



## DISEÑO DE UN CLÚSTER ORIENTADO A SERVICIOS PARA APLICACIONES WEB EN LA UNIVERSIDAD NACIONAL EXPERIMENTAL DEL TÁCHIRA

(Design of a cluster oriented to web applications services in the Universidad  
Nacional Experimental del Táchira)

Recibido: 12/09/2014 Aceptado: 11/11/2014

**Mireles, Jhonangel**

Universidad Nacional Experimental del Táchira, Venezuela

[jhonangelmireles@gmail.com](mailto:jhonangelmireles@gmail.com)

**Maldonado, Javier**

Universidad Nacional Experimental del Táchira, Venezuela

[jmaldo@unet.edu.ve](mailto:jmaldo@unet.edu.ve)

### RESUMEN

La Universidad Nacional Experimental del Táchira (UNET) es una institución académica que viene generando valiosa formación en el campo de la ingeniería, incentivando a sus estudiantes en el área de la informática a realizar investigaciones y desarrollos que se adecuen al contexto tecnológico de la actualidad. En lo que respecta al despliegue de aplicaciones en entornos web en la UNET, se observa con frecuencia el uso de componentes de hardware y software especializados, sin embargo, existen tecnologías alternas como el clúster que ofrece características como disponibilidad, fiabilidad y arquitecturas fácilmente escalables, en las cuales otras tecnologías son deficientes o tienen carencia. La investigación tiene por finalidad el diseño de un clúster orientado a servicios para aplicaciones web en la Universidad Nacional Experimental del Táchira. En aras de lograr el objetivo planteado, se emplearon metodologías cuantitativas y se aplicaron una serie de casos en un ambiente de prueba; así como también, se analizaron ciertas métricas sobre el clúster formado por un grupo de nodos de base de datos y un grupo de nodos servidores web, en una arquitectura orientada a la alta disponibilidad. Los valores obtenidos demostraron que existen métricas en las que la utilización del clúster se traduce en mejores resultados.

**Palabras claves:** clúster, sistemas distribuidos, alta disponibilidad.

### ABSTRACT

The Universidad Nacional Experimental del Táchira (UNET) is an academic institution that is providing valuable training in the field of engineering, encouraging its students in the area of computer science engineering to perform research and development that fit the current technological context. Regarding the deployment environments of web applications in the UNET, is frequently observed the use of specialized hardware and software, however, there are alternative technologies such as the cluster that provides features as availability, reliability, and easily scalable architectures, in which other technologies are deficient or have lack. This research aims to design a cluster oriented to web applications services in the Universidad Nacional Experimental del Táchira. In order to achieve the



stated objective, quantitative methodologies were used and a series of cases were applied in a test environment; as well as, certain metrics were analyzed on the cluster. The Infrastructure was composed of a database cluster and a group of web-server nodes in high-availability architecture. The values obtained in this research, showed that there are metrics in which the use of cluster leads to better results.

**Keywords:** cluster, distributed systems, high-availability.

## INTRODUCCIÓN

Las aplicaciones web ofrecen servicios en línea con diferentes propósitos, frecuentemente, haciendo uso de navegadores en los dispositivos del usuario final; estas poseen grandes ventajas con respecto a las aplicaciones desarrolladas para entornos de escritorio, como lo son portabilidad, accesibilidad, desarrollo multiplataforma, bajo uso de recursos por parte del cliente, entre otras; razón por la cual, empresas y organizaciones se han motivado a migrar a entornos web dinámicos en la última década.

No obstante, para el despliegue de este tipo de aplicaciones, se precisan plataformas tecnológicas que brinden soporte a los niveles de carga esperados para los diferentes servicios.

De esta manera, la investigación tuvo como principal propósito de estudio el diseño de un clúster orientado a servicios para dar soporte al despliegue de aplicaciones en entornos web desarrolladas en la Universidad Nacional Experimental del Táchira (UNET), con la finalidad de mejorar su disponibilidad y rendimiento.

Para el logro de los objetivos, la investigación estuvo enmarcada en el uso de metodologías cuantitativas, se realizó una revisión documental, se recolectó información mediante entrevistas, se utilizaron técnicas de observación en el contexto de estudio, se llevaron a cabo diferentes casos en un ambiente de prueba y se identificaron una serie de métricas para comparar el rendimiento y disponibilidad entre el clúster y un servidor.

## EL PROBLEMA

El despliegue de aplicaciones web en la UNET, comúnmente se basa en utilizar un conjunto de servidores independientes especializados, los cuales ofertan sus servicios a la comunidad universitaria. Sin embargo, a pesar de que algunos de estos dispositivos son flexibles y modulares, aumentar la capacidad de ellos puede incurrir en costos elevados, esto supone un límite al generarse la necesidad de escalar los servicios y aplicaciones allí dispuestas.

Otro de los aspectos que se debe tener en cuenta, es el caso en el que se produzca algún tipo de falla en los servidores, al no estar dispuestos en arquitecturas redundantes y tolerantes a fallas, provoca la no disponibilidad de las aplicaciones y caídas en los servicios, al igual que pérdida de información o datos, lo que incide negativamente en la confiabilidad de los usuarios de las aplicaciones y puede afectar el flujo de trabajo de la institución.



Un factor clave e indispensable para el éxito de las empresas y microempresas en el área de servicios es contar siempre con aplicaciones que apoyen sus operaciones y procesos de negocios principales, pues la falta de disponibilidad en estas, puede repercutir en costos, tiempo, esfuerzo y en la confianza y grado de satisfacción de sus clientes.

Varias aplicaciones a través de internet implican el tratamiento seguro, de cálculo intensivo, multimedia y altos anchos de banda para información. Muchas de estas aplicaciones requieren a gran escala transmisión de datos y de alto ancho de banda en los nodos de servidor. Sin embargo, al tratar de escalar dichos servidores a altos niveles de disponibilidad, flexibilidad y rendimiento, la necesidad de arquitecturas de software más sofisticadas se convierte en algo obvio. La evolución de las aplicaciones como servicios web, tecnología grid y computación punto a punto (P2P), necesitan aún más el uso software escalable y servidores de gran potencia, lo cual podría llevar a reparar en grandes gastos para comprar hardware y software especializado que dé solución a la demanda requerida (Basu et al., 2005).

Para dar solución a los diferentes problemas identificados, esta investigación propuso el diseño de un clúster orientado a servicios para aplicaciones web con la intención de mejorar su disponibilidad y rendimiento en la UNET.

Se planteó, hacer uso de componentes software libre para la implementación del clúster, entre los cuales destacaron un servidor web de alta disponibilidad y un servidor de base de datos distribuido, redundante y multimaestro.

Como complemento a la propuesta, se planificó la realización de una serie de casos en un ambiente de prueba para verificar la funcionalidad, así como el análisis de determinadas métricas para comparar el rendimiento y la disponibilidad entre el clúster y un servidor independiente.

## DIAGNÓSTICO

En esta fase se realizaron diferentes tareas, con el propósito de diagnosticar la gestión y despliegue de las aplicaciones web desarrolladas en la Universidad Nacional Experimental del Táchira. Para ello, se aplicaron instrumentos y técnicas de recolección de información y datos, como lo son la entrevista y la observación. Se realizó un análisis de los datos recopilados, lo que permitió llegar a una serie de conclusiones y resultados derivados del diagnóstico.

Existieron diferentes posiciones respecto a la problemática y las necesidades presentes en los servicios y aplicaciones en la UNET, la posibilidad de escalar equipos de fácil adquisición, con respecto a servidores especializados fue considerada una ventaja importante.



## IDENTIFICACIÓN LOS COMPONENTES DE SOFTWARE LIBRE PARA EL DISEÑO DEL CLÚSTER

Esta fase constó de la búsqueda y estudio de software libre disponible en la red para el diseño del clúster. Se realizó la comparación y contrastación de aplicaciones de software disponibles cuyas características estuvieran acordes a los objetivos planteados.

Inicialmente, para el diseño del clúster y desarrollo de caso de prueba se seleccionaron las distribuciones Debian y CentOS, ambas basadas en Linux. En la Tabla 1 se da una caracterización del hardware y los sistemas operativos usados para el desarrollo del presente proyecto de investigación:

**Tabla 1. Sistemas Operativos**

Sistema Operativo	Disco Duro	Memoria RAM	Procesador	Tarjetas de Red
Debian 7.4.0, Kernel Linux 3.2.0- 4-amd64. x86_64 GNU/Linux	10 GiB	1024 MiB	Intel® Xeon® CPU E5310@ 1.60 GHz	1 Gbps x 2
CentOS 6.5, Kernel Linux 2.6.32- 431.5.1.el6.x86_64 GNU/Linux	10 GiB	1024 MiB	Intel® Xeon® CPU E5310@ 1.60 GHz	1 Gbps x 2

**Fuente:** elaboración propia.

El servidor web seleccionado para el diseño del clúster fue apache HTTP en su versión 2.2, de la fundación de software Apache, ya que es el más utilizado en Internet, con más de 10 millones de sitios web estaban usándolo en el año 2013, además de ser un software robusto, con muchas características y de libre disposición e implementación del código fuente.

Por otro lado, la base de datos de fuente abierta que posee el primer lugar en popularidad en Internet es MySQL. Particularmente, MySQL Clúster en su versión 7.3.4 fue la seleccionada para el diseño del clúster, ya que es una base de datos transaccional en tiempo real altamente escalable, compatible con ACID, que combina 99,999% de disponibilidad con el bajo costo total de propiedad de código abierto.

De la misma manera, se vio la necesidad de utilizar un software que prestara servicios de Corta Fuegos (Firewall) y Puerta de enlace (Gateway). Para ello, se seleccionó una distribución basada en Linux, tomando en cuenta ciertos criterios de decisión como características para implementar alta disponibilidad, balanceo de carga y los servicios extra que pudieran ser administrados desde el mismo (NTP, DNS, NAT, entre otros).

En el caso de la investigación realizada se seleccionó la distribución firewall Zentyal, que está basada en la distribución Ubuntu, debido a que permite administrar todos los

servicios de red, como acceso a Internet, seguridad de redes, el intercambio de recursos, la infraestructura de red o de comunicaciones de una manera fácil a través de una única plataforma y cumple con la Licencia Pública General de GNU.

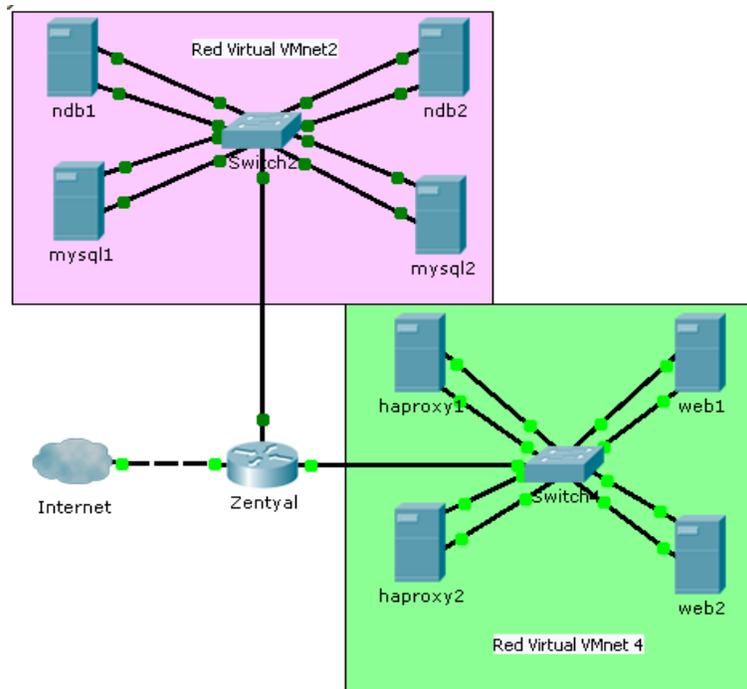
Para el diseño de la arquitectura del clúster, se vio necesario la utilización de un componente de software cuya principal característica fuera el balanceo de tráfico, por ello, se seleccionó HA Proxy específicamente en su versión 1.5, el cual se basa en protocolos HTTP/TCP, lo que permitió su uso para balancear tanto el tráfico web, como el tráfico de la base de datos, de igual forma, esta herramienta posee una larga trayectoria en la red, lo que provee amplia documentación.

### INTEGRACIÓN DE LOS COMPONENTES DE SOFTWARE

El ambiente de pruebas se realizó sobre una plataforma virtualizada, utilizando el producto VMware ESXi.

En esta fase, se diseñó la arquitectura base de red (ver Figura 1) y se aplicaron las configuraciones adecuadas a cada uno de los host que forman parte de la misma, así como se establecieron los servicios de firewall a nivel de kernel iptables y un dominio NTP para sincronización de tiempo de forma local.

**Figura 1. Arquitectura de Red**



**Fuente:** elaboración propia.



Los nombres de las m quinas fueron asignadas con el formato “servicioN”, donde servicio es el nombre del servicio principal que es ejecutado por la m quina, y N es el n mero de m quina que ejecuta dicho servicio.

A continuaci n, se realiza una descripci n de los servicios principales ubicados en cada uno de los equipos:

- Los equipos ndbN representan a un nodo de datos N de MySQL Cl ster, en ellos se ejecuta el demonio ndbmysd (el cual se explica m s adelante).
- En los nodos mysqlN se ejecuta un nodo de administraci n (ndbmgsd), un nodo SQL (mysqld), el servicio haproxy como balanceador de carga entre los nodos SQL, y el servicio Keepalived que mantiene una interfaz virtual a la que ingresar n peticiones de datos por parte de los servidores web.
- En los nodos haproxyN se ejecuta el servicio HA Proxy que funcionan como un proxy inverso, balanceador de carga, y administrador del tr fico HTTPS, para los servidores web y el servicio Keepalived que mantiene la interfaz virtual a la cual la puerta de enlace enviar  las peticiones de tr fico web que provengan de la red p blica.
- Los nodos webN ejecutan un servidor apache HTTP.

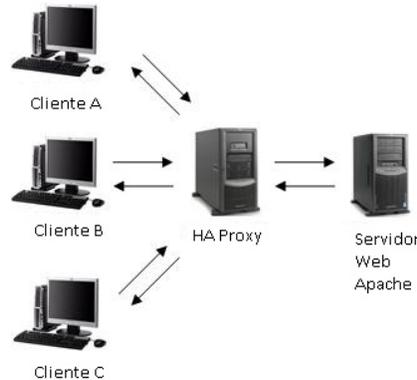
Se llev  a cabo la instalaci n de MySQL Cl ster, en la que hay tres tipos de nodos que formaron parte de la configuraci n b sica del cl ster:

- Nodo de gesti n (ndbmgsd): el papel de este tipo de nodo es administrar los otros nodos dentro del MySQL Cl ster, realizando datos de configuraci n, de inicio y apagado, y para ejecutar copias de seguridad.
- Nodo de datos (ndbd o ndbmysd): este tipo de nodo guarda la data del cl ster.
- Nodo SQL (mysqld): este es un nodo que accede a los datos del cl ster. En el caso de MySQL Cl ster, un nodo de SQL es un servidor MySQL tradicional que usa el motor de almacenamiento NDBCLUSTER.

Tambi n, se instal  y configur  el servidor Web Apache junto con m dulos de PHP para cada una de las distribuciones seleccionadas. As  como se aplic  una serie de medidas b sicas de seguridad tanto para Apache como para PHP, que incluyen evitar la fuga de informaci n hasta ataques DoS, utilizando los m dulos de seguridad ModSecurity y Suhosin.

Se realiz  la instalaci n de los balanceadores de carga con una arquitectura de alta disponibilidad. Haciendo uso de HA Proxy y Keepalived se tuvo como referencia la configuraci n y arquitectura de red, ya definidos. En la Figura 2 se puede observar un ejemplo del funcionamiento del HA Proxy.

**Figura 2. Ejemplo del funcionamiento del Proxy**



**Fuente:** elaboración propia.

Para las pruebas se usó Moodle, aplicación web libre de tipo ambiente educativo virtual, más conocida como un sistema de gestión de cursos que ayuda a educadores a crear comunidades de estudio, aprovechando diferentes herramientas y recursos en línea. Moodle es una plataforma de aprendizaje, diseñado para proporcionar a educadores, administradores y estudiantes un único sistema integrado, robusto y seguro para crear entornos de aprendizaje personalizados.

### **SELECCIÓN DE MÉTRICAS Y DE LAS HERRAMIENTAS DE MEDICIÓN**

Las métricas se constituyen con diferentes propósitos, alguno de estos son establecer una línea base de comportamiento para la resolución de problemas y para estimar y mejorar el crecimiento.

En la Tabla 2, se listan las métricas consideradas pertinentes, para lograr los objetivos propuestos por la presente investigación:

**Tabla 2 . M etricas seleccionadas**

Dimensi�n	Descripci�n	M�etricas
Rendimiento del Sistema	Se caracteriza por la cantidad de trabajo �til realizado por un computador en comparaci�n con el tiempo y los recursos utilizados.	Uso del CPU
		Uso de la Memoria
		Uso del Disco Duro
		Tr�fico de Red
Servidor HTTP	M�tricas en el contexto de los servidores web.	Tasa de Conexiones
		Tasa de Solicitudes
		Tasa de Respuesta
		Tasa de Sesiones
		Flujo de Datos
Servidor de Base de Datos	M�tricas el contexto de los servidores de base de datos.	Transacciones de base de datos
		Conexiones de Base de datos
		Consultas de base de datos

**Fuente:** elaboraci n propia.

Para las pruebas de carga y rendimiento del cl ster de servidores web, se seleccion  la herramienta httpperf en la versi n 0.9.0 (licencia GNU GPLv2), que junto al software Autobench en la versi n 2.1.2 (GNU GPLv2) se extiende la funcionalidad de httpperf para permitir automatizar las pruebas de forma distribuida.

Para las pruebas sobre MySQL Cl ster, se utiliz  una de las herramientas comparativas sugeridas en la documentaci n web de MySQL, esta es Sysbench 0.4.12, la cual es una herramienta de benchmarking modular, multiplataforma y multi-hilo para evaluar par metros del sistema operativo que son importantes para un sistema con una base de datos bajo carga intensiva.

### **APLICACI N DE PRUEBAS**

Los casos de prueba, son un conjunto de condiciones o variables sobre las cuales se determinar  si un elemento, servicio o caracter stica est  funcionando correctamente.

En la Tabla 3 se muestra el conjunto de pruebas aplicado para el cl ster implementado.

**Tabla 3. Conjunto de Pruebas**

Nº	Conjunto de Pruebas	Casos de Prueba	Descripción
1	Pruebas de sobre el clúster de servidores web	Comportamiento del clúster web con dos nodos.	Realizar mediciones de carga y rendimiento del servicio en diferentes ambientes de prueba (Escalabilidad Horizontal).
		Comportamiento del clúster al agregar un nodo web a la topología base.	
		Comportamiento del clúster al existir la caída de un nodo web.	Realizar mediciones de carga, rendimiento y disponibilidad del servicio al existir la falla de un nodo, una caída representa un evento de falla no planificado, que podría derivarse tanto en una falla eléctrica como en una falla a nivel de hardware de alguna interfaz de red.
	Pruebas sobre un Servidor Web	Comportamiento de un servidor web.	Realizar mediciones de carga y rendimiento sobre el servicio web para hacer comparativas.
2	Pruebas sobre el Clúster de servidores de base de datos	Comportamiento del clúster con dos nodos de datos.	Realizar mediciones de carga y rendimiento del servicio en diferentes ambientes de prueba (Escalabilidad Horizontal).
		Comportamiento del clúster al agregar dos nodos de datos a la topología base.	
		Comportamiento del clúster al existir la caída de un nodo de datos.	Realizar mediciones de carga, rendimiento y disponibilidad del servicio al existir la falla de un nodo, una caída representa un evento de falla no planificado, que podría derivarse tanto en una falla eléctrica como en una falla a nivel de hardware de alguna interfaz de red.
	Pruebas sobre un Servidor de Base de Datos	Comportamiento de un servidor de base de datos.	Realizar mediciones de carga y rendimiento sobre el servicio de base de datos para hacer comparativas.

**Fuente:** elaboración propia.

Para llevar a cabo la monitorización de las métricas en los diferentes sistemas operativos que están inmersos en cada una de las pruebas se utilizará la herramienta Pandora FMS 5.0SP4 de licencia GNU GPLv3, ya que esta es usada por el Centro de Estudios de Teleinformática (CETI) de la Universidad Nacional Experimental del Táchira, para la monitorización de ciertos servicios.



Para cada una de las pruebas es requerido definir qu  servicios est n ejecut ndose en el cl ster y los recursos de cada uno de los nodos. Para mantener la consistencia de las pruebas, tanto el cl ster como el generador de carga, est n dispuestos en diferentes equipos y las pruebas son realizadas en una red cerrada.

Se revis  el estado y la capacidad de la red, para saber en qu  momento este factor puede limitar las pruebas. Para lograr esto, se us  la herramienta de rendimiento de red Netperf y Ping.

En los servidores de alto rendimiento es muy f cil confundir las limitaciones del mismo con las de los nodos que llevan a cabo la prueba. Por lo que se asegur  de que las maquinas cliente o generadoras de carga no estuvieran usando el m ximo de recursos para llevar a cabo las pruebas.

### **CONJUNTO DE PRUEBAS N  1: PRUEBAS DE SOBRE EL CL STER DE SERVIDORES WEB**

Se realizaron las mediciones de rendimiento sobre diferentes entornos de pruebas, donde se simulan las solicitudes a p ginas din micas PHP de usuarios a un cl ster de servidores web.

A continuaci n, se muestran los resultados de las mediciones realizadas sobre cuatro diferentes arquitecturas configuradas, cada una de las series de datos utilizadas por las gr ficas es descrita a continuaci n:

- 2 Nodos Web: esta serie, hace referencia al comportamiento del cl ster con dos nodos web.
- 3 Nodos Web: esta serie, hace referencia al comportamiento del cl ster con tres nodos web.
- Falla Nodo: esta serie hace referencia al comportamiento del cl ster al existir la ca da de un nodo web, es decir, de tres nodos web funcionales pasa a solamente dos.
- 1 Servidor: esta serie hace referencia al comportamiento de un servidor web independiente sin el uso de un proxy.

La Figura 3, muestra los resultados ejecutados en t rminos de solicitudes en servidas (Eje Y) por solicitudes demandadas (Eje X).

Se puede ver que el servidor independiente se satura al existir tasas mayores de 400 solicitudes/segundo bajo demanda, a partir de all , las solicitudes del servidor independiente se mantienen entre valores m nimos de 353.9 y m ximos de 459.5.

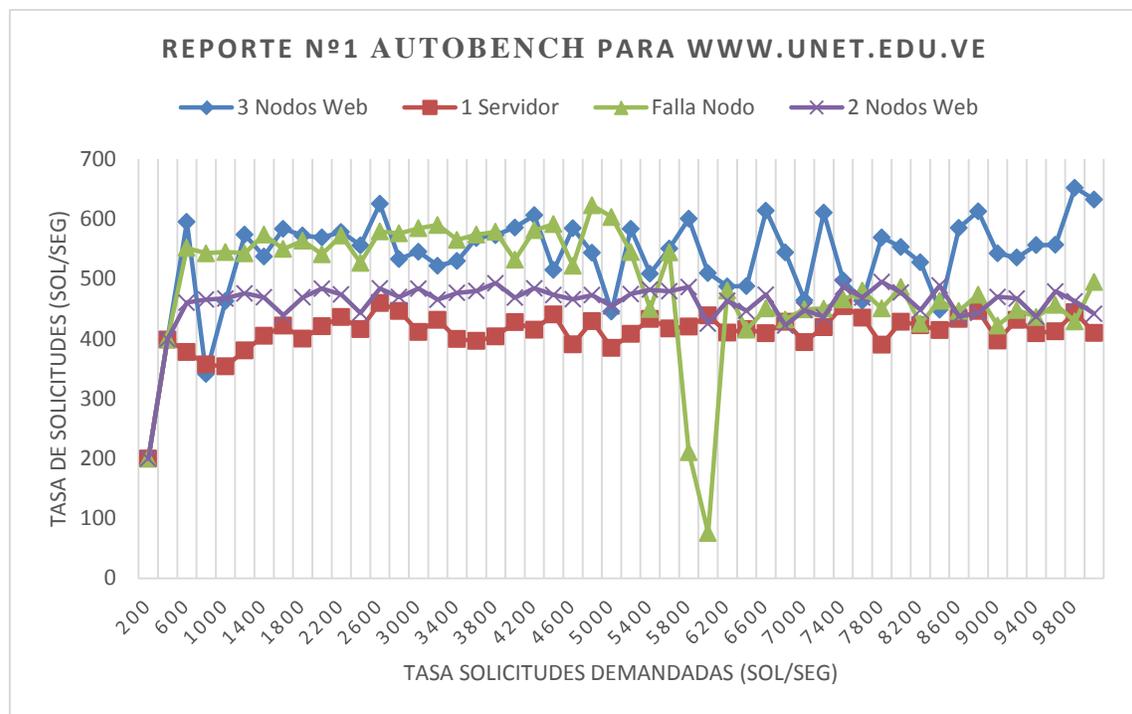
Tambi n se observa que los servidores con dos nodos, 475.7, se satura al existir cargas aproximadas de 1200 solicitudes/segundo. A partir de este punto, mantiene cargas

relativas entre valores de entre 422.1 y 494.8 solicitudes/segundo, a pesar de que la demanda va en aumento.

Para un clúster web, con la arquitectura base propuesta, al tener tres nodos, se observa cómo se satura a partir de 600 solicitudes/segundo, a pesar de tener un punto de saturación menor que un clúster de dos nodos, este clúster tiene tasas más altas de solicitudes servidas a lo largo de la prueba, para variar entre valores mínimos de 341.5 y máximos de 652.2.

Cuando existe la caída de un nodo web, aproximadamente en un momento a una tasa aproximada de 5600 solicitudes/segundo bajo demanda, las solicitudes servidas por el clúster baja momentáneamente hasta un mínimo de 75.6 sol./seg., el rendimiento de número solicitudes baja cerca de un 20%, sin embargo, sigue un desempeño bastante aceptable. Las sesiones que se vean afectadas por la caída de un nodo, serán despachadas a los nodos restantes del clúster por el balanceador de carga.

**Figura 3 . Reporte 1 Autobench (httpperf)**



**Fuente:** elaboración propia.

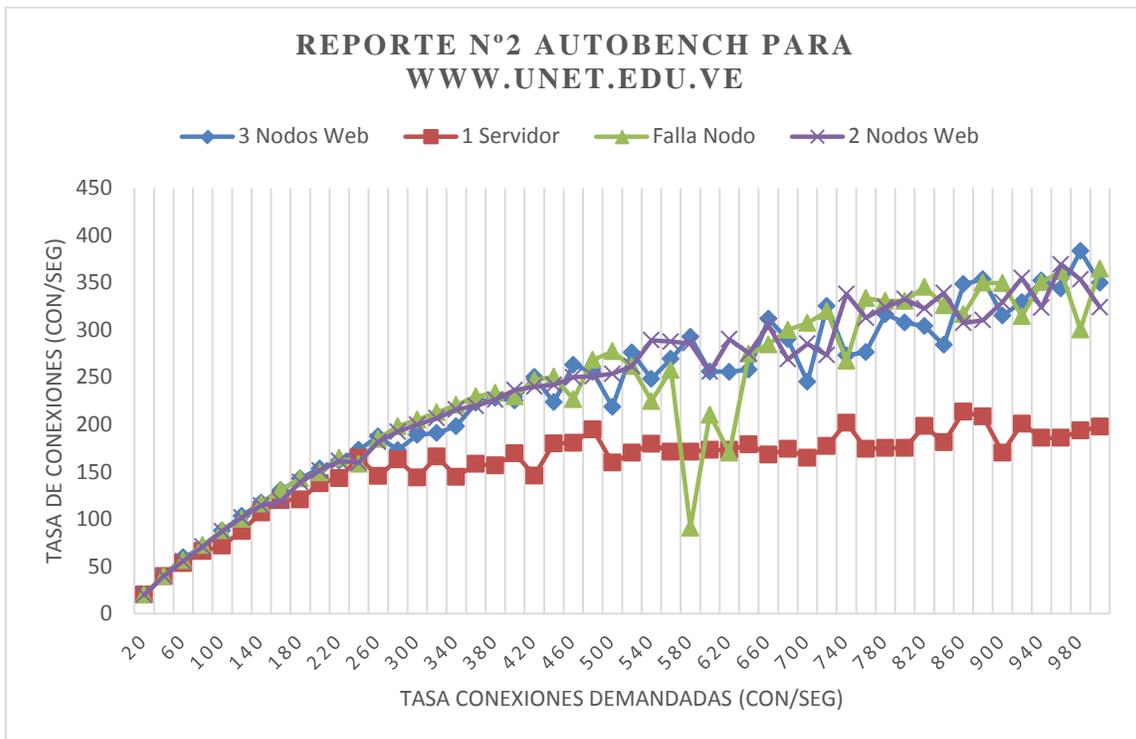
En los resultados mostrados en la Figura 4 se puede observar cuál tasa de conexiones servidas van en aumento respecto a la tasa de conexiones bajo demandada. En la prueba realizada los resultados arrojaron mejores valores con dos servidores web.

Se notó que los resultados para las arquitecturas compuestas de dos y tres nodos, variaron entre un rango de valores parecidos de conexiones/servidas con respecto a las demandadas. Sin embargo, al contrastar la Figura 4 y la Figura 3 se observa que un clúster de tres nodos genera mayor cantidad de solicitudes por conexión que uno compuesto de solo dos nodos.

La serie de 3 nodos se satura a partir de 260 conexiones/segundo bajo demanda, escalando entre valores que van de 173 a 383.5 conexiones/segundo. La serie de 2 nodos se satura a partir de 220 conexiones demandadas/segundo con una tasa de 161.4 con/seg., ascendió a valores de hasta 369.5 y descendió a 159.8 con/segundo, luego del punto de saturación.

Para un clúster con un evento de falla, las conexiones se pierden momentáneamente y desciende a una tasa de 90.8 al existir 560 conexiones/segundo. Para un servidor independiente, el punto de saturación aproximado fue de 164.9 a existir 240 conexiones bajo demanda/segundo.

**Figura 4 . Reporte 2 Autobench (httpperf)**



**Fuente:** elaboración propia.

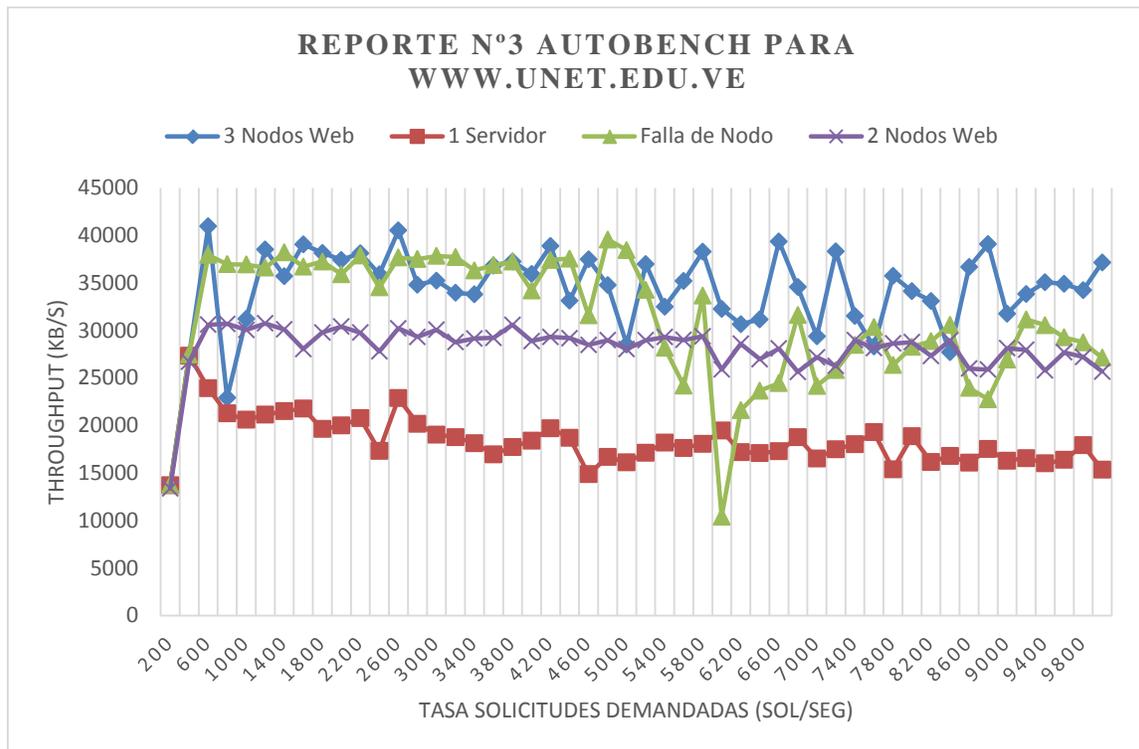
La Figura 5 muestra los resultados de las mediciones realizadas sobre cuatro diferentes arquitecturas configuradas en términos de Kilo Bytes/Segundo (Eje Y) por solicitudes demandadas (Eje X).

Se puede observar que la serie del servidor independiente mantiene una tasa de transferencia más alta para la tasa de 400 solicitudes/segundo con un flujo de 27346,5 KB/seg., y se observa como el flujo tiene una tendencia a disminuir al existir mayor demanda.

Se nota, de igual manera, la disminución del flujo de información para la serie 2 Nodos, que se mantiene entre valores de 25000 y 30000 KB/seg.

También, en la Figura 5 se percibió que con tres servidores web se mantiene un flujo de información de valores entre 40000 y 30000 KB/s. En el caso de existir una falla con tres nodos web, hay una disminución en el flujo de información, y luego se normaliza a un rango parecido al de dos nodos.

**Figura 5. Reporte 3 Autobench (httpperf)**



Fuente: elaboración propia.

### CONJUNTO DE PRUEBAS N° 2: PRUEBAS SOBRE EL CLÚSTER DE SERVIDORES DE BASE DE DATOS

Seguidamente se muestra los resultados de las mediciones realizadas sobre cuatro diferentes arquitecturas de base de datos usado MySQL Clúster. Cada una de las series de datos utilizadas por las gráficas es descrita a continuación:

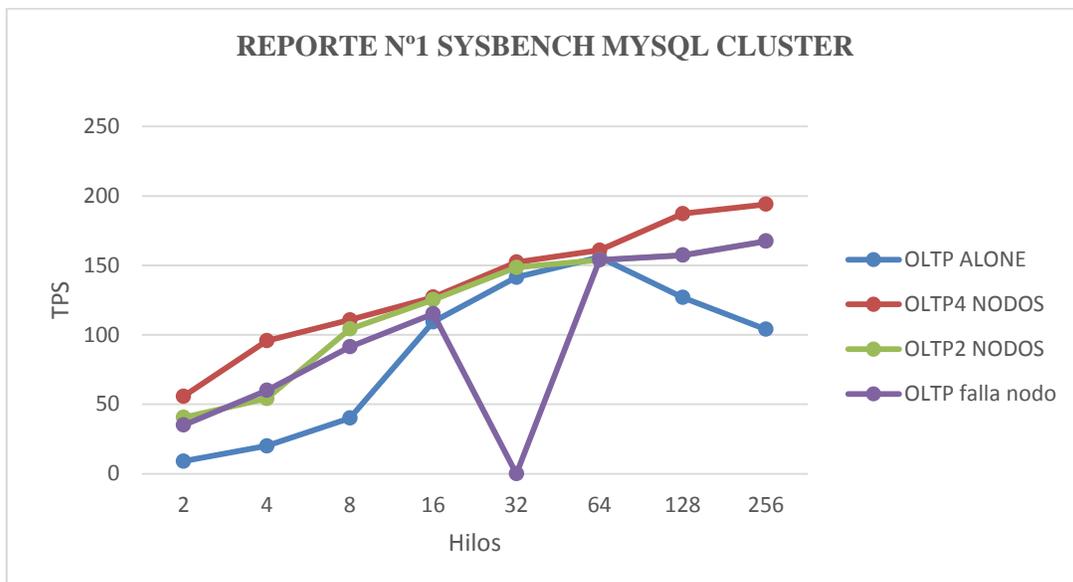
- OLTP2 NODOS: esta serie hace referencia al comportamiento del cl ster con dos nodos de datos.
- OLTP4 NODOS: esta serie hace referencia al comportamiento del cl ster con cuatro nodos de datos.
- OLTP falla nodo: esta serie hace referencia al comportamiento del cl ster con 4 nodos al existir la ca da de un nodo de dato.
- OLTP\_ALONE: esta serie hace referencia al comportamiento de un servidor de base de datos InnoDB sin el uso de un proxy.

En la Figura 6 se puede observar una gr fica del n mero de Transacciones por Segundo seg n el n mero de hilos.

Se puede observar que hay una p rdida moment nea de las transacciones en la serie OLTP falla nodo, pero en esta disminuci n el cl ster eventualmente se sobrepone a la situaci n de falla, y contin a en normal funcionamiento.

Tambi n se denota c mo el servidor independiente se satura al existir 64 hilos (clientes), con un valor m ximo de aproximadamente 154 Transacciones por Segundo.

**Figura 6. Reporte 1 Sysbench**

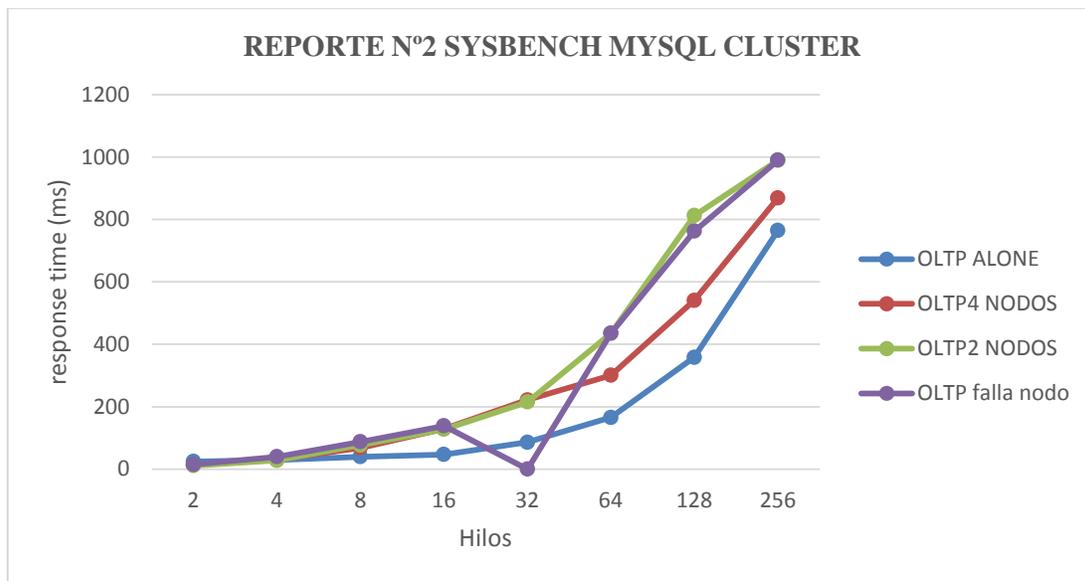


**Fuente:** elaboraci n propia.

En cuanto a los tiempos de respuesta en la Ilustraci n 7 se puede observar una tendencia, entre mayor es el n mero de nodos de datos de MySQL Cl ster disminuyen los tiempos de respuestas, debido a que la carga es distribuida y servida m s r pidamente.

Sin embargo, fue interesante observar que la serie OLTP\_ALONE, que se compone de una máquina MySQL InnoDB sin el uso de proxy, dio valores más bajos con respecto a los tiempos de repuesta.

**Figura 7. Reporte 2 Sysbench**

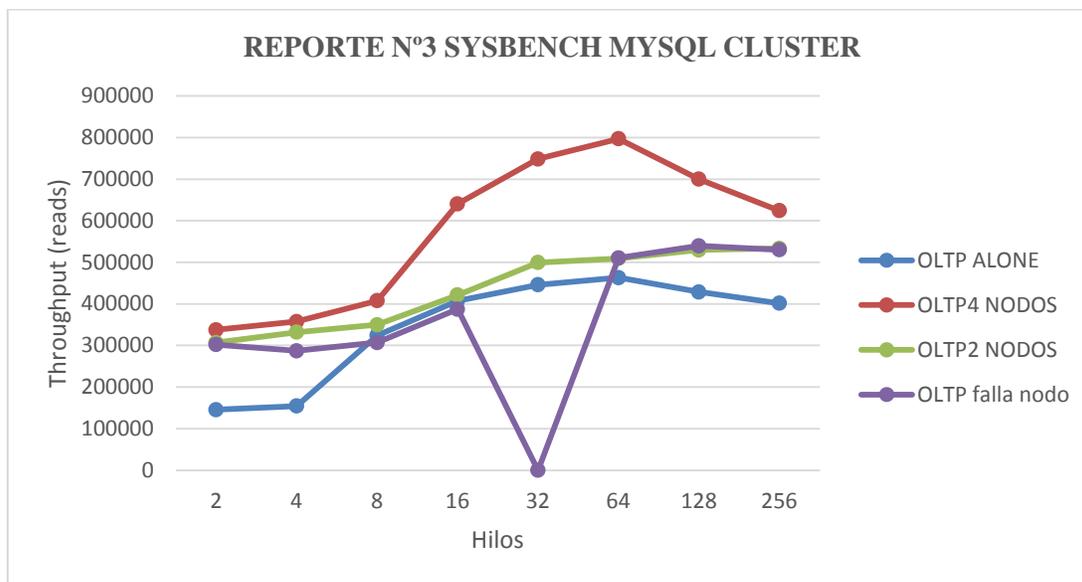


**Fuente:** elaboración propia.

En la Figura 8 se puede observar que una base de datos de cuatro nodos realiza casi el doble de lecturas que un clúster de dos nodos. Esta se satura con 64 hilos realizando aproximando de 797000 lecturas. El servidor MySQL InnoDB posee un punto de saturación de 510455 lecturas al existir 64 hilos.

Por otro lado, se nota la caída en transacciones por parte de la serie "OLTP falla nodo", pero la base de datos sigue disponible llevando a cabo 539000 lecturas con 128 hilos.

Figura 8. Reporte 3 Sysbench



Fuente: elaboraci n propia.

## CONCLUSIONES

Se logr  dar soluci n a la problem tica planteada, consiguiendo dise ar y probar un cl ster para aplicaciones web en la Universidad Nacional Experimental del T chira. Infraestructura tecnol gica que permitir  a aplicaciones alcanzar altos niveles de prestaci n de servicio, robustez, escalabilidad y disponibilidad.

Los resultados obtenidos durante la fase de diagn stico demostraron que la gesti n y despliegue de las aplicaciones web son eficaces, sin embargo, carecían de una infraestructura con caracter sticas de alta disponibilidad y tolerante a fallas, para brindar servicios web y de base de datos de alta fiabilidad.

Se logr  identificar componentes de software libre orientados a servicios web y de base de datos flexibles, adaptables y escalables para el dise o y desarrollo del cl ster. Para ello, se seleccion  la base de MySQL Cl ster y el servidor web Apache, con la ayuda de la herramienta balanceadora de carga HA Proxy.

Fue posible la integraci n de las herramientas de software seleccionadas para conseguir un funcionamiento tolerante fallas y redundante del cl ster, tal como lo demostraron las pruebas realizadas una vez implementado e instalada la aplicaci n de prueba seleccionada con base a una utilizada en CETI. Para este caso, se hizo la prueba con Moodle.

Se establecieron m tricas de carga y rendimiento sobre un cl ster orientado a brindar servicio web y el servicio de base de datos utilizando herramientas libres para la



realización de pruebas. Se identificaron las métricas más representativas en base al rendimiento del sistema, con respecto a los servicios web y de base de datos, utilizando las herramientas Pandora FMS, Httpperf y Sysbench respectivamente para su medición, monitorización y para realizar las concernientes comparaciones.

Las características como escalabilidad, alta disponibilidad y rendimiento de aplicaciones web sobre un clúster, fueron verificadas mediante la implementación de diferentes casos en un ambiente de pruebas. Dichas características, se comprobaron mediante el uso de cuatro casos para un clúster web, y un clúster de base de datos, obteniendo resultados satisfactorios.

Es importante señalar que el trabajo de investigación realizado no solo se vincula a los diferentes escenarios elaborados en cuanto al número de máquinas utilizadas, sino que es extensible y escalable a un número mucho mayor de nodos.

En conclusión, el diseño del clúster se ejecutó de una manera sistemática y organizada cumpliendo con los objetivos establecidos. Se notó cómo los servicios web y de base de datos que trabajan de manera distribuida permiten balancear la carga entre los servidores, aumentando las capacidades de los mismos, mientras se provee sistemas de alta disponibilidad y tolerante a fallas.

#### REFERENCIAS BIBLIOGRÁFICAS

- Basu, S., Jiani, G., Kumar, R., & Laxmi, B. (2005). QoS Aware Job Scheduling in a Cluster-based Web Server for Multimedia. 19th IEEE International Parallel and Distributed Processing Symposium. Abril, Estados Unidos.
- Blanco, M. (2003). Sistemas de Alta Disponibilidad. Telematique. Volumen 2, Número 1. (pp. 32-41).
- DB-ENGINES (2014). DB-Engines Ranking. Documento en línea. Disponible en: <http://db-engines.com/en/ranking>. Consulta: 15/03/2014.
- Maldonado, J. (2013). Entrevista Personal. Universidad Nacional Experimental del Táchira, Venezuela.
- Moodle (2014). About Moodle. Documento en línea. Disponible en: [http://docs.moodle.org/26/en/About Moodle](http://docs.moodle.org/26/en/About_Moodle). Consulta: 15/03/2014.
- MySQL (2013). MySQL Cluster FAQ. Documento en línea. Disponible en: <http://www.mysql.com/products/cluster/faq.html>. Consulta: 15/03/2014.
- Nations, D. (2012). What is a Web Application? Documento en línea. Disponible en: [http://webtrends.about.com/od/webapplications/a/web\\_application.htm](http://webtrends.about.com/od/webapplications/a/web_application.htm). Consulta: 20/01/2013.



Netcraft (2014). January 2014 Web Server Survey. Documento en línea. Disponible en: <http://news.netcraft.com/archives/2014/01/03/january-2014-web-server-survey.html>. Consulta: 15/03/2014.

Oracle (2014). MySQL Benchmark Tool. Documento en línea. Disponible en: <https://dev.mysql.com/downloads/benchmarks.html>. Consulta: 02/03/2014.

Agile Support Team (2013). Agileload Blog. Documento en línea. Disponible en: <http://www.agileload.com/agileload/blog/2013/02/18/web-applications-performance-testing-metrics>. Consulta: 05/03/2014.

Zentyal (2012). Zentyal Documentation. Documento en línea. Disponible en: <http://doc.zentyal.org/en/presentation.html>. 02/02/2014.