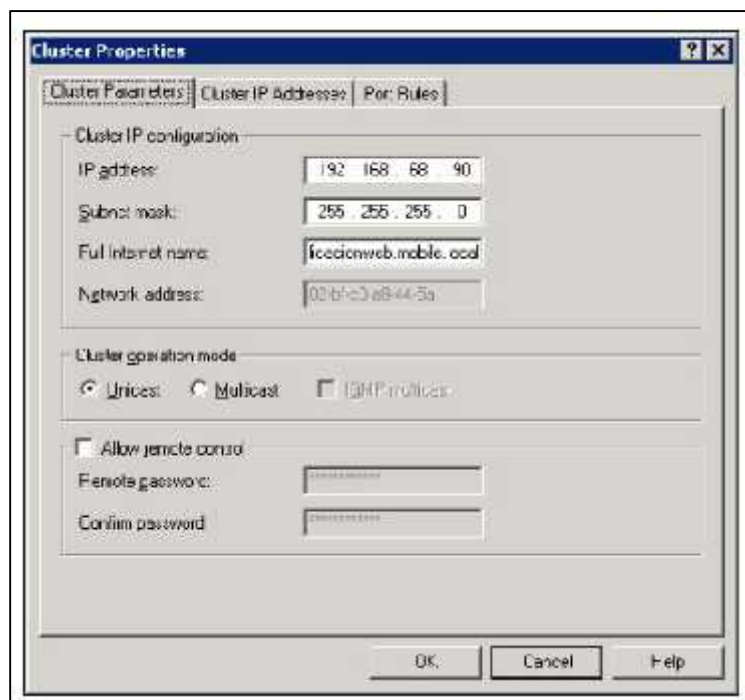
NLB proporciona tres opciones que pueden ser utilizadas (NONE, SINGLE y CLASS C) a la hora de configurar la afinidad múltiple para nuestros servidores.

- J) None; se realizará el balanceo de carga en función de la dirección IP del cliente junto con el puerto (TCP o UDP) de la conexión considerada entrante. Comúnmente es elegida cuando las aplicaciones no pueden mantener su estado, entonces en estos casos es necesario que cualquier host sea capaz de resolver las peticiones. El rebalanceo realizado con esta afinidad permite el uso eficiente del clúster en comparación con los otros modos, la dificultad es que no todas las aplicaciones pueden soportar este procedimiento.
- J) Single; a diferencia que el anterior método sólo se realiza el balanceo en función de la dirección IP del cliente, dando lugar al redireccionamiento de todas las peticiones con la misma dirección IP al mismo servidor. Esta configuración es recomendada cuando los servicios que serán ofrecidos utilizan sesiones con conexión a través de múltiples puertos como es el caso de servicios de conexión VPN o sesiones que requieren conexiones SSL utilizadas por HTTPS.
- J) Si nos encontramos en el caso de que nuestros clientes pasan a través de un único proxy haciendo NAT todas las peticiones provenientes de ellos serán enviadas al mismo servidor lo que puede provocar una sobrecarga implicando que el balanceo de carga no sea distribuido correctamente en la red funcionando sin el precepto de ser equitativo. Es preferible en el caso de que nuestros clientes envíen sus solicitudes a través de un arreglo de proxys utilizar el método CLASS C, porque la configuración single puede resultar errada enviando las peticiones a distintos servidores.

- J Class C; aquí el balanceo es realizado en función de los 24 bits más significativos de la dirección IP del cliente, haciendo el redireccionamiento de las peticiones que poseen la misma subred al mismo nodo del clúster, de esta manera en el caso de que los clientes accedan desde un arreglo de proxys con varias direcciones externas serán enviadas al mismo servidor.

#### 1.5.5.7. Modo de operación del clúster NLB

La propiedad modo de operación de clúster especifica cómo se desea realizar la gestión de las direcciones MAC de las tarjetas de red de propiedad del clúster NLB. El clúster NLB en principio realiza un MAC Spoofing o falseo de MAC de las respuestas ARP sobrescribiendo los paquetes salientes del clúster con una dirección MAC e IP virtual y recibe los paquetes a las mismas direcciones.



**Figura 24-1.** Panel de configuración de parámetros del clúster NLB  
Fuente: [https:// www.guillesql.es/Articulos/Cluster\\_NLB\\_Unicast\\_Multicast.aspx](https://www.guillesql.es/Articulos/Cluster_NLB_Unicast_Multicast.aspx)

EL balanceador de carga NLB presenta dos opciones o alternativas para la modalidad de operación del clúster:

- J UNICAST.- Se encuentra configurada por defecto y es la más recomendada. Se asigna la dirección MAC a todas las tarjetas de red del clúster, pero las MAC de cada tarjeta no son

utilizadas, es decir, todas tendrán o mantendrán la misma MAC del clúster, de esta manera se resuelven tanto la IP del clúster como la de la propia tarjeta con la MAC del clúster.

No es posible realizar una conexión desde un host del clúster a otro dentro del mismo a través de la tarjeta de red usada en el clúster ya que ambas comparten la misma MAC y por ello no pueden diferenciar entre el destinatario y el emisor.

J) **MULTICAST.**- La dirección MAC se asigna a todas las tarjetas de red del clúster, pero adicionalmente cada tarjeta mantiene su propia MAC, es decir, tendrán dos direcciones MAC, la MAC del clúster es usada únicamente para comunicarse con los clientes permitiendo la individualidad de cada tarjeta de red. Esto implica que una tarjeta de red puede manejar tanto el tráfico con los clientes como entre los nodos.

En ciertos equipos routers no es posible esta configuración debido a su implementación ARP como es el caso de los enrutadores CISCO. Multicast puede ofrecer un rendimiento inferior al modo de operación UNICAST debido a que utiliza una única tarjeta de red para ambos tipos de tráfico ya analizados. Es posible activar la opción IGMP Multicast que es utilizado en el caso de tener problemas de convergencia.

Microsoft recomienda la utilización del modo de operación UNICAST exceptuando el caso de que se tenga únicamente una sola tarjeta de red para soportar el tráfico y sea necesaria la comunicación intra-host. Siendo una de las principales razones de este consejo el evitar posibles fallos o errores con enrutadores.

#### ***1.5.6. Alta disponibilidad con heartbeat***

Heartbeat es un servicio de valga la redundancia provee de servicios de agrupamiento a servidores, es decir, permite saber a los administradores si un nodo está presente o ausente (UP o DOWN) a través del uso de mensajes entre los nodos pertenecientes a dicho clúster. Se encuentra diseñado específicamente para brindar el agrupamiento de alta disponibilidad para ofertar cualquier tipo de servicio.

Inmediatamente cuando un servidor llega a su punto de saturación por la inundación de peticiones por parte de los clientes NLB necesita realizar el cambio automático para el cambio de servidor y reenviar las nuevas peticiones a un segundo servidor, esto es posible gracias a Heartbeat que a través de su informe constante de estado y situación de los servidores del clúster permite consolidar esta acción por parte del NLB, sin embargo suele pasar mucho tiempo antes de que el NLB considere que un servidor se encuentra saturado y realice el cambio, en dicho lapso de tiempo la conexión se ralentiza y adquiere un nivel alto de latencia.

Encontrándonos en esta situación NLB permite la ejecución de un cambio manual de servidor con la finalidad de establecer comunicación y atender las peticiones sin llegar a iniciar una pérdida de tiempo en la respuesta de peticiones.

### ***1.5.7. Propiedades del equilibrio de carga de red***

Dentro del balanceador NLB podremos encontrar varias propiedades que actúan para realización equilibrada de la distribución de la carga, entre ellas tenemos a:

- J Direcciones IP del clúster. - Aquí se especifica la dirección IP principal adjudicada al clúster, teniendo libre la posibilidad de agregar IPs adicionales.
- J Parámetros del clúster. - Podemos configurar tanto la dirección IP, la máscara de subred, el nombre completo, la dirección física o MAC, el modo de operación del clúster.
- J Reglas de puerto. - La ficha muestra únicamente el resumen de todas las reglas existentes, mismas que pueden ser agregadas o editadas dependiendo de los parámetros dirección IP del clúster, intervalo de puerto, protocolos, modo de filtración, afinidad, carga, prioridad de tratamiento.
- J Parámetros del host. - Pueden ser nombrados los parámetros de prioridad o identificador de host único, dirección IP específica o única de host, máscara de subred y estado del host inicial.

### ***1.5.8. Control de servidores en NLB***

El control de host que puede realizarse con NLB conforman; el manejo de parada e inicio del tráfico del clúster NLB, la suspensión o resumen del NLB, la habilitación o deshabilitación del manejo de tráfico con NLB para puertos específicos, y, la deshabilitación de un nuevo manejo de tráfico con NLB para puertos específicos.

Para el primer caso debemos encontrarnos en situaciones como mantenimiento de operaciones, el tratamiento de problemas con aplicaciones con uno o más hosts en el clúster o a su vez querer disminuir el número de servidores debido a que el tráfico generado no requiere del uso de mayor recurso.

Si deseamos acudir al tercer caso entonces debemos encontrarnos en la situación en la que todos los puertos especificados por la regla de puerto sean afectados por la deshabilitación del manejo del tráfico, cabe mencionar que se puede volver a habilitar sin ningún problema.

Todos los cambios o configuraciones se pueden realizar en el NLB manager, la interfaz es bastante intuitiva, únicamente es necesario dirigirse a la pestaña de Control de Hosts e indicar con un click que se desea para, iniciar, reiniciar, habilitar o deshabilitar un host.

Debe considerarse que todos los cambios realizados lo pueden hacer únicamente el administrador o equipo de administrador por lo tanto como prioridad debemos ser parte de dicho equipo.

## ) **Linux Virtual Server**

Este software de balanceo de carga es utilizado muy comúnmente por empresas pequeñas cuya carga de datos es relativamente baja, es adoptada como medida de solución de fácil acceso ya que se puede configurar sobre un servidor reutilizado obteniendo resultados positivos sobre la red de servidores, las principales ventajas que podemos citar son; aumento en el rendimiento de la red, características como las conseguidas con hardware especializado, aumento de seguridades, costo de implementación muy bajo en comparación con los ADC, licencia GPL, sin embargo las dificultades son muchas cuando nuestra red posee requerimientos muy altos, las limitaciones físicas de los equipos utilizados se vuelven evidentes y la solución deja de serlo.

Es configurada sobre distribuciones de Linux con mayor frecuencia sobre Red Hat.



**Figura 25-1.** Servidores virtuales linux  
Fuente: mhosting.in

## ) **Perlbal**

Posee funciones tanto de balanceo de carga como de servidor web, nos brinda flexibilidad junto con la agilidad de añadir o remover servicios, es utilizado además como un servidor

proxy en sentido reverso. Al igual que en el ya mencionado LVS las dificultades son halladas en las características físicas que presentan los servidores en los que se implementan.



**Figura 26-1.** Perlbal balanceador de carga  
Fuente: <https://www.perl.org>

#### **1.5.9. Mecanismos de balanceo**

##### **) ECMP**

Equal Cost Multi-Path, este mecanismo emplea una agregación de todas las puertas de enlace en una misma ruta, realiza un balanceo persistente por conexión y es sencillo de implementar, pero resulta impráctico debido a que no se puede controlar la forma en la que se realiza el balanceo, no funciona con puertas de enlace iguales.

##### **) PCC**

Peer Connection Classifier, en este procedimiento la dificultad al momento de su implementación ya es mayor en comparación al anterior, utiliza algoritmos de combinación y clasificación de campos tomados de las cabeceras IP de origen y destino y de los puertos de origen y destino.

La desventaja que presenta el mecanismo la hallamos en la distribución de la cantidad de conexiones que no es realizada de forma equitativa.

##### **) NTH**

Permite una distribución de conexiones equitativa y adiciona un contador, el cual es incrementado cuando una de las reglas recibe un paquete, cuando el contador llega al máximo se reinicia automáticamente. (Microsoft, 2016)

### ***1.5.10. Tipos de balanceo de carga***

#### **) Balanceo de Carga Estático**

Es conocido también como planificación del problema o mapeado del problema, antes de realizar cualquier proceso de balanceo es el primero en ser tomado en cuenta, pero presenta gran desventaja cuando algunos sistemas presentan retardos en sus comunicaciones motivados por cualquier circunstancia y que son variables, en este punto la incorporación de una variable de retardo de comunicación es casi una misión imposible.

#### **) Balanceo de carga dinámico**

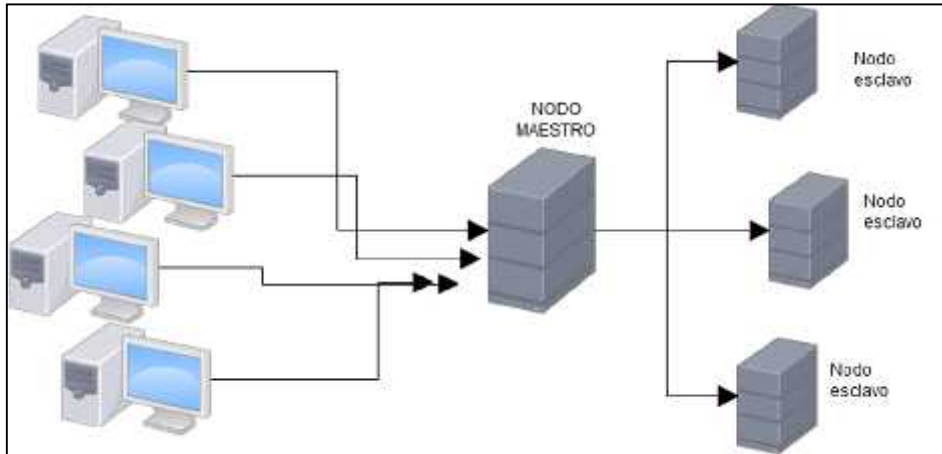
A diferencia del anterior éste tipo de balanceo se trata durante la ejecución de los procesos. La división de la carga computacional dependerá de las tareas que se ejecutan en ese momento y no de una estimación de tiempo que pueden tardarse en ser ejecutadas, a pesar de que en algoritmo de funcionamiento del balanceo de carga dinámico desarrolla una sobrecarga consigo, resulta mucho más eficiente que el estático.

Existen dos modos definidos dentro del balanceo dinámico, el primero conocido como centralizado, se refiere a un nodo principal en el que se configura el comportamiento general del sistema, lo cual facilita su administración. El segundo modo se denomina descentralizado, aquí la gestión se la realiza en cada nodo, los ficheros de configuración se encuentran localmente, esto brinda al sistema una mayor tolerancia a fallos, pero dificulta su administración. (Gómez, 2011, pp.3-10)

#### **) Centralizado**

Su arquitectura consta de un nodo maestro, a él llegan todas las tareas solicitadas y éste a su vez las envía a cada uno de los nodos esclavos, estos últimos se encargan de realizar las tareas y cuando las terminan solicitan nuevamente al maestro se le asigne una nueva. Esta técnica es empleada también con problemas que poseen tareas de diferentes tamaños y se la denomina bolsa de trabajo o programación por demanda, puede ser utilizada cuando el número de tareas varía durante la ejecución. Será utilizada para tareas que tengan poca carga computacional y se tengan pocos servidores esclavos. (Teixeira, 2016, pp.5-12)

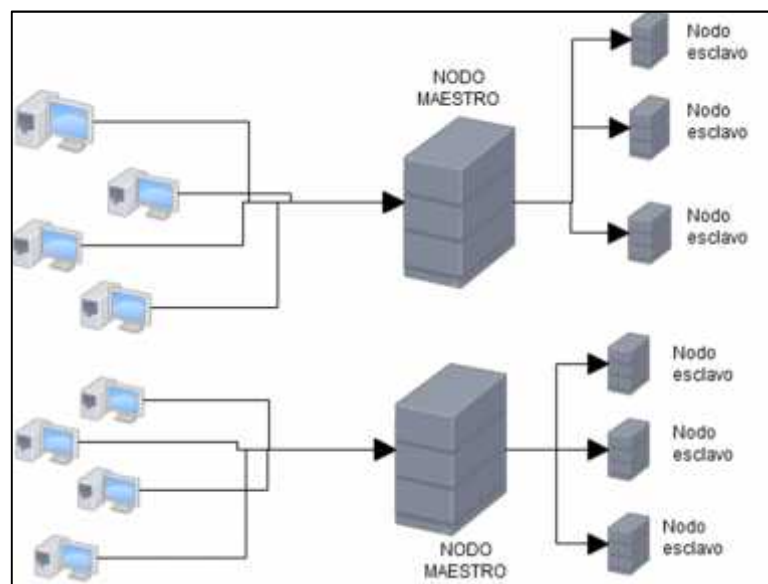




**Figura 27-1.** Arquitectura del balanceo de carga dinámico centralizado  
 Realizado por: Aguilar R. 2016

) **Distribuido**

Al contrario del centralizado en esta técnica ya no se cuenta con un solo nodo maestro, ahora se tiene varios, eliminando así la monopolización de la distribución de las tareas, se asigna a cada maestro un conjunto de tareas que serán enviadas de manera organizada a su propio conjunto de nodos esclavos, de esta manera no se permite que se generen colisiones de peticiones realizadas simultáneamente por parte de los esclavos a sus maestros una vez que estos terminan sus tareas. Aplicaremos el balanceo de carga dinámico distribuido cuando el número de servidores esclavos que posee la red sea alto. (Munera, 2009, pp.11-13)



**Figura 28-1.** Arquitectura del balanceo de carga dinámico distribuido  
 Realizado por: Aguilar R. 2016

## 1.6. Estructura la red del DTIC

### 1.6.1. Topología de la red

La red de la Dirección de Tecnologías de la Información y Comunicación se encuentra constituida a rasgos generales por una unidad de administración centralizada encargada de tomar las peticiones de los clientes y redirigirlas al clúster de servidores, el que a su vez se encuentra conectado a las bases de datos y terminan el ciclo con las respuestas de servicio para el usuario.

### 1.6.2. Número de usuarios aproximados

Para el año 2016 la red ha manejado un tráfico diario de usuarios desde un mínimo de 4899 hasta un máximo de 13382, obteniendo un promedio diario de clientes en la red de 6127.

### 1.6.3. Recursos del clúster

El clúster utilizado por la red DTIC es del tipo homogéneo y se encuentra conformado por servidores virtuales.

## 1.7. Soluciones implementadas

### 1.7.1. Solución con software

**Tabla 7-1. Solución elegida en software.**

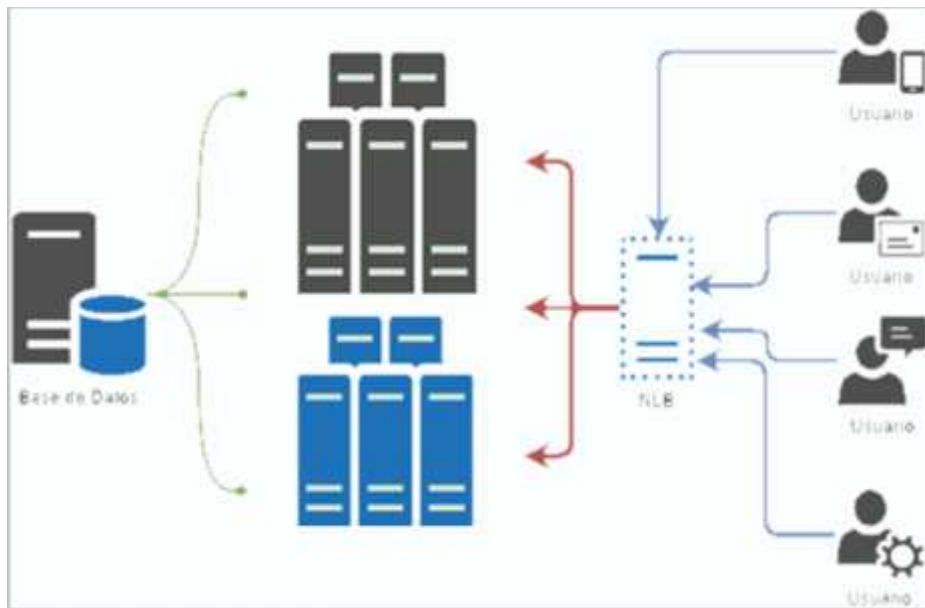
Solución por Software	
1	Desactivar actualizaciones de Windows Server 2008.
2	Activación del Balanceo de carga.
3	Disponibilidad de 5 servidores
4	Elección de Afinidad “Sencilla” se acopla a las demandas de la red.
5	Activación de “Heartbeat”
6	Recursos utilizados en servidores de 8 gb de RAM.
7	Utilización de 2 monitores para mejor administración.

Realizado por: Aguilar R. 2016

El software que fue implementado es NLB, éste fue escogido primordialmente por el licenciamiento del sistema operativo Windows Server 2008 con el que cuenta la ESPOCH, de modo que el recurso no se vea desperdiciado y sea bien aprovechado.

La configuración establecida en esta solución fue a través del método cíclico, es decir, todas las solicitudes se envían al primer servidor cuando éste se satura, el balanceador envía todas las peticiones al siguiente servidor, cuando el segundo servidor se satura entonces el balanceador envía las peticiones al tercer servidor y así sucesivamente.

El balanceador de carga NLB poseía un máximo de capacidad de 8Gb para realizar el manejo de las peticiones hacia cada servidor. Se encontraba implementado en comunicación unicast/multidifusión.



**Figura 29-1.** Arquitectura red DTIC con solución de balanceo por software.  
Realizado por: Aguilar R. 2016

### 1.7.2. Solución con hardware

**Tabla 8-1.** Solución elegida para hardware.

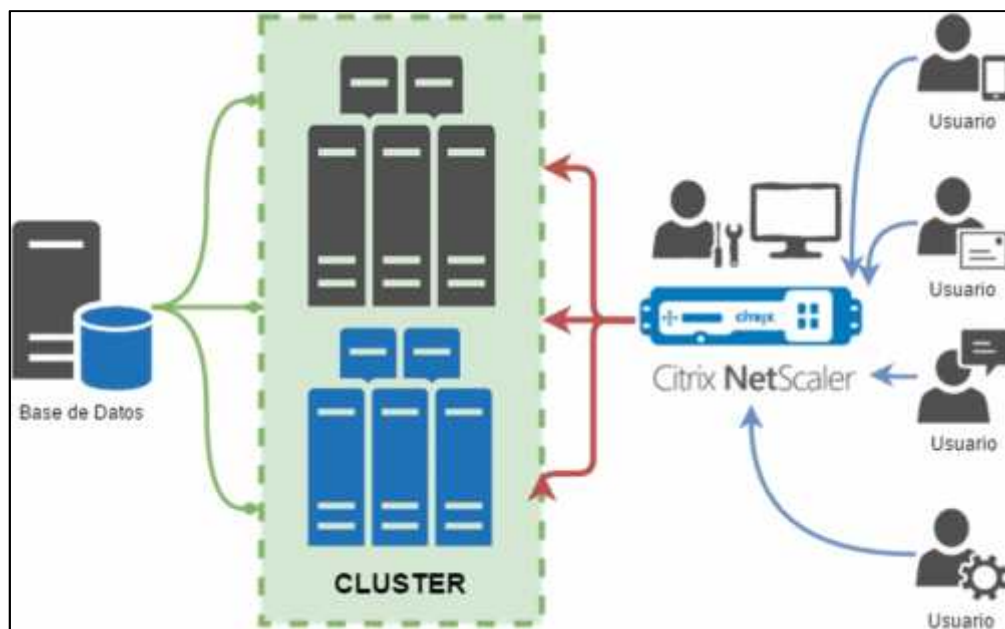
Solución por Hardware	
1	Disponibilidad de 5 servidores.
2	Elección del algoritmo SOURCEIPHASH
3	Elección del algoritmo de persistencia COOKIEINSERT

Realizado por: Aguilar R. 2016

Previo a un estudio técnico que reposa en las instalaciones de la Dirección de Tecnologías de la Información y Comunicación, se seleccionó a Citrix NetScaler 5650 como el hardware para esta solución de balanceo.

Se lo implementó bajo la comunicación unicast/multidifusión, posee asignado una capacidad de 8Gb para realizar el respectivo manejo de las solicitudes para cada servidor. El balanceador de

carga Citrix se encuentra centralizado y utiliza el algoritmo SOURCE IP HASH, añadiendo niveles de persistencia en las conexiones establecidas, la persistencia utilizada en la configuración del Citrix en la Dirección se conoce como COOKIE INSERT, esta persistencia consiste en la inserción de una cookie HTTP en el espacio de la cabecera de la respuesta HTTP, la cookie contiene información del servidor en donde deben ser enviadas las peticiones del cliente, el cliente almacena esta cookie y la incluye en las subsecuentes peticiones, NetScaler utiliza proceso para elegir al mismo servidor para las posteriores solicitudes. Como ya se mencionó en los métodos que se podían utilizar con Citrix, el método HASH permite que las peticiones de una misma sesión no sean distribuidas a otros servidores al contrario con la persistencia todas las que posean la misma dirección de origen y destino sean comunicadas a un mismo servidor generando una misma llave hash. La ventaja del uso de este algoritmo es que es muy útil cuando se desea mantener una sesión activa a pesar de existir una desconexión o reconexión del cliente.



**Figura 30-1.** Arquitectura red DTIC con solución de balanceo por hardware.  
Realizado por: Aguilar R. 2016

#### 1.7.2.1. Selección y configuración de persistencia

La persistencia puede ser configurada en un servidor virtual si se desea mantener los estados de conexión en los servidores representados por este servidor virtual. A continuación, se detalla los tipos de persistencia que pueden ser configurados (tabla 9-1).

**Tabla 9-1. Tipos de persistencia para cada tipo de servidor virtual.**

Persistence TypeHeader 1	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge
Source IP	YES	YES	YES	YES	YES
CookieInsert	YES	YES	NO	NO	NO
SSL Session ID	NO	YES	NO	NO	YES
URL Passive	YES	YES	NO	NO	NO
Custom Server ID	YES	YES	NO	NO	NO
Rule	YES	YES	NO	NO	NO
SRCIPDESTIP	N/A	N/A	YES	YES	N/A
DESTIP	N/A	N/A	YES	YES	N/A

**Fuente:** <https://docs.citrix.com/en-us/netscaler/11/getting-started-with-netscaler/load-balancing/configure-persistence-settings.html>

Pueden ser especificadas las persistencias para un grupo de servidores virtuales. Cuando se habilita la persistencia en el grupo, las peticiones del cliente se redirecciona al mismo servidor seleccionado independientemente de que servidor del grupo haya recibido dicha petición. Una vez que el tiempo establecido para persistencia haya transcurrido se puede seleccionar cualquier servidor virtual del grupo para que maneje las solicitudes entrantes de los clientes.

#### *1.7.2.2. Configuración de la persistencia basada en cookies*

Cuando se elige esta opción NetScaler añade una cookie HTTP en el apartado de la cabecera de la respuesta HTTP. Esta cookie contiene información sobre el servidor que fue elegido para manejar las solicitudes a continuación.

NetScaler inserta la cookie de la siguiente manera:

`<NSC_XXXX>=<SERVICE IP><SERVICE PORT>`

En esta expresión `<NS_XXXX>` es el ID del servidor virtual que se deriva del nombre del servidor virtual.

`<SERVICE IP>` es la dirección IP del servidor y se encuentra en hexadecimal.

`<SERVICE PORT>` es el valor del puerto del servidor seleccionado expresado en hexadecimal.

El balanceador encripta tanto la IP del servidor como el puerto del mismo cuando inserta la cookie y los desencripta cuando este recibe una cookie.

Para realizar estos cambios en el sistema, nos debemos dirigir hasta el panel SYSTEM > SETTINGS, en el panel detallado encontraremos la opción CHANGE HTTP PARAMETERS dando click en este vínculo nos dirigimos a un cuadro de diálogo CONFIGURE HTTP PARAMETERS, debajo del parámetro cookie seleccionamos VERSION 0 o VERSION 1.

Si lo deseamos podemos hacer los cambios desde el command prompt de la siguiente manera como lo indica la figura 30-1.

```
> set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
Done
> show lb vserver vserver-LB-1
vserver-LB-1 (10.102.29.60:60) - HTTP  Type: ADDRESS
.
.
.
Persistence: COOKIEINSERT (version 0) Persistence Timeout: 2 min
.
.
.
Done
>
```

**Figura 31-1.** Comandos de configuración de persistencia por cookies.

Fuente: <https://docs.citrix.com/en-us/netscaler/11/getting-started-with-netscaler/load-balancing/configure-persistence-settings.html>

Si deseamos configurar esta persistencia basada en cookies utilizando la utilidad de configuración entonces debemos realizar lo siguiente:

- a) Nos dirigimos al apartado TRAFFIC MANAGEMENT > LOAD BALANCING > VIRTUAL SERVERS.
- b) En el panel que aparece seleccionamos el servidor virtual en cual queremos realizar la configuración de persistencia y le damos un click para abrirlo.
- c) En el cuadro de diálogo CONFIGURE VIRTUAL SERVER (LOAD BALANCING) en la pestaña METHOD AND PERSISTENCE en la lista PERSISTENCE seleccionamos COOKIEINSERT.
- d) Damos click en OK y procedemos a verificar que el servidor seleccionado posee correctamente realizados los cambios en la sección de vista de detalles en la parte superior del panel.

### 1.7.2.3. Configuración de la persistencia basada en los ids de los servidores en URLs

Puede mantenerse este tipo de persistencia a través de una técnica llamada persistencia pasiva de URL, NetScaler extrae el ID del servidor desde la respuesta del mismo y es colocado en la consulta URL de la solicitud del cliente. El ID del servidor es una dirección IP y el puerto especificado como número hexadecimal. Citrix extrae el ID del servidor de la siguiente petición del cliente y lo usa para seleccionar al servidor.

Para configurar esta opción a través del prompt debemos realizarlo utilizando las siguientes líneas de comandos:

- o `set lb vserver <name> -persistenceType URLPASSIVE`
- o `show lb vserver <name>`

```
> set lb vserver vserver LB 1 persistenceType URLPASSIVE
Done
> show lb vserver vserver-LB-1
vserver-LB-1 (10.102.29.60:00) - HTTP Type: ADDRESS
.
.
.
Persistence: URLPASSIVE Persistence Timeout: 2 min
.
.
.
Done
>
```

**Figura 32-1.** Comandos de configuración de persistencia por URL PASSIVE.  
Fuente: <https://docs.citrix.com/en-us/netscaler/11/getting-started-with-netscaler/load-balancing/configure-persistence-settings.html>

Si lo que deseamos es hacerlo a través de la utilidad de administración debemos hacer lo que se detalla:

- a) Dentro del administrados nos dirigimos a **LOAD BALANCING > VIRTUAL SERVERS**.
- b) En el panel a continuación seleccionamos el servidor en el que deseamos configurar el parámetro.
- c) En el cuadro de diálogo **CONFIGURE VIRTUAL SERVER (LOAD BALANCING)** en la pestaña **METHOD AND PERSISTENCE** en la lista **PERSISTENCE** seleccionamos **URLPASSIVE**.
- d) En el cuadro de texto **TIME-OUT (MIN)** escogemos el tipo de valor para el time-out.
- e) En el cuadro de texto **RULE** ingresamos una expresión válida damos click en **CONFIGURE** para proceder a **CREATE EXPRESSION** y crear la expresión.
- f) Damos click en **OK** y procedemos a verificar como en el anterior caso.

## CAPITULO II

### 2. MARCO METODOLÓGICO

#### 2.1. Introducción

Durante el desarrollo de éste capítulo se podrá observar la metodología utilizada para la selección y configuración de servidores virtuales, sistemas de clustering, balanceadores o distribuidores de carga con sus respectivos algoritmos de distribución.

Para la elección de las soluciones de balanceo a ser implementadas para su posterior confrontación fue necesario un escogimiento selectivo apoyado en los parámetros más representativos de cada tecnología como: valor referencial de compra, licenciamiento, soporte técnico, número de conexiones, eficiencia teórica, entre otros.

Las actividades de configuración muestran en detalle los múltiples caminos adoptados para la consecución de los escenarios de prueba que se requieren.

Todos los procesos son implementados en la red de DTIC bajo los estándares que poseen establecidos el departamento técnico para los procesos de análisis y prueba con la finalidad de no comprometer la integridad de los servicios.

#### 2.2. Escenario de pruebas

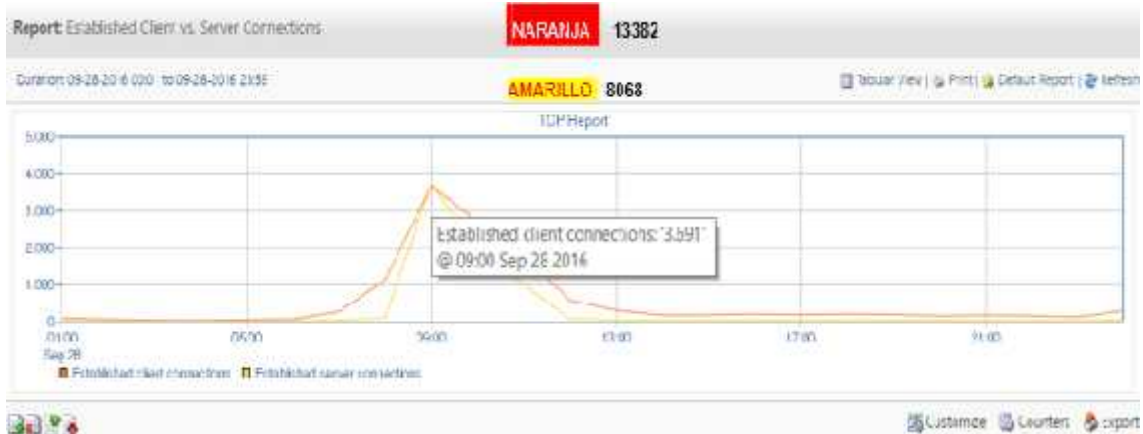
Los datos obtenidos en el balance de hardware se obtuvieron las 2 semanas de matrículas que empezó del día 19 de septiembre de 2016 hasta el 30 de septiembre de 2016 que fueron días picos por la excesiva conexión de usuarios hacia la aplicación OASIS.

Al balanceador de carga por software se lo conecto en los días que se asignaron el pase de notas del primer parcial en la ESPOCH donde fue una semana que no obtuvo tantas conexiones de usuarios como se registró el día miércoles 28 de septiembre de 2016 que se ubica en la segunda semana de matrículas de la institución con una cantidad de 13382 conexiones establecidas hacia los servidores del sistema académico. (Figura 1-2)

La cantidad que fue escogida como límite máximo de conexiones para la comparación de ambos balanceadores fue de 13382 conexiones, para recrear esta cantidad máxima de conexiones en el

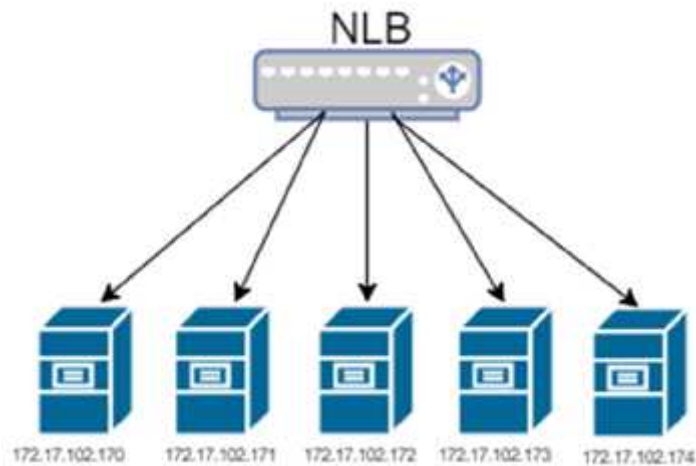


balanceo por software fue necesario aplicar un ataque DoS para llegar a estresar los servidores con el número de conexiones que se estableció.

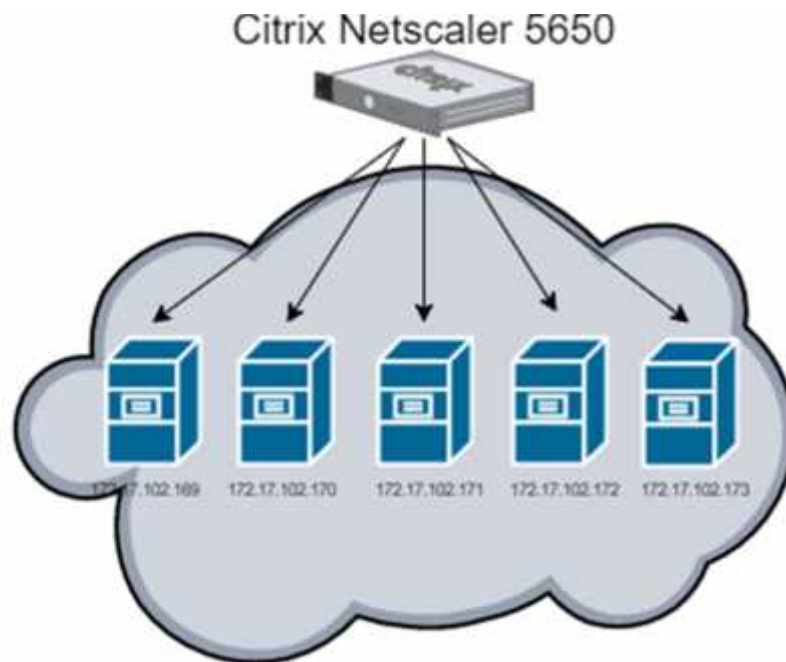


**Figura 1-2.** Número máximo de conexiones.  
Realizado por: Aguilar R. 2016

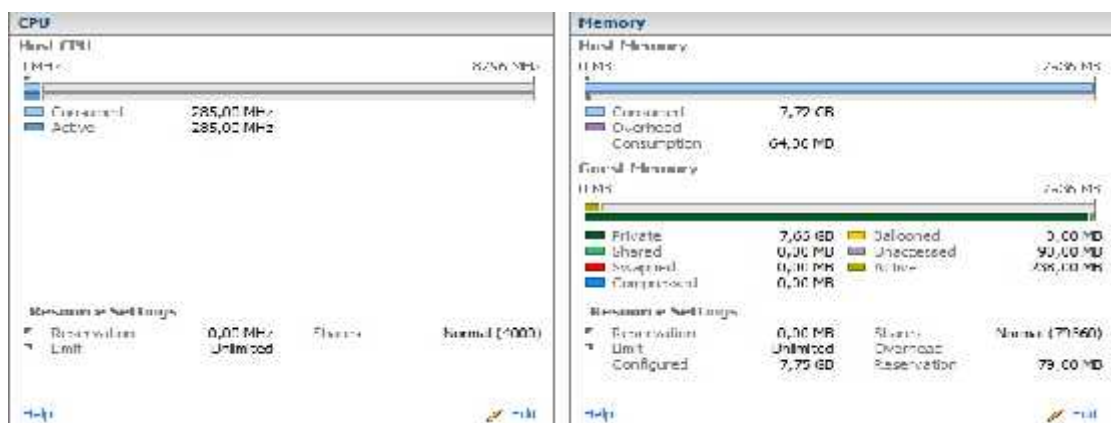
El equilibrio de carga por software y hardware trabaja con 5 servidores que mantuvieron los mismos recursos en ambos, para la obtención de datos en el balanceo por software los servidores no permitían Clustering, su esquema se lo puede observar en la Figura 2-2 mientras el balanceo por hardware si permite cluster entre los servidores y podemos observar su escenario en la Figura 3-2, para verificar los recursos utilizados en ambos servidores en la Figura 4-2.



**Figura 2-2.** Escenario de balanceo por software.  
Realizado por: Aguilar R. 2016

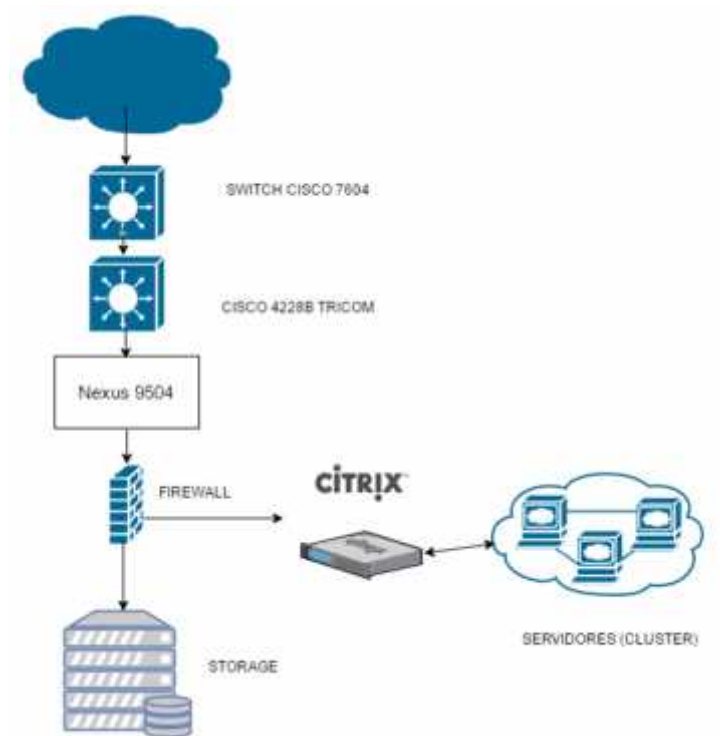


**Figura 3-2.** Escenario de balanceo por hardware.  
Realizado por: Aguilar R. 2016



**Figura 4-2.** Especificación del servidor.  
Realizado por: Aguilar R. 2016

Para mayor observación de la conexión del Citrix Netscaler 5650 con el data center podemos observar en la Figura 5-2 dando a conocer su estructura que se mantuvo en el estudio comparativo del equilibrio de carga para el sistema académico ESPOCH adaptado y funcionando a las nuevas actualizaciones que se tiene en el DTIC, indicando que se acoplo de manera rápida a las instalaciones del Data Center sin generar mal funcionamiento con los demás dispositivos de hardware.



**Figura 5-2.** Conexión del Citrix con el Data Center ESPOCH.  
Realizado por: Aguilar R. 2016

### 2.3. Selección de las soluciones por hardware y por software a través del método cualitativo por puntos

La elección de las soluciones que serán implementadas será efectuada mediante un análisis de propuestas, para el hardware las propuestas que se posee son: Barracuda, Citrix NetScaler, AVANU WebMux, y para el software: NLB, F5 BIG-IP Local Traffic Manager (LTM), Kemp Technologies.

Los parámetros escogidos para la selección tanto para la solución de hardware como de software son los siguientes:

P1→ Costo referencial de adquisición

P2→ App caché

P3→ Interfaz fácil de manejar

P4→ SSL transacciones/segundo

P5→ Memoria

P6→ HTTP respuestas/segundo

La escala para la ponderación relativa que será adjudicada en cada parámetro comprende los valores:

Máximo Valor: 3

Valor Medio: 2

Mínimo Valor: 1

El parámetro de respuestas/ segundo de HTTP tiene igual importancia que las transacciones/segundo de SSL. Tanto el cacheo de aplicaciones como la memoria de las soluciones tienen el mismo valor de importancia. Es más importante las transacciones/segundo de SSL que el cacheo de aplicaciones. El cacheo de aplicaciones tiene mayor relevancia que el costo de adquisición y a su vez éste último tiene más importancia que la interfaz.

Bajo estos criterios las condiciones halladas para nuestro escenario son las establecidas a continuación:

$P4 = P6$

$P2 = P5$

$P4 > P2$

$P2 > P1 > P3$

### 2.3.1. Selección de hardware

Como ya se mencionó las siguientes son las soluciones por hardware elegidas:

SH1 → Barracuda

SH2 → Citrix NetScaler

SH3 → AVANU WebMux

**Tabla 1-2. Tabla de datos técnicos de cada solución por hardware.**

Parámetros Solución	P1	P2	P3	P4	P5	P6
SH1	\$65000	Si	Sencilla	43000	128GB	1500000
SH2	\$50000	Si	Sencilla	42000	256GB	2400000
SH3	\$40000	No	Compleja	18000	64GB	420000

Realizado por: Aguilar R. 2016

En la tabla 1-2 tenemos cada uno de los datos obtenidos a través de investigación de las soluciones que se analizarán en base a cada parámetro anteriormente establecido, posteriormente la tabla 2-2 nos muestra las ponderaciones relativas asignadas a cada parámetro en dependencia de la escala prefijada.

**Tabla 2-2. Tabla de ponderaciones relativas de soluciones por hardware.**

Solución \ Parámetros	Parámetros					
	P1	P2	P3	P4	P5	P6
SH1	1	3	3	3	2	2
SH2	2	3	3	2	3	3
SH3	3	1	1	1	1	1
SH3	\$40000	No	Compleja	18000	64GB	420000

Realizado por: Aguilar R. 2016

En base a cada una de las condiciones obtenidas del enfrentamiento de los parámetros obtenemos la tabla 3-2 se realiza la determinación de las ponderaciones, cuando los parámetros son mayores o iguales que el otro se le asigna un 1 y cuando es menor se le asigna un 0, de este proceso se podrá deducir el peso real de cada parámetro.

**Tabla 3-2. Determinación de ponderaciones de cada parámetro solución por hardware.**

Parámetros	P1	P2	P3	P4	P5	P6	total	Pondera.
P1	0	0	1	0	0	0	1	0,06
P2	1	0	1	0	1	0	3	0,18
P3	0	0	0	0	0	0	0	0,00
P4	1	1	1	0	1	1	5	0,29
P5	1	1	1	0	0	0	3	0,18
P6	1	1	1	1	1	0	5	0,29
							17	1

Realizado por: Aguilar R. 2016

**Tabla 4-2. Cuadro de calificaciones solución por hardware.**

Parám Solución	P1		P2		P3		P4		P5		P6	
	Val	Calif.	Val	Calif.	Val	Calif.	Val	Calif.	Val	Calif.	Val	Calif.
SH1	1	0.17	3	0.43	3	0.43	3	0.5	2	0.33	2	0.33
SH2	2	0.33	3	0.43	3	0.43	2	0.33	3	0.5	3	0.5
SH3	3	0.5	1	0.14	1	0.14	1	0.17	1	0.17	1	0.17
TOTAL	6	1	7	1	7	1	6	1	6	1	6	1

Realizado por: Aguilar R. 2016

En la tabla 4-2 se realiza la calificación de los parámetros para después poder compararlos con su ponderación real a través de una multiplicación de la calificación por la ponderación, como se encuentra establecido en la tabla 5-2.

**Tabla 5-2. Cuadro de calificaciones solución por hardware.**

Solución Parámetros	Ponderación	SH1		SH2		SH3	
		Calif.	Po. Re.	Calif.	Po. Re.	Calif.	Po. Re.
<b>P1</b>	0,06	0.17	0.0102	0.33	0.0198	0.5	0.03
<b>P2</b>	0,18	0.43	0.0774	0.43	0.077	0.14	0.0252
<b>P3</b>	0,00	0.43	0	0.43	0	0.14	0
<b>P4</b>	0,29	0.5	0.145	0.33	0.0957	0.17	0.0493
<b>P5</b>	0,18	0.33	0.0594	0.5	0.09	0.17	0.0306
<b>P6</b>	0,29	0.33	0.0957	0.5	0.145	0.17	0.0493
<b>TOTAL</b>			0.3873		0.6057		0.1844
<b>PORCENTAJE</b>			38.77%		42.75%		18.44%

Realizado por: Aguilar R. 2016

Si analizamos los resultados obtenidos en la tabla 5-2 podemos darnos cuenta que el hardware con mejor valoración es el correspondiente a la solución número dos, por lo tanto, la selección para la solución de hardware recaerá en Citrix NetScaler.

### 2.3.2. Selección de software

Las soluciones de software propuestas por parte del departamento técnico de DTIC son:

SS1 → NLB

SS2 → F5 BIG-IP Local Traffic Manager (LTM)

SS3 → Kemp Technologies

En la siguiente tabla (Tabla 6-2) se encuentra encasillado los datos en base a cada parámetro investigado para su solución por software respectivamente.

**Tabla 6-2. Tabla de datos técnicos de cada solución por software.**

Solución \ Parámetros	P1	P2	P3	P4	P5	P6
SS1	\$---	Si	Sencilla	2700	32GB	80000
SS2	\$6089	Si	Sencilla	3000	64GB	100000
SS3	\$1990	Si	Sencilla	2600	32GB	70000

Realizado por: Aguilar R. 2016

La tabla 7-2 nos indica cuales son las ponderaciones relativas que fueron asignadas a cada parámetro en función de la escala que se ubicó al inicio del proceso de selección y que también fue manejada para la selección del hardware.

**Tabla 7-2. Tabla de ponderaciones relativas de soluciones por software.**

Solución \ Parámetros	P1	P2	P3	P4	P5	P6
SS1	3	3	3	2	2	2
SS2	1	3	2	3	3	3
SS3	2	3	3	1	2	1

Realizado por: Aguilar R. 2016

Para realizar la comparación de las calificaciones de cada parámetro con su valor real hemos de utilizar la misma tabla 8-2 en la cual se constatan la determinación de los parámetros y que a través del producto de la calificación por la ponderación obtendremos los pesos que indicarán cual será la solución adecuada para nuestro estudio.

**Tabla 8-2. Cuadro de calificaciones solución por software.**

Parámetros Solución	P1		P2		P3		P4		P5		P6	
	Valor	Calif.	Valor	Calif.	Valor	Calif.	Valor	Calif.	Valor	Calif.	Valor	Calif.
SS1	3	0.5	3	0.33	3	0.38	2	0.33	2	0.29	2	0.33
SS2	1	0.17	3	0.33	2	0.25	3	0.5	3	0.43	3	0.5
SS3	2	0.33	3	0.33	3	0.38	1	0.17	2	0.29	1	0.17
<b>TOTAL</b>	6	1	9	1	8	1	6	1	7	1	6	1

Realizado por: Aguilar R. 2016

**Tabla 9-2. Cuadro de calificaciones solución por software.**

Solución Parámetros	Ponderación	SS1		SS2		SS3	
		Calif.	Po. Re.	Calif.	Po. Re.	Calif.	Po. Re.
P1	0,06	0.5	0,03	0.17	0,0102	0.33	0,0198
P2	0,18	0.33	0,0594	0.33	0,0594	0.33	0,0594
P3	0,00	0.38	0	0.25	0	0.38	0
P4	0,29	0.33	0,0957	0.5	0,145	0.17	0,0493
P5	0,18	0.29	0,0522	0.43	0,0774	0.29	0,0522
P6	0,29	0.33	0,0957	0.5	0,145	0.17	0,0493
<b>TOTAL</b>			0,333		0,437		0,23
<b>PORCENTAJE</b>			33.3%		43.7%		23%

Realizado por: Aguilar R. 2016



Luego del proceso de ponderaciones obtenemos la tabla 9-2 que en conjunto indica las calificaciones obtenidas por cada solución en cada parámetro bajo investigación. En función de cada porcentaje de puntuación podemos deducir que la mejor solución sería la número dos, pero bajo el análisis de presupuesto para su adquisición la solución número uno cuenta ya con su licencia por lo que esto determina que la solución a ser implementada y probada será la número uno que corresponde al balanceador de carga NLB de propiedad de Microsoft.

## **2.4. Configuración de las soluciones**

### **2.4.1. Configuración solución por software**

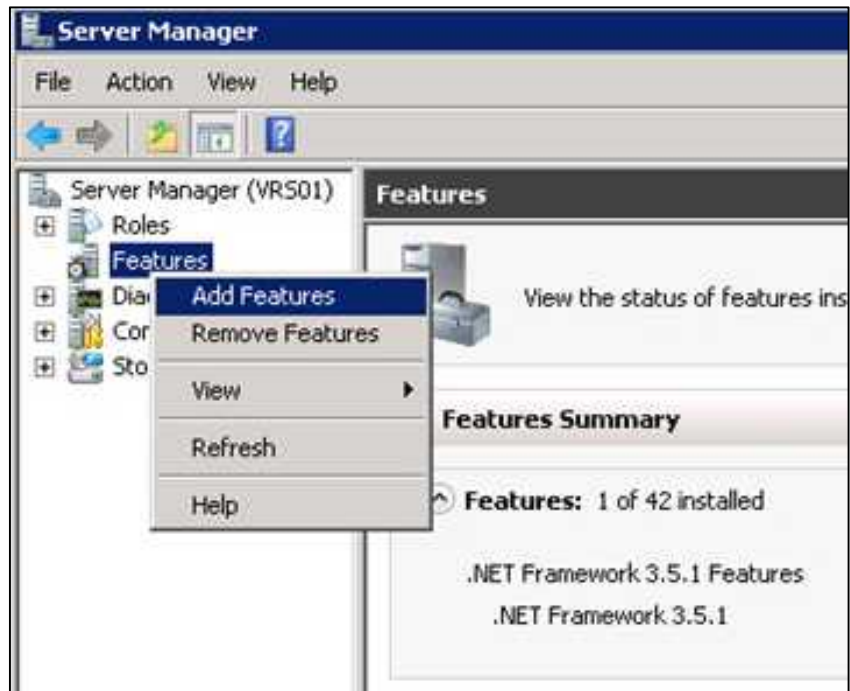
El siguiente procedimiento muestra cuales son los pasos a seguir para la configuración de un clúster NLB en Windows Server 2008 R2, el balanceador será parametrizado bajo el modo de operación unicast que es lo que comúnmente recomienda Microsoft y porque los equipos Cisco utilizados por la DTIC que poseen su distintiva configuración ARP.

Previo al proceso de configuración de NLB es imprescindible el contar con una dirección IP para ser utilizada como IP compartida, y, lógicamente tener todos los permisos administrativos correspondientes. Nuestra configuración será realizada sobre dos servidores, ya que teóricamente cuando el primero se sature las peticiones pasarán al segundo, hasta que el segundo servidor se sature el primero podrá resolver sus solicitudes y quedará listo para volver a recibir peticiones cuando el segundo se sature y así sucesivamente volviéndose un ciclo.

#### **Pasos de configuración:**

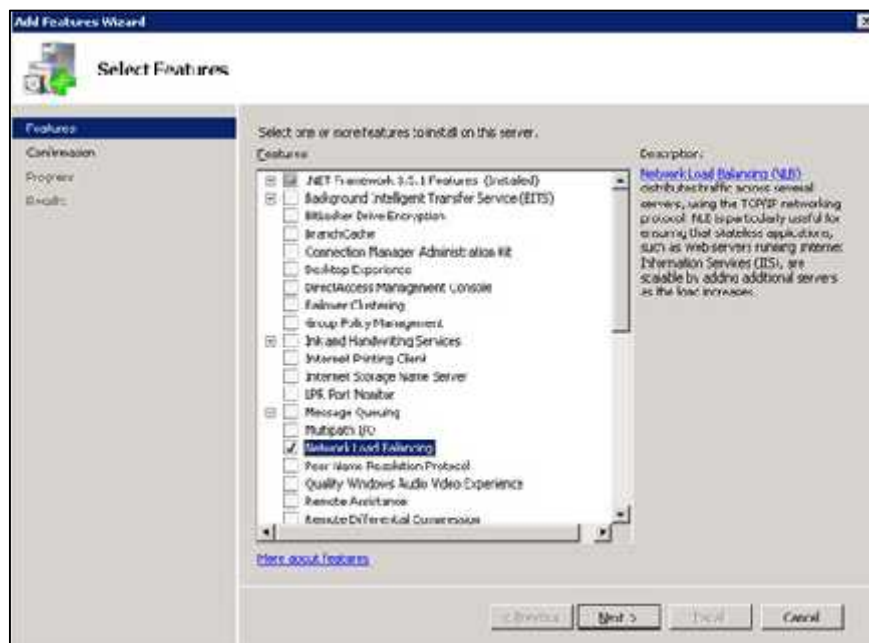
##### **a) Añadir la característica de NLB**

Para añadir esta característica en las máquinas elegidas podemos hacerlo a través de la herramienta administrativa *Server Manager* y desde esta utilizar la opción de menú *Add Features* para poder empezar la instalación de dicha característica.



**Figura 6-2.** Server manager >> Features >> Add features.  
 Fuente: [http://www.guillesql.es/Articulos/Instalar\\_Configurar\\_Network\\_Load\\_Balancing](http://www.guillesql.es/Articulos/Instalar_Configurar_Network_Load_Balancing)

Debemos seleccionar de la lista el elemento *Network Load Balancing* en la pantalla *Select Features* y damos click en *next*.



**Figura 7-2.** Pantalla Select Features del Server Manager.  
 Fuente: [http://www.guillesql.es/Articulos/Instalar\\_Configurar\\_Network\\_Load\\_Balancing](http://www.guillesql.es/Articulos/Instalar_Configurar_Network_Load_Balancing)

Revisamos que todo se encuentre puesto en *OK* en la pantalla de confirmación e iniciamos la instalación.

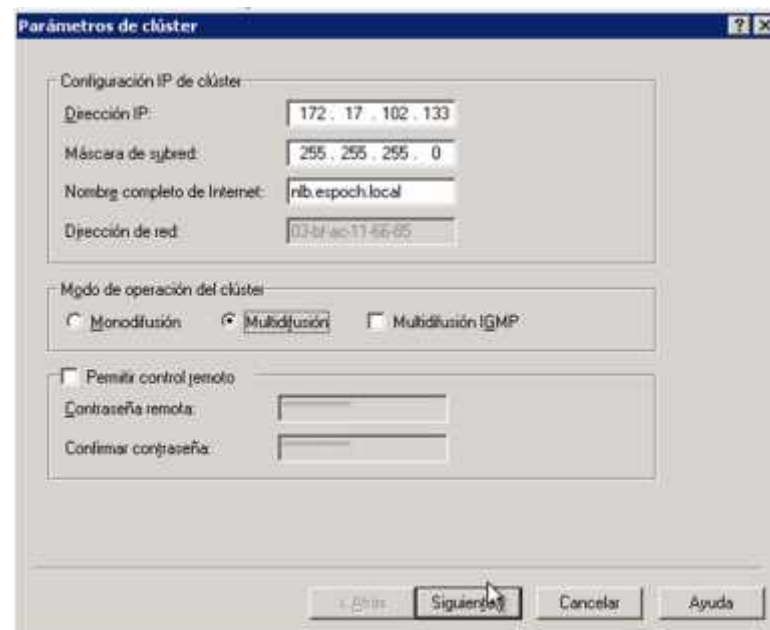
**b) Crear el clúster NLB con el primer servidor**

Volvemos a abrir la herramienta administrativa *Network Load Balancing Manager* y dentro de la pestaña clúster escogemos *New*.

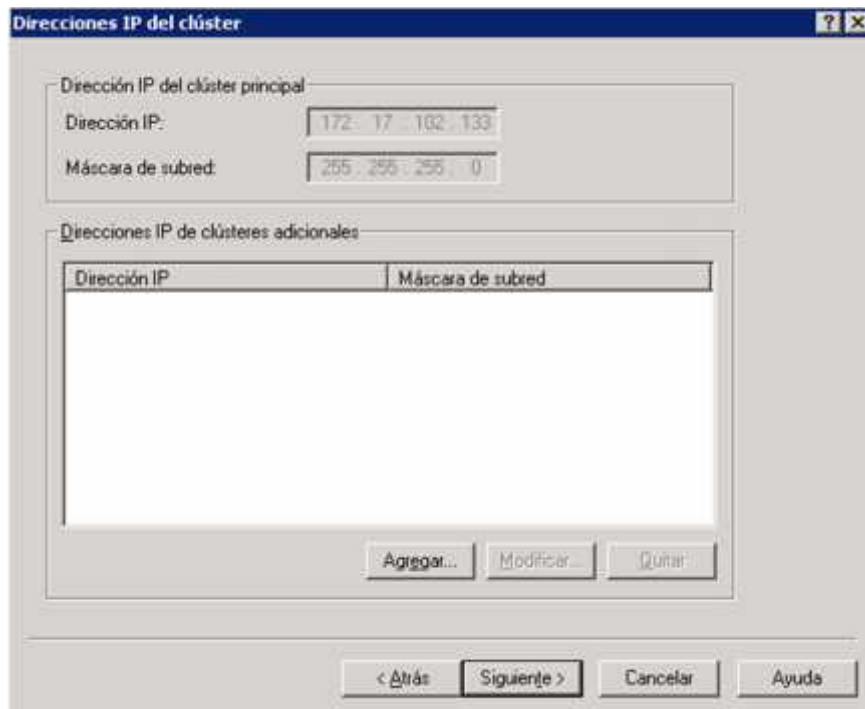


**Figura 8-2.** Crear nuevo clúster.  
Realizado por: Aguilar R. 2016

Se configuran los parámetros del clúster (nombre, dirección ip, modo de operación)

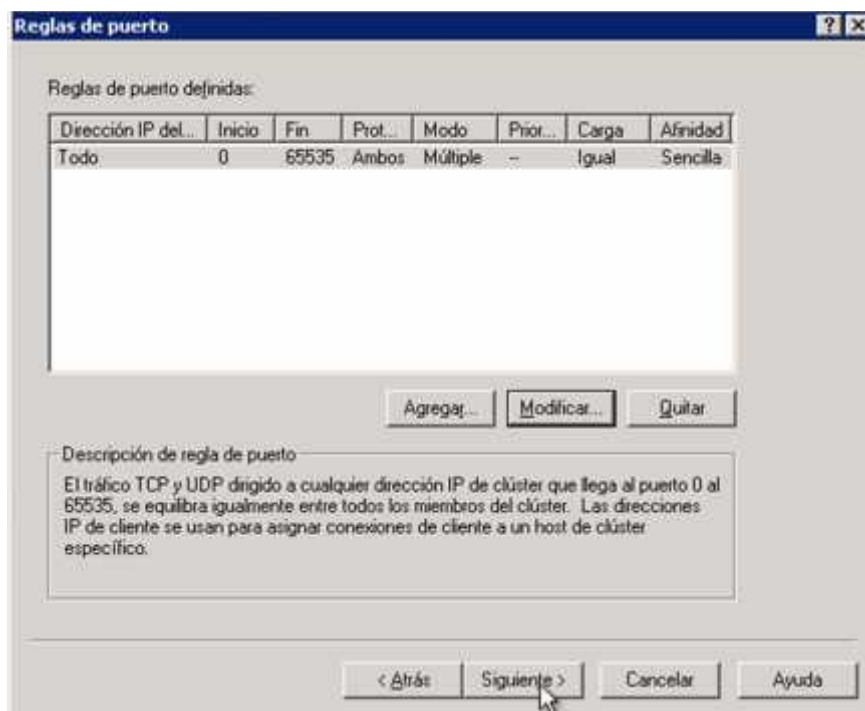


**Figura 9-2.** Parámetros clúster.  
Realizado por: Aguilar R. 2016



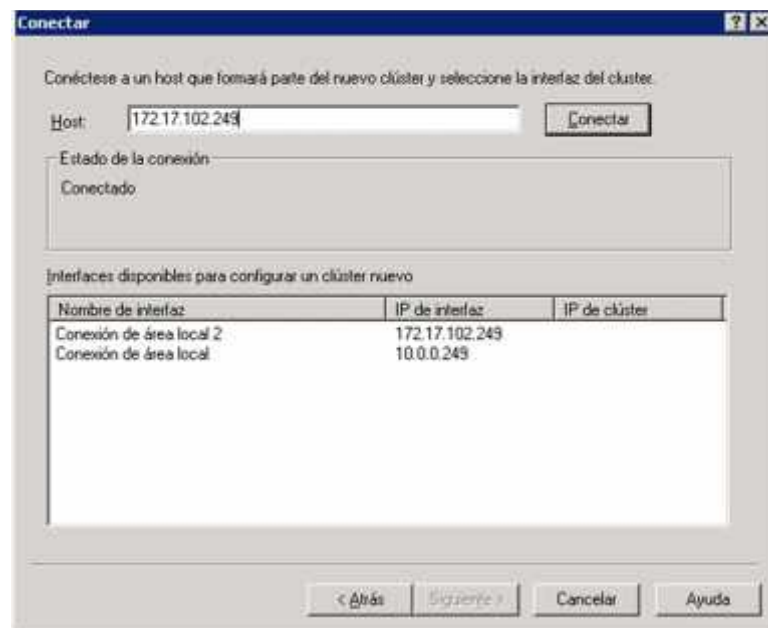
**Figura 10-2.** Direcciones ip del clúster.  
Realizado por: Aguilar R. 2016

A continuación se abrirá el cuadro *Nuevo Clúster: Reglas de Puertos*, en ella se pueden crear y modificar las reglas de puerto.

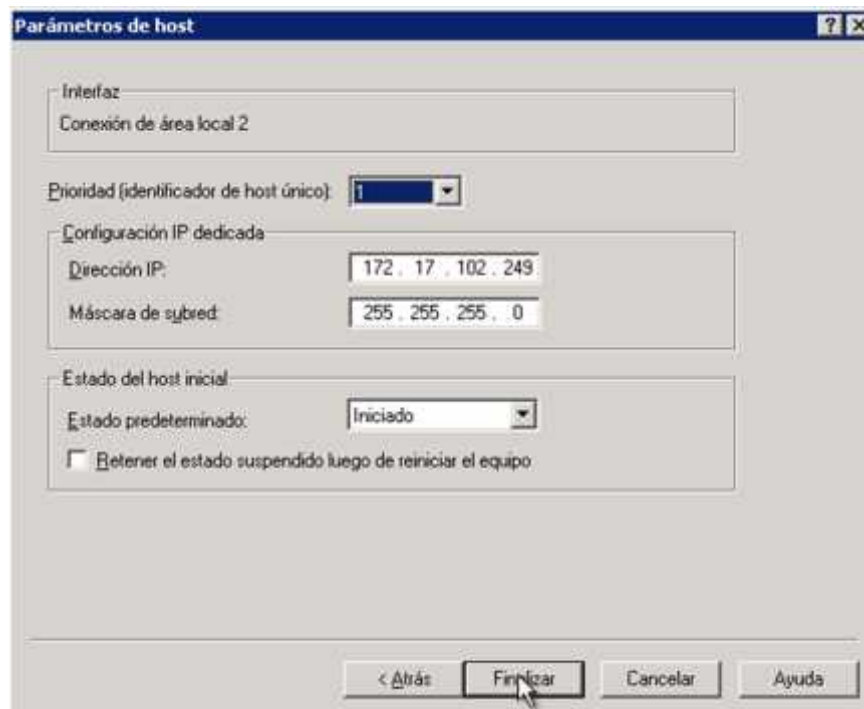


**Figura 11-2.** Reglas de puertos de clúster.  
Realizado por: Aguilar R. 2016

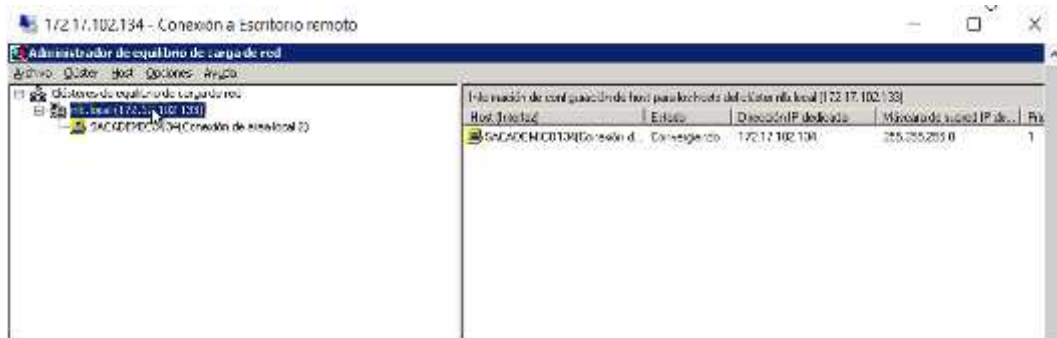
En el diálogo que aparece *Nuevo Clúster: Conectar* se debe especificar la dirección ip asignada al servidor sobre el cual queremos crear el nuevo clúster NLB y después dar click en *Conectar* seleccionamos la tarjeta de red sobre la que se montará el clúster NLB y damos click en *Siguiente*.



**Figura 12-2.** Conexión al servidor.  
Realizado por: Aguilar R. 2016



**Figura 13-2.** Parámetros de host.  
Realizado por: Aguilar R. 2016

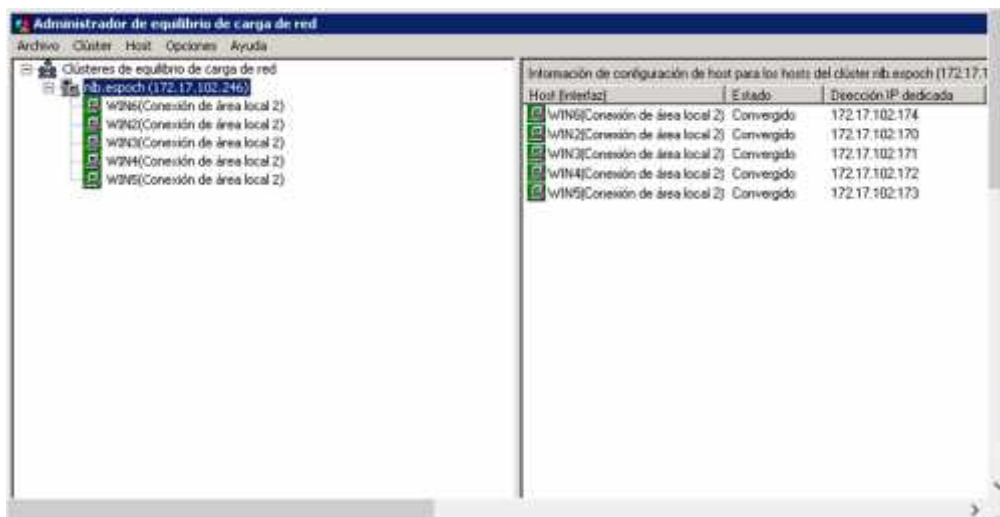


**Figura 14-2.** Pantalla principal con un clúster creado y un único host.  
Fuente: [http://www.guillesql.es/Articulos/Instalar\\_Configurar\\_Network\\_Load\\_Balancing](http://www.guillesql.es/Articulos/Instalar_Configurar_Network_Load_Balancing)

Cuando regresemos a la pantalla principal notaremos que se ha creado un clúster con un único servidor, por un pequeño lapso se perderá la comunicación en la red para que se efectúen los cambios, finalmente la red convergerá y se encontrará completamente operable.

**c) Añadir un nuevo host al clúster.**

Para añadir un nuevo clúster nos dirigimos al panel de navegación donde se encuentra nuestro clúster creado, le damos click derecho y seleccionamos la opción *Add Host To Clúster*, y continuamos con los mismos pasos que se siguieron para la asignación del primer servidor.



**Figura 15-2.** Pantalla principal los hosts añadido al clúster.  
Fuente: Configuración NLB DTIC

Podemos añadir los hosts que necesitemos, es necesario recalcar que para el monitoreo del clúster y por ende de los servidores, se debe acceder de forma individual a cada host para constatar, seguir

y realizar el reporte de todos los nodos ya que no existe un panel de navegación que englobe a todos.

#### **2.4.2. Configuración solución por hardware**

Nuestro balanceador Citrix NetScaler se encontrará configurado como pasarela de todas las comunicaciones para el acceso externo de todos nuestros clientes a los servicios web de la institución.

El equipo físico debe ser montado en el rack de servidores, se conectarán a él los cables de red provenientes de los servidores y firewalls, se constatará que posee correcto suministro de energía y se proseguirá a encenderlo a través de su switch.

- a) En un navegador web accedemos a la siguiente dirección IP asignada al puerto LOM por defecto: <http://192.168.1.3>.
- b) En el apartado *User Name* colocaremos *nsroot*.
- c) Y en el apartado de *Password* colocamos *nsroot*.
- d) Si deseamos acceder a través de consola lo que debemos hacer es ingresar con las credenciales expuestas y dentro de la herramienta administrativa seleccionar en la barra de Menú *Remote Control*. Nos dirigimos a *Options* damos click en *Console Redirection* y luego en *Launch Console*, Finalmente damos en *Yes* y listo nos encontraremos dentro de la consola.
- e) En la utilidad GUI de NetScaler podremos añadir de forma intuitiva el o los clúster que deseemos agregar y encontraremos detallado cada procedimiento para la asignación y selección de los parámetros.

Nuestro Citrix se encuentra configurado sobre cinco servidores homogéneos, cada uno posee 8Gb de capacidad, el método de distribución de carga elegido para nuestro balanceador es el algoritmo Source Ip Hash y junto con él añadida la persistencia de tipo Cookie Insert.

**CITRIX®**

**Login**

User Name  
nsroot

Password  
.....

Deployment Type  
 NetScaler Gateway  
 NetScaler ADC  
 NetScaler Gateway  
 XenMobile MDM  
 CloudBridge Connector

To use Secure HTTPS [Click here](#)

**Figura 16-2.** Pantalla de autenticación Citrix.

Fuente: <https://www.citrix.com/blogs/2013/07/03/citrix-netscaler-gateway-10-1-118-7-quick-configuration-wizard/>

**System**

NetScaler IP Address\* 10 . 102 . 189 . 165

Subnet IP Address\* . . .

Netmask\* 255 . 255 . 255 . 0

Hostname NS\_Test

DNS (IP Address) +

Time Zone\* GMT+05:30-IST-Asia/Kolkata

Change Administrator Password

**Continue**

**Figura 17-2.** Pantalla de configuración del direccionamiento Citrix.

Fuente: <https://www.citrix.com/blogs/2013/07/03/citrix-netscaler-gateway-10-1-118-7-quick-configuration-wizard/>



The screenshot shows the 'NetScaler Gateway Settings' configuration window. It contains the following fields and options:

- Name\*:** vserver1
- IP Address\*:** 10 . 102 . 189 . 167
- Port\*:** 443
- Redirect requests from port 80 to secure port\*
- Gateway FQDN:** vpn.citrix.agee

At the bottom, there are two buttons: 'Continue' (highlighted in blue) and 'Cancel'.

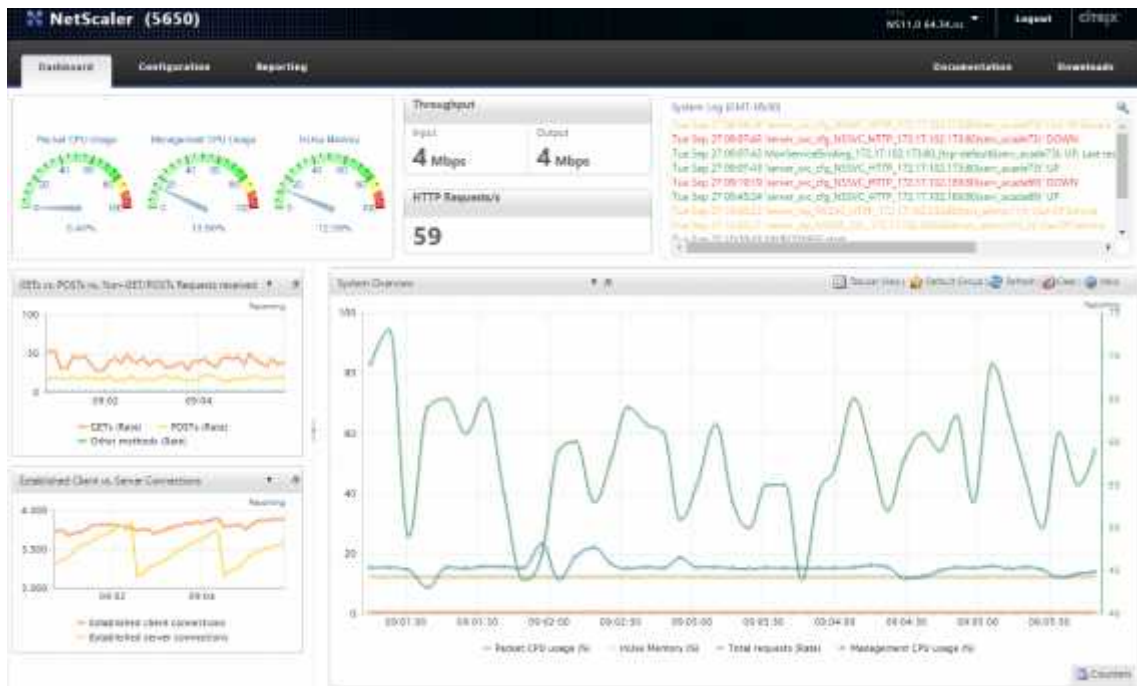
**Figura 18-2.** Pantalla de configuración de servidores.  
Fuente: <https://www.citrix.com/blogs/2013/07/03/citrix-netscaler-gateway-10-1-118-7-quick-configuration-wizard/>

The screenshot shows the 'Authentication Settings' configuration window. It contains the following fields and options:

- Primary Authentication:** LDAP
- Choose LDAP:**  **Configure New:**
- IP Address:** 10 . 102 . 80 . 102  Use
- Port:** 389
- Timeout (seconds):** 3
- Base DN:** dc=cytest,dc=agee
- Admin Base DN:** administrator@cytest.agee
- Server Login Name Attribute:** sAMAccountName
- Password:** [masked]
- Confirm Password:** [masked]
- Secondary Authentication:** None (dropdown menu with options: None, RADIUS)

At the bottom, there are two buttons: 'Continue' (highlighted in blue) and 'Cancel'.

**Figura 19-2.** Pantalla de configuración de seguridad por autenticación.  
Fuente: <https://www.citrix.com/blogs/2013/07/03/citrix-netscaler-gateway-10-1-118-7-quick-configuration-w/>



**Figura 20-2.** Pantalla de interfaz de monitoreo NetScaler.  
Fuente: Citrix Netscaler, DTIC; 2016

## CAPITULO III

### 3. MARCO DE RESULTADOS

#### 3.1. Evaluación del funcionamiento por software

El balanceador de carga durante el período de tiempo en el cual fue empleado, presentó el siguiente comportamiento al ser configurado sobre un clúster de dos servidores.

**Tabla 1-3. Datos estadísticos comprendidos desde el día uno hasta el día cinco de solución NLB.**

	día 1 de matrículas		día 2 de matrículas		día 3 de matrículas		día 4 de matrículas		día 5 de matrículas	
	Número	%	Número	%	Número	%	Número	%	Número	%
USO DE ANCHO DE BANDA	[3-5] (Mbps)	--	[3-4] (Mbps)	--	[4-6] (Mbps)	--	[5-6] (Mbps)	--	[4-6] (Mbps)	--
CONEXIONES ESTABLECIDAS CLIENTE	541	64,51 %	654	65,49%	1565	66,74%	2981	67,58%	3024	68,76%
CONEXIONES ESTABLECIDAS SERVIDOR	201	14,35 %	213	14,26%	546	13,44%	782	15,15%	795	15,87%
PAQUETES TCP RECIBIDOS	7739651	--	12730651	--	13239651	--	15857329	--	1479180	--
PAQUETES TCP TRANSMITIDOS	14371792	--	20877977	--	24378131	--	27439793	--	2552423	--

Realizado por: Aguilar R. 2016

**Tabla 2-3. Datos estadísticos comprendidos desde el día seis hasta el día diez de solución NLB.**

	día 6 de matrículas		día 7 de matrículas		día 8 de matrículas		día 9 de matrículas		día 10 de matrículas	
	Número	%	Número	%	Número	%	Número	%	Número	%
USO DE ANCHO DE BANDA	[4-5] (Mbps)	--	[5-6] (Mbps)	--	[4-5] (Mbps)	--	[28-33] (Mbps)	--	[25-30] (Mbps)	--
CONEXIONES ESTABLECIDAS CLIENTE	1754	66,13 %	1623	65,88 %	1684	65,34 %	4315	68,26 %	4256	68,14%
CONEXIONES ESTABLECIDAS SERVIDOR	584	15,89 %	503	15,42 %	568	15,22 %	1312	16,20 %	1298	16,79%
PAQUETES TCP RECIBIDOS	9840068	--	9971475	--	4500873	--	23070501	--	45528824	--
PAQUETES TCP TRANSMITIDOS	18248806	--	19919486	--	17983257	--	92182232	--	90921803	--

Realizado por: Aguilar R. 2016

Al observar los datos ubicados en las tablas podemos deducir que el porcentaje de usuarios que pudieron establecer una conexión con los servidores se encuentra en una media del 65% aproximadamente, lo que quiere decir, que cerca del 35% de las solicitudes no pueden ser

atendidas de forma positiva imposibilitando el acceso por parte de los usuarios a los servicios de la red.

Los días de mayor concentración de conexiones establecidas de cliente y de servidor son el noveno y décimo día del período de matrículas, en los cuales se dio lugar la solicitud de cupos en el Centro de Idiomas.

La Dirección de Tecnologías de la Información y Comunicación cuenta con el licenciamiento de Windows Server 2008, otro punto por el cual optar por la implementación del balanceador por software NLB propiedad del gigante de tecnologías Microsoft.

El funcionamiento de NLB sobre la red de DTIC es direccionado por el método de saturación de servidores, por lo cual el establecimiento de conexiones cliente-servidor adquiere puntos de dificultad conllevando a una solución con efectos mínimos.

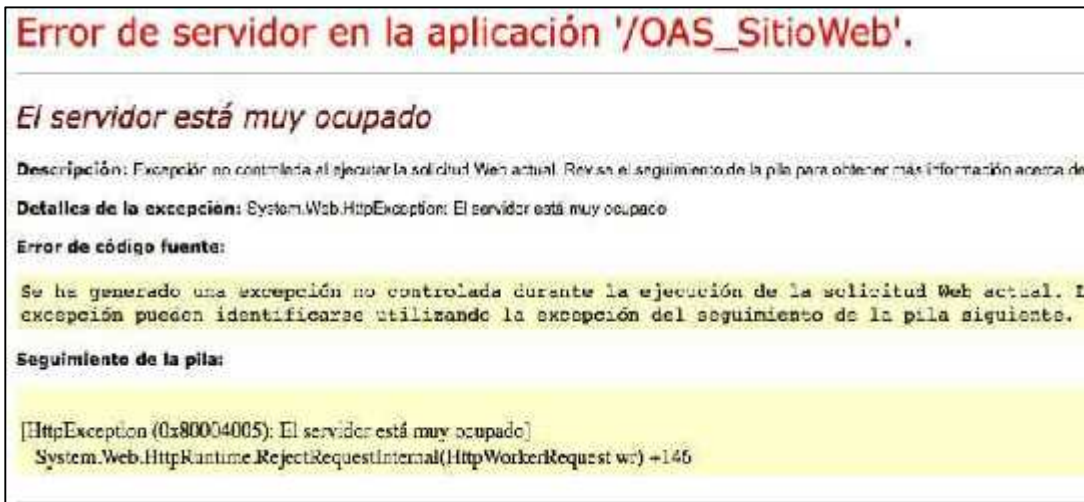
NLB cumple con su la función que le fue encargada, el balanceo lo hace en referencia a la afinidad establecida (Single) escogida porque es la que mejor se acopló a la necesidad proveniente del sistema OASIS de guardar estados en la aplicación Web, de igual manera este balanceador generaba todo el tráfico de peticiones a un servidor hasta que se sature, y cada vez que uno de ellos caía era necesario un nuevo establecimiento de convergencia creando lapsos de desconexión total.

La administración de NLB resulta dificultosa, sobre todo cuando se requiere realizar un análisis del funcionamiento de los servidores a pesar de todo ser virtualizado y administrado vía remota.

El dashboard que presenta no es agradable y es poco funcional, para mejorarlo se debe acudir a softwares de terceros y en cuanto a los datos estadísticos resultaba tedioso el tener que abrir de forma individual el panel de monitoreo en cada servidor.

Para la toma de datos fue necesario recurrir a una estrategia de descubrimiento de vulnerabilidades en la red, el método utilizado fue a través de ataques DDoS; lo podemos encontrar con mayor detalle en el libro “The Browser Hacker’s Handbook” de Wade Alcorn, debido a que el estudio fue realizado en intervalos de tiempo en los que las peticiones se encontraban por lejos del máximo límite para llegar a una saturación de los servidores, con esta técnica se logró inundar la red con las peticiones necesarias y comprobar el funcionamiento de los servidores.

Las prioridades establecidas sobre los servidores en muchos casos no eran respetadas, y la orden de manejo de peticiones eran tomadas de forma aleatoria por los nodos del clúster. Creaban conmutaciones por error entre servidores.



**Figura 1-3.** Servidor caído por saturación.  
Fuente: Sistema Académico



**Figura 2-3.** Servidor activo.  
Fuente: Sistema Académico

### 3.2. Evaluación del funcionamiento por hardware

El comportamiento del balanceador de carga Citrix NetScaler se puede ver resumido en las siguientes tablas obtenidas a través del estudio durante los períodos de mayor inmersión de los usuarios sobre la red (período de matriculación).



**Figura 3-3.** Gráfica de rendimiento de Citrix NetScaler.  
Realizado por: Aguilar R. 2016

**Tabla 3-3. Datos estadísticos comprendidos desde el día uno hasta el día cinco de solución Citrix Netscaler.**

	día 1 de matrículas		día 2 de matrículas		día 3 de matrículas		día 4 de matrículas		día 5 de matrículas	
	Número	%	Número	%	Número	%	Número	%	Número	%
USO DE ANCHO DE BANDA	[1-2] (Mbps)	--	[4-5] (Mbps)	--	[4-5] (Mbps)	--	[4-6] (Mbps)	--	[4-6] (Mbps)	--
CONEXIONES ESTABLECIDAS CLIENTE	1823	98,90 %	4869	95,49 %	4899	98,74 %	5438	97,85 %	5309	98,76%
CONEXIONES ESTABLECIDAS SERVIDOR	612	30,03 %	1528	29,70 %	1511	32,84 %	1485	31,40 %	1486	33,81%
PAQUETES TCP RECIBIDOS	15480998	--	25480998	--	26480998	--	31726354	--	2958365	--
PAQUETES TCP TRANSMITIDOS	28757870	--	41757870	--	48757452	--	54879684	--	5106814	--

Realizado por: Aguilar R. 2016

**Tabla 4-3. Datos estadísticos comprendidos desde el día seis hasta el día diez de solución Citrix Netscaler.**

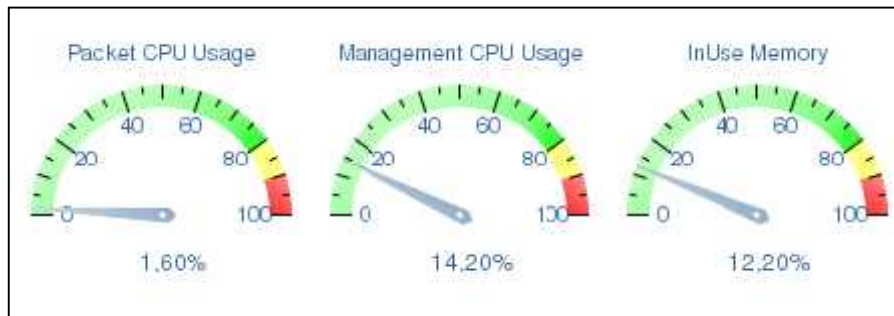
	día 6 de matrículas		día 7 de matrículas		día 8 de matrículas		día 9 de matrículas		día 10 de matrículas	
	Número	%	Número	%	Número	%	Número	%	Número	%
USO DE ANCHO DE BANDA	[4-5] (Mbps)	--	[4-5] (Mbps)	--	[18-24] (Mbps)	--	[19-26] (Mbps)	--	[4-6] (Mbps)	--
CONEXIONES ESTABLECIDAS CLIENTE	2022	98,84 %	2207	98,94 %	10069	95,48 %	13382	94,68 %	3390	98,87 %
CONEXIONES ESTABLECIDAS SERVIDOR	986	32,34 %	1875	29,87 %	3752	28,69 %	8068	28,42 %	1242	31,36 %
PAQUETES TCP RECIBIDOS	19849532	--	19958920	--	91058615	--	121019603	--	30657335	--
PAQUETES TCP TRANSMITIDOS	36512582	--	39858235	--	181845296	--	241677798	--	61223116	--

Realizado por: Aguilar R. 2016

En las tablas 1-3 y 2-3 se puede apreciar el uso realizado del ancho de banda por parte de los clientes, además podemos observar tanto el número de conexiones establecidas por los clientes y número de conexiones establecidas por los servidores, obteniendo los mayores de conexiones los días octavo y noveno de la recolección de datos, en dichos días se llegaron a tener conexiones de hasta un máximo de 10069 en el octavo día y 13382 usuarios en el noveno día, ambos corresponden al período de obtención de cupos en el Centro de Idiomas. Así mismo podemos ver en las tablas el comportamiento de la recepción y transmisión de paquetes a través del balanceador de carga.

Citrix posee un panel administrativo que se maneja de forma sencilla, se convierte en un aliado para el técnico de monitoreo al facilitar herramientas de configuración y análisis sobre el rendimiento tanto del equipo como de la red, el conocimiento del comportamiento de una red permite la proyección de posibles amenazas y detectar la aparición de vulnerabilidades.

El uso de los recursos del dispositivo Citrix a pesar de la alta demanda de distribución de carga generada por las miles de peticiones y conexiones fue de un 14% aproximadamente, se evidencia un trabajo para nada forzado del balanceador (figura).



**Figura 4-3.** Porcentaje uso de recurso durante trabajo Citrix Netscaler.  
Realizado por: Aguilar R. 2016

### 3.3. Análisis comparativo de las soluciones utilizadas

**Tabla 5-3. Cuadro comparativo NLB vs Citrix NetScaler.**

	NLB	CITRIX NETSCALER
Tipo de solución	Software	Hardware
Capacidad manejo peticiones	8Gb	8Gb
Modo de difusión configurado	Unicast	Unicast
Límite de servidores por clúster (Escalabilidad)	32	Sin límite
Dashboard de resultados e informes	Complejo de usar y poco funcional	Intuitivo y fácil de usar
Complejidad de configuración	Intermedio	Fácil
Detección de fallos	X	Si, a través de su ventana de monitoreo
Protección contra DoS	X	Si

Métodos o algoritmos	- Método de distribución por saturación de servidor.	<ul style="list-style-type: none"> <li>- Least Connection Method</li> <li>- Using Weights with the Least Connection Method</li> <li>- Round Robin Method</li> <li>- Least Response Time Method</li> <li>- Using Weights with the Least Response Time Method</li> <li>- Least Response Time Method with Monitors</li> <li>- Using Weights with the Least Response Time Method</li> <li>- Hash Methods</li> </ul>
Uso de ancho de banda promedio	11.55 Mbps	9.35 Mbps
Número máximo de conexiones establecidas clientes	4315	13382
Número máximo de conexiones establecidas servidor	1312	8068
Número máximo de paquetes TCP recibidos	45528824	121019603
Número máximo de paquetes TCP transmitidos	90921803	241677798

**Realizado por:** Aguilar R. 2016

En la Tabla 5-3 figuran tanto características externas e internas como el cuadro comparativo de los parámetros evaluados durante el estudio de las soluciones. La primera diferencia y punto a favor de Citrix es que éste último se encuentra desarrollado sobre una robusta plataforma de hardware proveyendo el soporte más que necesario para actividades de alto rendimiento, mientras que el balanceador NLB totalmente opuesto se encuentra implementado en software volviéndolo adecuado para redes con requerimientos pequeños.

La capacidad de memoria asignado en NLB para el manejo de peticiones es de 8Gb frente a 8Gb que provee Citrix, otro punto para enfrentar una mejor solución de balanceo de carga.



Una amplia ventaja es visible en el factor de escalabilidad, el límite permisivo por parte de NLB de conexiones de servidor por clúster es de 32 servidores mientras que en Citrix este límite no existe adquiriendo la posibilidad de incluir servidores como bien sea necesario.

Método o algoritmos de conexión que ofrece cada uno, como se puede observar la única modalidad de NLB es la de distribución por saturación de servidor imposibilitando una mejor adaptabilidad del servicio de balanceo a la red, NetScaler posee un gran abanico de alternativas desde la más simple Round Robin hasta algoritmos hash que incluyen características de distribución adaptables y más inteligentes.

En el uso del ancho de banda por parte de las soluciones, encontramos que el que genera mayor uso en NLB con un promedio de 11,55 Mbps esto en conjunto con el número de conexiones establecidas tanto por cliente como por servidor demuestra la pobre eficacia para proporcionar el servicio solicitado, el ancho de banda utilizado por Citrix en promedio es de 9,35 Mbps y si es comparado con el número de conexiones establecidas es notable que la eficiencia del uso de tan preciado recurso es alta y significativa.

En las dependencias de DTIC se encuentran dos equipos CITRIX NETSCALER comprados por la institución con la finalidad de mantener uno activo y el otro en estado pasivo como backup en caso de problemas con el principal (configuración en alta disponibilidad), adicional a ello como política de protección de clientes la empresa distribuidora de los equipos Citrix dio un equipo más a la ESPOCH sin ningún costo el mismo que servirá para potencializar la central.

En la etapa de recolección de datos para el balanceo por hardware, se lo realizó en días con mayor tiempo de utilización del sistema académico, lo que permitió que las peticiones sean atendidas por NetScaler con el método Source IP Hash, las solicitudes no fueron suficientes para provocar las saturaciones del balanceador debido a que el sistema OASIS es una aplicación interactiva que basa su conexión en la creación de sesiones a través de autenticación de usuarios. Una vez conectado el usuario puede realizar varias tareas conservando la sesión que posee activa, si la persistencia no se encuentra configurada el usuario podría provocar que las peticiones que eran dirigidas para el servidor 1 lleguen al servidor 2 y su solicitud se vería negada por la aplicación, por ello el parámetro de persistencia COOKIE INSERT permite mantener la sesión y redirigir los paquetes al servidor adecuado característica fundamental que carece NLB.

El balanceador NetScaler pudo mantener activas todas las sesiones y sostuvo con éxito el tráfico generado sin dificultades evidenciando como máximo un uso aproximado del 43% de su memoria RAM equilibrando los recursos de forma correcta.

## CONCLUSIONES

La infraestructura sobre la cual se desenvuelve la red de la Dirección de Tecnologías de la Información y Comunicación ha sido partícipe de cambios progresivos. Una actualización inminente y adaptación de nuevas herramientas ha marcado que el nivel de rendimiento, calidad de servicio y soporte haya ido en alza: Los estudios previos del equilibrio de carga en la data center permitieron tomar las decisiones más acertadas sobre el balanceo adecuado en la red de la ESPOCH.

El servicio más utilizado en la red, es el Sistema Académico OASIS, cuando se inicia un nuevo período académico, lo cual conlleva demora en el acceso por lo que la red se ve saturada. Para solucionar este problema se eligió el algoritmo adecuado que se acople a la red, mejorando la administración de peticiones de los usuarios y estableciendo una conexión segura sin perder el inicio de sesión al sistema.

Se aplicaron encuestas a los usuarios de mayor uso de la red de la ESPOCH, donde se demuestran que el sistema mejoró en comparación con el período educativo anterior. La mayoría de los encuestados manifestaron notar esta diferencia en el tiempo de acceso a aplicación de matriculación en la red.

El estudio comparativo fue determinante y eficiente al demostrar que la mejor solución para realizar el balanceo de carga es por hardware por los requerimientos y características que la red demanda; así como el número de usuarios y actividades iniciadas en la red solicitan un nodo adicional donde se aprovechó el uso eficiente de la licencia de Windows server 2008 aumentando así la memoria RAM a 8 GB permitiendo el acceso de más peticiones por parte de los usuarios de la red.

Las características y parámetros analizados a través de la comparación y análisis se determinaron que la interfaz del Citrix se convirtió en la elección adecuada para la red, mejorando las denegaciones de peticiones generadas por los usuarios innecesarios.

Mediante el uso del dispositivo Citrix se mantiene el inicio de sesiones en estado de espera de forma autónoma hasta que la base de datos vuelva a re direccionar las solicitudes pendientes de los usuarios.

EL algoritmo aplicado al NLB demostró que el equilibrio de carga en la red de la aplicación OASIS fue óptima en días normales. Al momento en que los servidores se saturan debido a las

denegaciones de peticiones generadas por los usuarios al momento de ingresar a la aplicación surgen pérdida de inicio de sesión, pérdida de tiempo y peticiones al servidor en el momento que se normalicen.

Citrix Netscaler demostró soportar las todas conexiones establecidas se obtienen en el tiempo de matrículas obteniendo un número total de 13382 conexiones de usuarios, siendo así la mejor solución de balanceo de carga para el Sistema Académico de la ESPOCH con una clara diferencia del 30% de conexiones establecidas el balanceo por software.

## **RECOMENDACIONES**

Contar con la aprobación y autorización del personal responsable a cargo del DTIC para continuar con el análisis y estudio de problemas reales que detecten debilidades en la red y así brindar soluciones que optimicen su funcionamiento.

Para el NLB, se debe tomar en cuenta la versión y tipo del sistema operativo no siendo la versión de 32bits, debido a que no se puede aumentar a más de 4Gb de memoria RAM a cada servidor.

Sería indispensable utilizar el NLB en ambientes en donde no exista gran cantidad de usuarios, debido a que es una aplicación web que depende de varios estados al momento de ingresar. Cabe indicar también que la utilización de los recursos brinda un mejor funcionamiento y gestión en la red.

La configuración del balanceador de carga, sea por hardware o por software debe ser realizada y supervisada por personal completamente capacitado para aprovechar al máximo los recursos que nos ofrecen las dos soluciones.

El problema radica en la arquitectura de las bases de datos, las cuales no soportan un gran número de peticiones generadas por los servidores. Por lo que se recomienda realizar un balanceo de carga por medio de clonación de la base de datos y sincronizarlas para establecer un manejo adecuado de solicitudes de la aplicación y evitar así colisión o caída del sistema mediante el dispositivo Citrix.

## BIBLIOGRAFÍA

**BARRERA, Manuel.** *Mitigación de DDOS Mediante una Técnica de Minería de Datos Usando Ambientes Virtuales en Linux.* [en línea] (tesis). (Maestría) Instituto Tecnológico y de Estudios Superiores Monterrey. 2007. pp. 25-29. [Consulta: 10 agosto 2016]. Disponible en: [https://repositorio.itesm.mx/ortec/bitstream/11285/567905/1/DocsTec\\_5446.pdf](https://repositorio.itesm.mx/ortec/bitstream/11285/567905/1/DocsTec_5446.pdf)

**BERNAL, Iván, MEJÍA, David & FERNÁNDEZ, Diego.** *Computación de Alto Rendimiento con Clústers de PCs* [En Línea] Ecuador: Escuela Politécnica Nacional, 2012. [Consulta: 2 agosto 2016]. pp.4-10. Disponible en: <http://clusterfie.epn.edu.ec/clusters/Publicaciones/Download/ComputacionAltoRendimiento.pdf>

**BUSTOS, Andrés.** *Configuración de un clúster de alta disponibilidad y balanceo de carga en Linux para satisfacer gran demanda web y servicios de resolución de nombres.* [en línea] (tesis). Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica, Escuela de Ingeniería. Quito-Ecuador. 2007. pp. 98-99. [Consulta: 1 octubre 2016]. Disponible en: <http://bibdigital.epn.edu.ec/bitstream/15000/276/1/CD-0691.pdf>

**BUYYA, Rajkumar.** *High performance cluster computing: Architectures and systems (volume 1).* Prentice Hall, Upper SaddleRiver, NJ, USA, 1999, vol. 1, p. 999.

**CÁCERES, Gerson.** *Estrategia de implementación de un clúster de alta disponibilidad de n nodos sobre Linux usando software libre.* [en línea] (tesis). Universidad San Francisco De Quito. Quito-Ecuador. 2012. p. 54. [Consulta: 10 septiembre 2016]. Disponible en: <http://repositorio.usfq.edu.ec/bitstream/23000/1943/1/104087.pdf>

**CHUQUIGUANCA, Leonardo, et al.** *Arquitectura Clúster de Alto Rendimiento Utilizando Herramientas de Software Libre.* [en línea]. Latin American Journal of Computing Faculty of Systems Engineering National Polytechnic School. Quito-Ecuador. 2015. pp. 3-7. [Consulta: 3 octubre 2016]. Disponible en: <http://lajc.epn.edu.ec/index.php/LAJC/article/download/59/44>

**CITRIX SYSTEMS, INC.** *Load Balancing - Citrix.* [en línea]. How does load balancing work?. [Consulta: 20 septiembre 2016]. Disponible en: <https://www.citrix.com/glossary/load-balancing.html>

**CITRIX SYSTEMS, INC.** *NetScaler ADC - Controlador de entrega de aplicaciones, equilibrador de carga L4-7, GSLB.* [en línea]. NetScaler ADC. [Consulta: 05 octubre 2016]. Disponible en: <https://www.citrix.es/products/netscaler-adc>

**DÍAZ, Adolfo & OCHOA, Anderson.** *Implementación de un prototipo de plataforma tecnológica tipo clúster, para la computación de algoritmos realizados en mpi presentados en el campus universitario.* [en línea] (tesis). Universidad Tecnológica De Pereira, Ingeniería en Sistemas y Computación. Pereira-Colombia. 2015. pp. 35-37. [Consulta: 1 septiembre 2016]. Disponible en: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/6301/00538D542.pdf>

**FLAVIO, Mauricio & GALLARDO, Padilla.** *Diseño de una solución para servidores de alta disponibilidad y balanceo de carga con open Source.* [en línea] (tesis). Universidad Alfredo Pérez Guerrero. Quito-Ecuador. 2011. pp. 3-10. [Consulta: 5 octubre 2016]. Disponible en: <https://es.scribd.com/doc/57937293/Balanceo-de-Carga-en-Centos>

**GÓMEZ, Iliana,** *Clusters de Alto Rendimiento,* [en línea]. Universidad de Guadalajara, México. 2004. [Consulta: 10 octubre 2016]. pp.6-13. Disponible en: [http://cgti.udg.mx/sites/default/files/adjuntos/cluster\\_alto\\_rendimiento2.pdf](http://cgti.udg.mx/sites/default/files/adjuntos/cluster_alto_rendimiento2.pdf)

**INDACOCHEA, Gustavo & MERIZALDE, Galo.** *Estudio y diseño de la tecnología de información y comunicación e-gobierno para el muy ilustre municipio de Portoviejo.* [en línea] (tesis). Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica, Escuela de Ingeniería. Quito-Ecuador. 2004. p. 230. [Consulta: 11 agosto 2016]. Disponible en: <http://bibdigital.epn.edu.ec/bitstream/15000/5411/1/T2259.pdf>

**MICROSOFT.** *Overview of Network Load Balancing.* [en línea]. Overview of Network Load Balancing. [Consulta: 20 octubre 2016]. Disponible en: [https://technet.microsoft.com/en-us/library/cc725691\(v=ws.11\).aspx](https://technet.microsoft.com/en-us/library/cc725691(v=ws.11).aspx)

**MÚNERA, Andrea.** *Método para el manejo del balanceo de carga en sistemas de cómputo distribuido de alto desempeño.* [en línea] (tesis). (Maestría). Universidad Nacional De Colombia, Facultad de Minas, Escuela de Sistemas. Medellín-Colombia. 2009. pp.11-13. [Consulta: 8 octubre 2016]. Disponible en: [http://www.bdigital.unal.edu.co/3497/1/32296964.2009\\_1.pdf](http://www.bdigital.unal.edu.co/3497/1/32296964.2009_1.pdf)

**PINTO Aquilino, PRECIADO Alejandra**, *Arquitectura: Clusters*. [en línea], (tesis) (Pregrado) Universidad Simón Bolívar, Venezuela. 2015. [Consulta: 1 octubre 2016]. pp.10-17. Disponible en: <http://ldc.usb.ve/~yudith/docencia/ci-4821/Temas/2015/Clusters.pdf>

**TEIXEIRA Alejandro**. *BALANCEO DE CARGA*. [en línea]. Mikrotik, Santiago-Chile. 2016. [Consulta: 30 septiembre 2016]. pp.5-12. Disponible en: [http://mum.mikrotik.com/presentations/CL16/presentation\\_3125\\_1456819785.pdf](http://mum.mikrotik.com/presentations/CL16/presentation_3125_1456819785.pdf)

## ANEXOS

### ANEXO A. Resultados de la encuesta

#### PREGUNTA 1: ¿Con qué rapidez logro hacer su pre-matricula?

Pregunta 1		
rápida	65%	130
2 intentos	25%	50
3 intentos	7%	14
4+ intentos	3%	6
total		200



#### PREGUNTA 2: ¿Elija en que pestaña Selecciono para realizar su pre-matricula?

Pregunta 2		
Pestaña 1	30%	61
Pestaña 2	70%	139
Total		200





**PREGUNTA 3: ¿Existió algún problema al momento de generar la Pre-Matricula a través del Internet?**

Pregunta 3		
si	26%	52
no	74%	148
Total		200

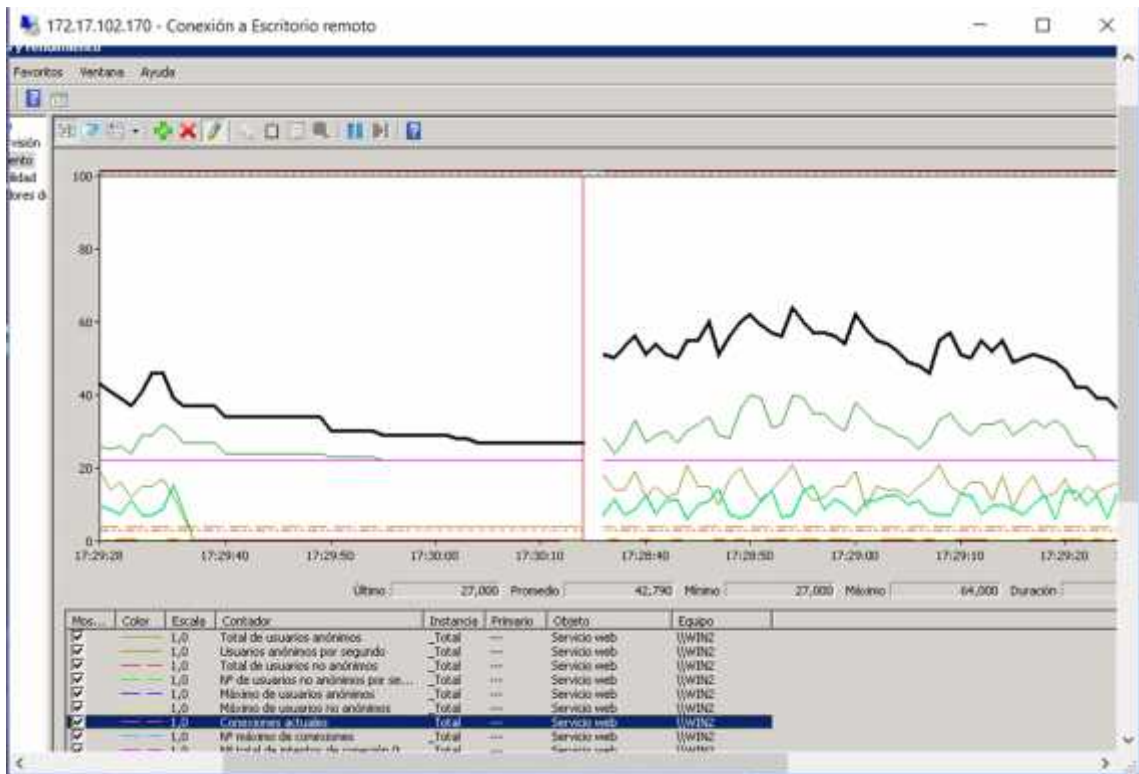
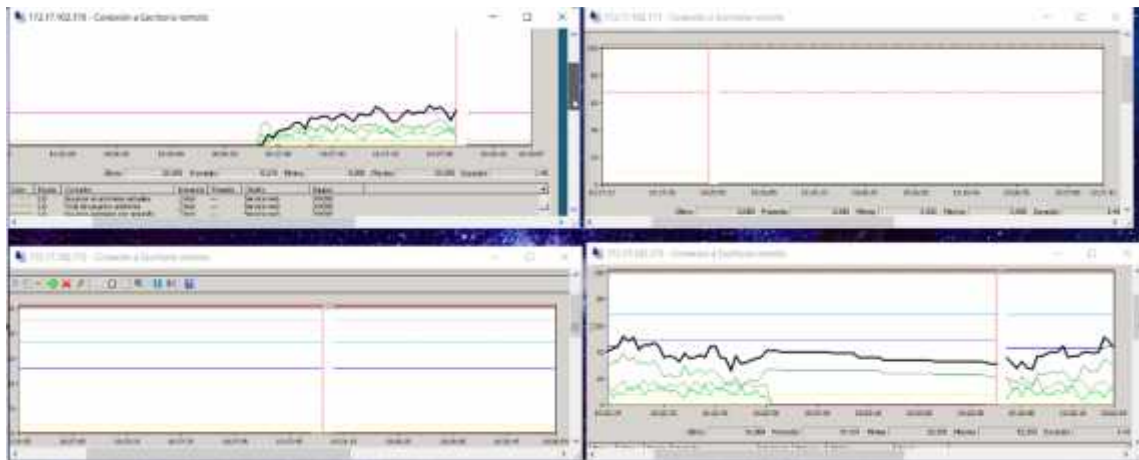


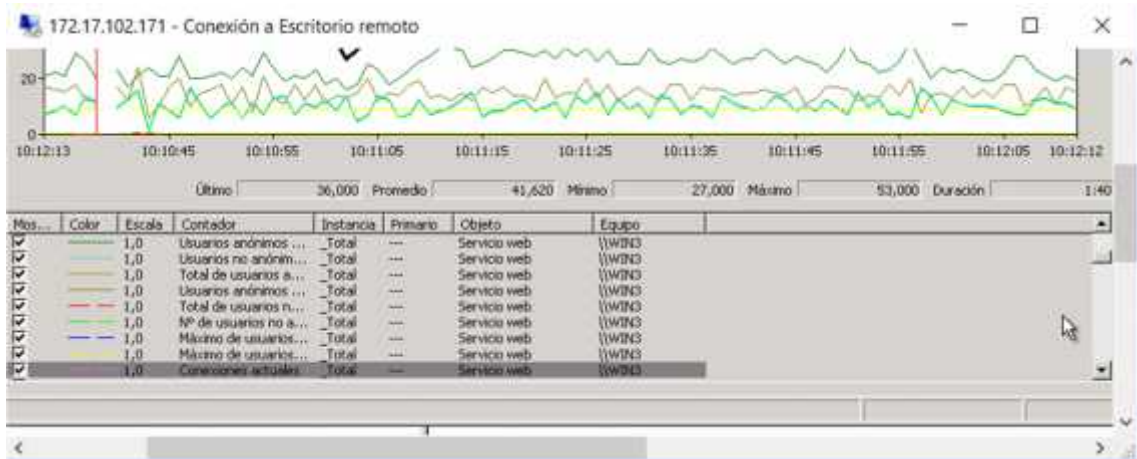
**PREGUNTA 4: ¿Si te matriculaste en 3er, 4to, 5to, 6to, 7mo, 8vo, 9no, 10mo semestre que tal te pareció la rapidez con que realizaste tu pre-matricula a las anteriores del año 2015?**

Pregunta 4		
muy bueno	42%	83
se nota el cambio	22%	44
no noto la diferencia	19%	37
peor que antes	0%	0
no pertenezco a ese periodo	18%	36
Total		200



## ANEXO B. Capturas de pantalla del tráfico balanceado por NLB





Administrador de equipos de carga de red

Inicio | Configuración | Herramientas | Ayuda

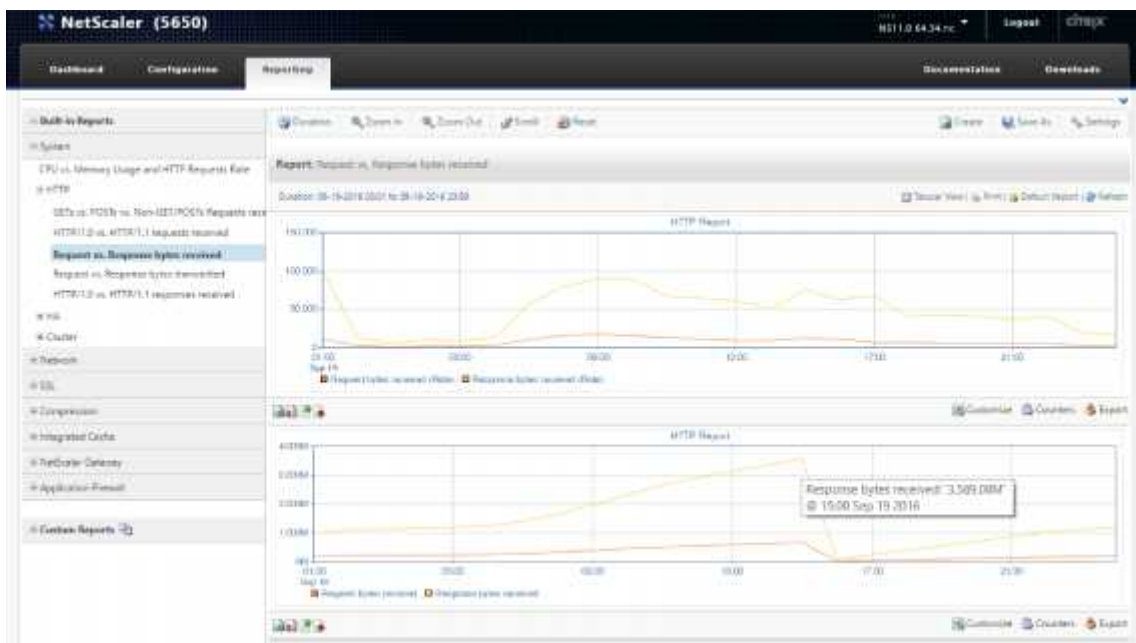
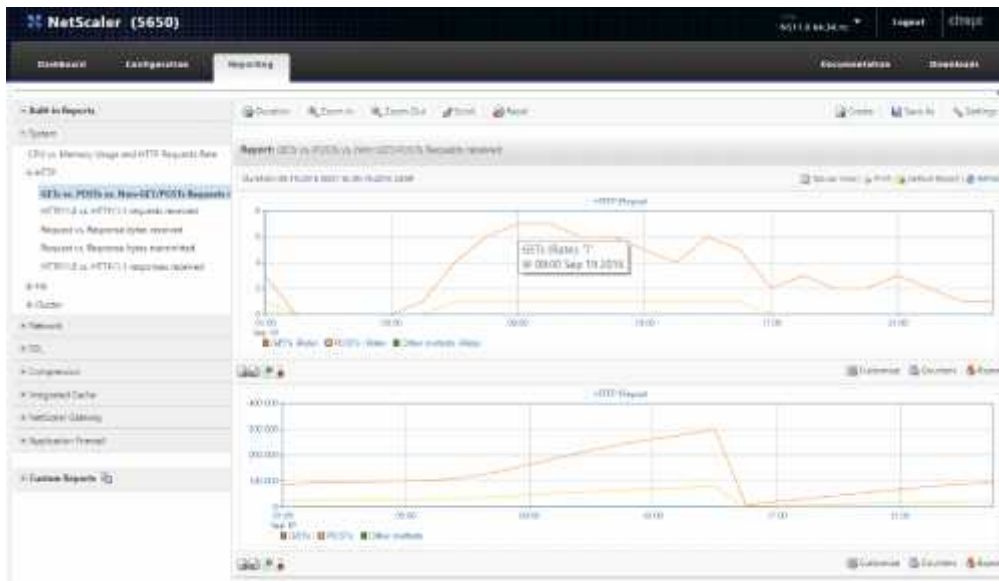
Administrador de equipos de carga de red

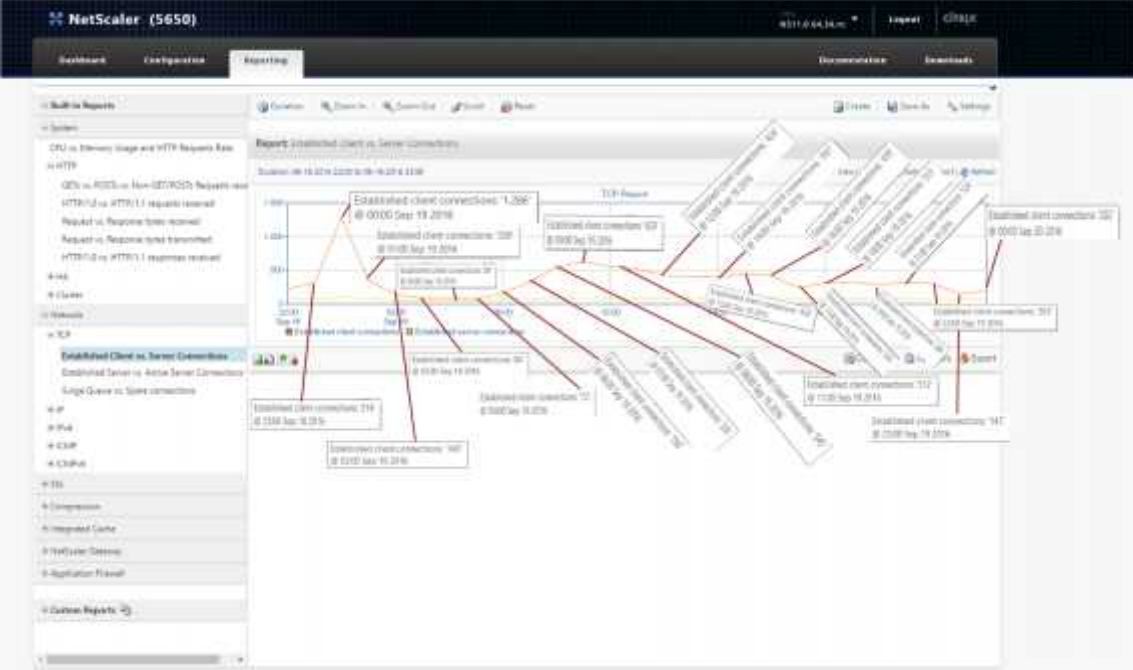
- Administrador de equipos de carga de red
- Administrador de equipos de carga de red
- Administrador de equipos de carga de red
- Administrador de equipos de carga de red
- Administrador de equipos de carga de red

Panel de puertos configurados en VM3 (Conexión de área local 2)

Nombre IP del equipo	Estado	Host	IP	Protocolo	Modo	Protocolo	Carga	Alimentación
Todo	El host es ...	0	0000	Autón	Múltiple	-	Ignor	Desactivado

## ANEXO C. Capturas de pantalla del tráfico balanceado por Citrix Netscaler







**NetScaler (5650)**

**Reports**

IP	Conn	Conn	IP Addr	IP Addr
10.1.1.1	1	1	10.1.1.1	10.1.1.1
10.1.1.2	1	1	10.1.1.2	10.1.1.2
10.1.1.3	1	1	10.1.1.3	10.1.1.3
10.1.1.4	1	1	10.1.1.4	10.1.1.4
10.1.1.5	1	1	10.1.1.5	10.1.1.5

```

Transaction:      200 hits
Availability:    100.00 %
Elapsed time:    14.00 secs
Data transferred: 1.33 MB
Response time:  0.01 secs
Transaction rate: 14.28 trans/sec
Throughput:     0.04 MB/sec
Concurrency:    0.10
Successful transactions: 200
Failed transactions: 0
Longest transaction: 0.04
Shortest transaction: 0.00

PTUI: /var/log/sleep.log
You can disable this annoying message by editing
the _sleepd file in your home directory; change
the directive 'show-logfile' to false.
[error] Unable to write to file: /var/log/sleep.log; Permission denied
riscar@06a121:/#
riscar@06a121:/#
riscar@06a121:/#
riscar@06a121:/# sleep -y -l -s 10 -r 10 -l http://172.17.102.140/alive.cgi
riscar@06a121:/#
  
```

**NetScaler (5650)**

**Reports**

Throughput: 1 Mbps / 1 Mbps

HTTP Requests: 20

System Overview

System Status

Item	Value	Default
Pool CPU usage (%)	100	100
Web server CPU usage (%)	100	100
Memory usage (%)	100	100
Web Server CPU	100	100

System Status

Item	Value	Default
Pool CPU usage (%)	100	100
Web server CPU usage (%)	100	100
Memory usage (%)	100	100
Web Server CPU	100	100

```

Transaction:      200 hits
Availability:    100.00 %
Elapsed time:    14.15 secs
Data transferred: 1.33 MB
Response time:  0.01 secs
Transaction rate: 14.14 trans/sec
Throughput:     0.04 MB/sec
Concurrency:    0.10
Successful transactions: 200
Failed transactions: 0
Longest transaction: 0.04
Shortest transaction: 0.00

PTUI: /var/log/sleep.log
You can disable this annoying message by editing
the _sleepd file in your home directory; change
the directive 'show-logfile' to false.
[error] Unable to write to file: /var/log/sleep.log; Permission denied
riscar@06a121:/#
riscar@06a121:/#
riscar@06a121:/#
riscar@06a121:/# sleep -y -l -s 10 -r 10 -l http://172.17.102.140/alive.cgi
riscar@06a121:/#
  
```

```
root@kali:~# curl -i http://192.168.1.100/
HTTP/1.1 200 OK
Content-Type: text/html
Date: Wed, 07 Jun 2017 19:30:45 GMT
Server: Apache/2.4.18 (Ubuntu)
Transfer-Encoding: chunked

<!DOCTYPE html>
<html>
<head>
<title>Ubuntu</title>
</head>
<body>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 300 860 330" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 270 860 300" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
</body>
</html>

root@kali:~# curl -i http://192.168.1.100/
HTTP/1.1 200 OK
Content-Type: text/html
Date: Wed, 07 Jun 2017 19:30:45 GMT
Server: Apache/2.4.18 (Ubuntu)
Transfer-Encoding: chunked

<!DOCTYPE html>
<html>
<head>
<title>Ubuntu</title>
</head>
<body>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 240 860 270" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 210 860 240" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
</body>
</html>

root@kali:~# curl -i http://192.168.1.100/
HTTP/1.1 200 OK
Content-Type: text/html
Date: Wed, 07 Jun 2017 19:30:45 GMT
Server: Apache/2.4.18 (Ubuntu)
Transfer-Encoding: chunked

<!DOCTYPE html>
<html>
<head>
<title>Ubuntu</title>
</head>
<body>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 180 860 210" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 150 860 180" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
</body>
</html>

root@kali:~# curl -i http://192.168.1.100/
HTTP/1.1 200 OK
Content-Type: text/html
Date: Wed, 07 Jun 2017 19:30:45 GMT
Server: Apache/2.4.18 (Ubuntu)
Transfer-Encoding: chunked

<!DOCTYPE html>
<html>
<head>
<title>Ubuntu</title>
</head>
<body>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 120 860 150" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 90 860 120" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
</body>
</html>

root@kali:~#
```

```
root@kali:~# cat /dev/null > /var/log/sleaze.log
root@kali:~# curl -i http://192.168.1.100/
HTTP/1.1 200 OK
Content-Type: text/html
Date: Wed, 07 Jun 2017 19:30:45 GMT
Server: Apache/2.4.18 (Ubuntu)
Transfer-Encoding: chunked

<!DOCTYPE html>
<html>
<head>
<title>Ubuntu</title>
</head>
<body>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 60 860 90" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
<div style="background-color: #e6f2ff; padding: 5px 0 0 0; margin-bottom: 10px; border: 1px solid #cfe2f3; border-radius: 4px 0 0 0;">
<div style="float: right; text-align: right; padding: 5px 10px 0 0;">
<img alt="Ubuntu logo" data-bbox="830 30 860 60" style="vertical-align: middle;"/>
<span style="font-size: 0.8em; vertical-align: middle; padding-left: 5px;">
Ubuntu
</span>
</div>
</div>
</body>
</html>

root@kali:~#
```