

Servidor Proxy Caché: Comprensión y asimilación tecnológica¹

Artículo de Investigación - Fecha de recepción: 1 de agosto de 2012 - Fecha de aceptación: 6 de septiembre de 2012

Carlos E. Gómez

Magíster en Ingeniería en Sistemas y Computación, Licenciado en Matemáticas y Computación, Universidad del Quindío. Armenia, Colombia, carloseg@grid.edu.co

Luis E. Sepúlveda

Ingeniero de Sistemas y Computación, Magíster en Software Libre, Universidad del Quindío. Armenia, Colombia, lesepulveda@grid.edu.co

Christian A. Candela

Ingeniero de Sistemas y Computación, Magíster en Comercio Electrónico, Universidad del Quindío. Armenia, Colombia, christiancandela@grid.edu.co

RESUMEN

Los proveedores de acceso a Internet usualmente incluyen el concepto de aceleradores de Internet para reducir el tiempo promedio que tarda un navegador en obtener los archivos solicitados. Para los administradores del sistema es difícil elegir la configuración del servidor proxy caché, ya que es necesario decidir los valores que se deben usar en diferentes variables. En este artículo se presenta la forma como se abordó el proceso de comprensión y asimilación tecnológica del servicio de proxy caché, un servicio de alto impacto organizacional. Además, este artículo es producto del proyecto de investigación “Análisis de configuraciones de servidores proxy caché”, en el cual se estudiaron aspectos relevantes del rendimiento de *Squid* como servidor proxy caché.

Palabras clave

Proxy server, proxy caché server, proxy caché, caché web, Squid.

1. Artículo derivado del proyecto de investigación: *Análisis de configuraciones de servidores proxy caché*, del Grupo de Investigación en Redes, Información y Distribución - GRID, Universidad del Quindío, 2011.

*Web Proxy Caching:
Understanding and technological assimilation*

ABSTRACT

Internet Service Providers (ISPs) usually include the concept of internet accelerators to reduce browser's response time for obtaining requested files. For system's administrators is difficult to choose the proxy cache server configuration, considering that it is necessary to decide the values to be used in different variables. This article sets out the way in which the process of understanding and technology assimilation of the proxy cache server was address, which in turn is a high-impact organizational service. Moreover, this article is a product from the research project titled "Configuration Analysis of Proxy Cache Servers", where relevant aspects of Squid as proxy cache server were studied.

Keywords

Proxy server, proxy cache server, proxy caching, web caching, *Squid*.

INTRODUCCIÓN

Una de las principales actividades que desarrolla un usuario de Internet es navegar entre las millones de páginas y sitios web. Como es bien conocido, la web es ampliamente usada en todo el mundo y el Protocolo de Transferencia de Hipertexto (HyperText Transfer Protocol - HTTP) es considerado como el motor que permite el intercambio de información y define la forma como los navegadores (programas cliente web) solicitan objetos a los servidores web y cómo los servidores entregan la información solicitada [12].

HTTP es un protocolo de la capa de aplicación del modelo TCP/IP y es el corazón de la web. Las especificaciones técnicas del protocolo HTTP están definidas en los documentos RFC 1945 [3] y 2616 [7]. HTTP es implementado por dos tipos de programas: un programa servidor tal como “Apache Web Server” [1] y un programa cliente, por ejemplo el navegador “Mozilla Firefox” [13].

El programa cliente y el programa servidor usualmente son ejecutados en diferentes hosts y se comunican entre sí mediante el intercambio de mensajes HTTP. El protocolo HTTP define la forma como los navegadores solicitan objetos (archivos) a un servidor y cómo los servidores los transfieren a los clientes. Cuando un usuario solicita una página web (o hace clic en un enlace), el navegador genera y envía uno o varios mensajes de solicitud HTTP al servidor. El servidor recibe cada solicitud y responde con un mensaje de respuesta HTTP el cual

típicamente contiene el objeto solicitado [12]. La interacción entre el cliente y el servidor web es ilustrada en la Fig. 1.

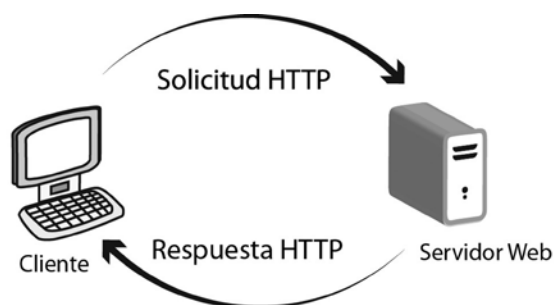


Fig. 1 Interacción entre cliente y servidor web

Cuando se solicita un archivo desde un navegador a un servidor web, el archivo tarda un tiempo en ser recibido por el navegador. El tiempo total está relacionado con variables como la cantidad de bytes del archivo solicitado, la distancia entre el servidor web y el navegador; la capacidad de comunicación de los distintos enlaces físicos por donde debe pasar el mensaje, la eficiencia de los dispositivos activos intermedios y el tráfico [12].

Compartir el acceso a Internet para una organización, más que una opción es una necesidad, bien sea por la falta de direcciones IP públicas para acceder a la red, o por razones de seguridad. Las dos soluciones más aceptadas para ofrecer el servicio de acceso compartido a Internet son el uso de un servicio de traducción de direcciones de red (NAT), o la implementación de un servidor proxy.

El servicio NAT consiste en la interceptación de datagramas IP salientes y entrantes, de manera que sea posible convertir las direcciones IP pertenecientes a una red

interna para hacerlas compatibles con una red externa. Típicamente, el servicio NAT es prestado por un router o un host configurado con más de una tarjeta de red y se utiliza generalmente para pasar de una red interna, en la cual los hosts tienen configuradas direcciones IP privadas, para permitir el acceso a Internet. En este proceso, el dispositivo intermedio modifica tanto la dirección IP origen (la dirección local), como el número de puerto origen y esta información es registrada en una tabla. Cuando llegan los mensajes de respuesta desde el exterior, los mensajes también son interceptados, y se modifican nuevamente la dirección IP y el número de puerto destino, de acuerdo con la información registrada. De este modo es posible entregar correctamente cada datagrama IP recibido. Esta solución es realizada en la capa de red, aunque de alguna manera utiliza información de la capa de transporte según el modelo de referencia TCP/IP [12].

Por su parte, un servidor proxy es una so-

lución software implementada en la capa de aplicación la cual intercepta los mensajes de solicitud HTTP (y otros protocolos), para hacer la solicitud en representación de los usuarios de la red corporativa. Generalmente un servidor proxy se ubica en la frontera entre la red corporativa y la red del proveedor de acceso a Internet [8].

En las dos soluciones mencionadas, todas las solicitudes pasan a través de un dispositivo intermedio, lo que ocasiona un cuello de botella y en consecuencia se hace más lento el servicio de acceso a Internet a los usuarios. Sin embargo, es posible aprovechar la ubicación de un servidor proxy para guardar allí copias de archivos que hayan sido descargados por algún navegador de la red corporativa, los cuales probablemente sean solicitados en el futuro por algún usuario de la red. Las copias de los objetos estarán más cerca de los usuarios, por lo tanto, serán entregadas mucho más rápido, mejorando significativamente el rendimiento de todo el sistema. Esta técnica es am-

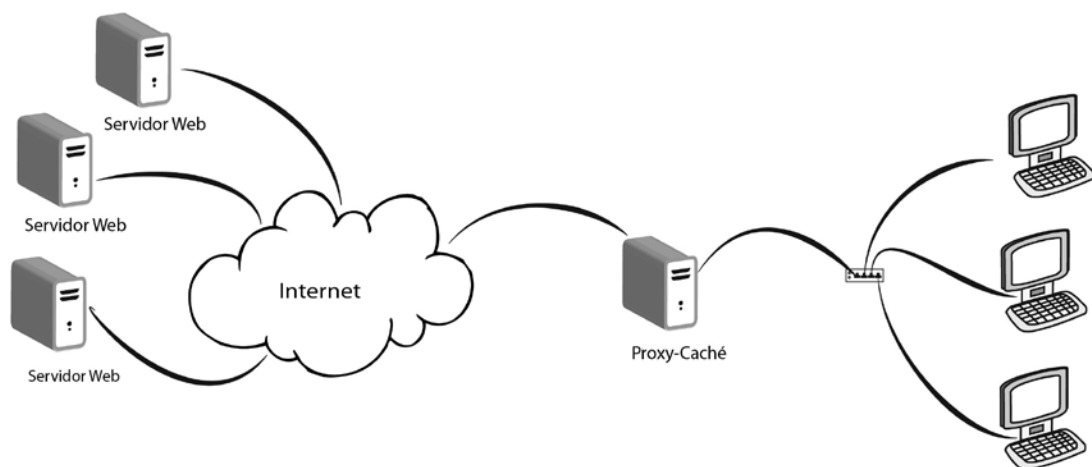


Fig. 2 Servidor proxy caché en una red corporativa

pliamente utilizada en diferentes sistemas y es conocida con el nombre de caché, dando lugar al término servidor *proxy caché* [6]. La interacción entre el cliente y el servidor web con un servidor proxy en el medio es ilustrada en la Fig. 2.

La configuración de *Squid* es realizada mediante la asignación de valores a un número cercano a 200 directivas en un archivo de configuración, similar al de otros servicios en sistemas tipo Unix. La mayoría de estas directivas tiene un valor asignado por defecto; sin embargo, es necesario ajustar algunos de estos valores para tener un servidor *Squid* operando con una configuración básica [15].

Las directivas de *Squid* se pueden agrupar en siete categorías, de acuerdo con sus características. Un primer grupo de directivas de identificación; un segundo grupo establece el control de acceso básico; el tercer grupo establece el control de acceso avanzado; en cuarto lugar, las directivas de configuración básica de caché; en quinto lugar, la configuración avanzada de caché; un sexto grupo dedicado al caché distribuido; y un grupo de directivas dedicado al control de ancho de banda. Las directivas que tienen relación con el proyecto de investigación son las directivas de configuración básica del caché, entre las que se encuentran la especificación del directorio de caché, el espacio en disco, los límites al tamaño de los archivos que se van a almacenar en caché, la estructura de directorios de caché, y las políticas de reemplazo; y las directivas de configuración avanzada de caché, específicamente los sis-

temas de archivos y los esquemas de almacenamiento [15].

La selección de los parámetros con que se va a configurar y los valores que se van a asignar, generalmente está basada en la experiencia, en heurísticas y en los valores que se asignan por defecto [10].

El rendimiento de un servidor proxy caché depende de su ubicación en la red corporativa, de las solicitudes realizadas por los usuarios y de los intereses comunes de ellos. Para medir el rendimiento de un servidor proxy caché, se contabilizan los aciertos y fallos en la búsqueda de los objetos que sean solicitados por los usuarios [6].

Usualmente, es necesario esperar un tiempo determinado para obtener estadísticas acerca de las solicitudes de los usuarios y de las respuestas del servidor proxy. Con base en esas estadísticas se pueden tomar decisiones para su reconfiguración.

El proyecto de investigación “Análisis de configuraciones de servidores proxy caché”, del cual este artículo es producto, fue desarrollado con el fin de identificar parámetros de configuración de un servidor proxy caché relacionados con su desempeño, en busca de soluciones que mejoren las condiciones en las que los usuarios de una organización utilicen un servicio de acceso compartido a Internet. Lo anterior permite acercar este conocimiento tecnológico a la comunidad académica y, en consecuencia, a la sociedad y la industria mediante su interacción con profesionales en el área de infraestructura de *tecnología informática*.

La metodología empleada en el desarrollo del proyecto corresponde a una adaptación de la Metodología de la Investigación Tecnológica, propuesta por Sonia Ríos en 1998 [14], para su aplicación en proyectos de Infraestructura de Tecnología Informática. Producto de la adaptación de esta metodología, se identificaron cinco etapas: Estudios previos, Análisis del problema, Diseño de la solución, Implementación y Pruebas. Para adelantar la investigación, fue necesario entender el funcionamiento del servicio prestado por los servidores proxy caché y asimilar la tecnología. El servidor proxy caché seleccionado para el desarrollo del trabajo fue *Squid*, entre otras, por las siguientes cuatro razones: está basado en software libre, lo que implica acceso al código fuente y la posibilidad de modificarlo de acuerdo con las necesidades; es ampliamente utilizado en el mundo; está bastante documentado; y proporciona diferentes posibilidades para analizar su comportamiento. Además, se analizaron antecedentes de investigaciones realizadas en trabajos relacionados, a partir de los cuales se identificaron algunas estrategias de investigación, métricas de desempeño utilizadas en estudios similares, y se analizaron los resultados de un estudio publicado sobre el rendimiento de *Squid*.

El resto del artículo está estructurado de la siguiente manera: A continuación se presenta el trabajo relacionado; luego se hace una descripción de la metodología empleada en el desarrollo del proyecto; después se muestra el esquema de trabajo y la automatización del proceso de pruebas se encuentra posteriormente. Al final, las conclusiones.

TRABAJO RELACIONADO

Desde que se originó la web, a comienzos de la década del noventa, el número de usuarios ha crecido continuamente y el desempeño ha sido una preocupación constante para los investigadores. El uso de caché en la web ha sido una estrategia ampliamente estudiada para esta problemática, por lo que se realizó un análisis de la forma como se han desarrollado otros trabajos de investigación en temas relacionados con el análisis de desempeño de servidores proxy caché.

Se destacan tres elementos encontrados. En primer lugar, la estrategia utilizada por los investigadores para realizar sus estudios; en segundo lugar, las métricas utilizadas para medir el desempeño; y en tercer lugar, pruebas de desempeño que han sido publicadas por [15] como punto de referencia del desempeño de *Squid*.

Estrategias de investigación

Según [5] y [16], los trabajos de investigación sobre el desempeño de servidores proxy caché utilizan diferentes técnicas para la recolección de datos. Algunas siguen métodos formales, mientras que otras son evaluaciones empíricas; sin embargo, todas son ampliamente aceptadas y utilizadas. Entre las técnicas más populares se encuentran la simulación de redes con datos artificiales; la implementación de sistemas reales en redes aisladas; y el análisis de archivos *log* reales. A continuación una breve descripción de estas técnicas.

Simulación de redes con datos artificiales

La simulación es un mecanismo de evaluación que no obliga implementaciones completas aunque requieren el conocimiento detallado del sistema que se desea evaluar.

Sistemas reales en redes aisladas

Esta técnica consiste en crear escenarios controlados en los cuales se eliminan variables que a pesar de ser reales pueden distorsionar los resultados obtenidos.

Análisis de archivos *log* reales

Es una práctica usada con frecuencia debido a que los datos son obtenidos a partir del comportamiento real de los usuarios.

Métricas de desempeño

En estudios acerca del desempeño de los servidores proxy caché [16] señala que hay trabajos orientados a mejorar las métricas como el *hit rate* y el *byte hit rate*; otros trabajos pretenden disminuir la latencia experimentada por el usuario; y algunos más intentan disminuir el tráfico hacia el exterior de la red. Sin embargo, hay algunas métricas no tradicionales que pueden tener incidencia en el desempeño del servidor proxy caché como la cancelación de la conexión entre el cliente y el servidor, y la influencia que tiene el uso de cookies [4]; la proporción de reducción de costo [2]; las tasas máxima y promedio de ancho de banda consumido, el porcentaje de tráfico de red ahorrado debido al uso local de recursos almacenados en caché y la cantidad de tráfico

que puede ser atribuido al uso y administración del caché [16].

Las dos métricas de desempeño más aceptadas para evaluar el desempeño de un caché son el *hit rate* y el *byte hit rate* [6]. Sin embargo, la reutilización de objetos que estén almacenados en el caché depende en gran medida de las preferencias de los usuarios. Además, alcanzar un nivel mayor en el *hit rate* o en el *byte hit rate*, no necesariamente conduce a mejorar significativamente el desempeño del caché [16].

Desempeño de *Squid* con respecto al sistema de archivos y al esquema de almacenamiento

Squid ofrece una variedad de opciones en el proceso de instalación y configuración, especialmente relacionadas con el almacenamiento de los archivos en disco. Los sistemas de archivos son característicos de un sistema operativo particular. Casi todas las versiones de Unix tienen una implementación del *UFS (Unix File System)*, el cual también es conocido como el *FFS (Berkeley Fast File System)*. El sistema de archivos por defecto de *GNU/Linux* es *ext2fs*, aunque recientemente nuevos sistemas de archivos han sido desarrollados como *ext3fs* y *ext4fs*. Muchos sistemas operativos también soportan tecnologías de sistemas de archivos más nuevas. Estas incluyen nombres y acrónimos como *advfs*, *xfs* y *reiserfs*. *Squid* puede funcionar con cinco esquemas de almacenamiento: *ufs*, *aufs*, *diskd*, *coss* y *null*. Estos esquemas tienen diferentes propiedades y técnicas para organizar y acce-

der a los datos almacenados en el caché mediante llamadas a operaciones del sistema de archivos [15].

En [15] se publicaron los resultados de unas pruebas de desempeño de *Squid* en las cuales se hicieron análisis sobre sistemas de archivos y esquemas de almacenamiento en sistemas operativos distintos. Las pruebas realizadas al utilizar el sistema operativo *GNU/Linux* se relacionan mejor con el proyecto de investigación.

METODOLOGÍA

El proyecto de investigación se enmarcó dentro de la investigación tecnológica en ciencias de la ingeniería. La metodología empleada correspondió a una adaptación de la metodología propuesta por [14] para su aplicación en proyectos de infraestructura de tecnología informática. Producto de la adaptación de esta metodología, se identificaron cinco etapas: Estudios previos, Análisis del problema, Diseño de la solución, Implementación y Pruebas.

Los estudios previos corresponden a la primera etapa del proyecto. En esta etapa se identificó el problema objeto de estudio a partir de la oportunidad que representa para el Grupo GRID impactar positivamente el desarrollo de sus actividades misionales como son la investigación, la docencia y la proyección social. Posteriormente, se realizó la búsqueda y organización de la información, continuando con la revisión bibliográfica y el análisis de trabajos relacionados, obteniendo el estado general del

tema. Esta etapa finalizó con la formulación del proyecto.

El análisis del problema consiste en entender la situación objeto de estudio. Se inició con la identificación de los servicios fundamentales que ofrece *Squid* como servidor proxy caché. Luego, se examinó el formato del archivo de configuración de *Squid* para reconocer y analizar los parámetros relacionados con su desempeño.

El diseño de la solución se refiere a la búsqueda de ideas y al planteamiento de alternativas que conduzcan a resolver el problema objeto de estudio. El diseño inició con la adaptación de un modelo abstracto por capas con un enfoque descendente, definido en [11]. Este modelo es un referente abstracto del cual se pueden derivar diseños con especificaciones particulares, como las de este proyecto, ya que permite independizar funciones o responsabilidades y ocultar detalles internos de las distintas capas. Para este caso se diseñó un ambiente virtual controlado sobre el cual se puso en funcionamiento *Squid* en diferentes topologías de red, las cuales fueron de gran utilidad en la realización del estudio. Posteriormente, se diseñaron escenarios de prueba con diferentes configuraciones del servidor *Squid*, y se crearon archivos para ser almacenados en un servidor web (Apache Web Server). Para finalizar esta etapa se diseñó un sistema automatizado de pruebas a partir de sincronización de archivos.

La implementación corresponde a la etapa en la que se materializa el diseño realizado. Se utilizaron los recursos físicos disponi-

bles, se instaló una herramienta de virtualización empresarial, se crearon las máquinas y dispositivos virtuales con sus respectivos sistemas operativos y configuración de red. La parte final de la implementación consistió en la instalación y configuración del servidor web y del servidor proxy caché.

La etapa de pruebas permite validar el trabajo realizado. Para facilitar la comprensión del procedimiento utilizado en el proyecto de investigación, se establecieron dos fases: la estructura y el proceso de pruebas, las cuales se describen a continuación.

Estructura de las pruebas

Fue necesario definir ciertos elementos para establecer la estructura de las pruebas. Estos elementos son: la métrica de desempeño empleada; los archivos de prueba almacenados en el servidor web; y los parámetros de configuración del servidor *Squid* que fueron analizados.

Métrica de desempeño

Las métricas de desempeño que se tuvieron en cuenta son *hit rate*, *byte hit rate* y *tiempo promedio de descarga* de cada objeto web (archivo de prueba). Sin embargo, las dos primeras fueron descartadas debido a que tienen una alta relación con las preferencias de utilización del servicio por parte de los usuarios, las cuales no están directamente relacionadas con el objeto de estudio del proyecto de investigación. Por lo tanto, las pruebas realizadas se orientaron a medir el tiempo promedio de descarga de archivos de prueba previamente identificado.

Archivos de prueba

Como se ha mencionado, para realizar una prueba en este estudio, se debía medir el tiempo promedio de descarga de un objeto web. Por lo tanto, se necesitaban archivos disponibles para descargar, los cuales debían estar almacenados en el servidor web y que el cliente los tuviera debidamente identificados al momento de hacer las solicitudes. Es de anotar que el tamaño de cada archivo está directamente relacionado con el tiempo que tarda obtenerlo desde cuando se realizó la solicitud. En este estudio se utilizaron archivos de tamaños diferentes.

Parámetros de configuración del servidor *Squid*

El objetivo del proyecto de investigación era analizar distintas configuraciones en un servidor proxy caché y la herramienta seleccionada para realizar las pruebas fue *Squid Cache*. Dado que la configuración de *Squid* se realiza mediante un archivo plano de configuración en el cual se especifican los valores de un número importante de parámetros, se procedió a identificar los parámetros más relevantes con respecto al desempeño del servidor *Squid*. Los principales parámetros de configuración estaban relacionados con el sistema de archivos y el esquema de almacenamiento, y son coherentes con lo expresado por [15].

Proceso de pruebas

Dado que el objetivo era calcular el tiempo promedio al realizar descargas de objetos web, se diseñó el proceso de pruebas con los pasos que se describen a continuación:

1. Selección de un archivo de configuración específico con un conjunto de parámetros cuyos valores están determinados.
2. Inicio del servidor proxy caché *Squid* con el caché vacío.
3. Solicitud de un archivo determinado al servidor web. Esta solicitud se realiza un número de veces especificado, para un conjunto determinado de archivos.
4. Registro del tiempo de inicio y fin de cada archivo descargado, para medir el tiempo transcurrido en cada descarga.
5. Realización de los análisis respectivos.

ESQUEMA DE TRABAJO

Para la realización de las pruebas del proyecto de investigación se diseñó un esquema de trabajo dividido en capas, cada una de ellas orientada a un conjunto de actividades relacionadas. El esquema propuesto se puede observar en la Fig. 3.

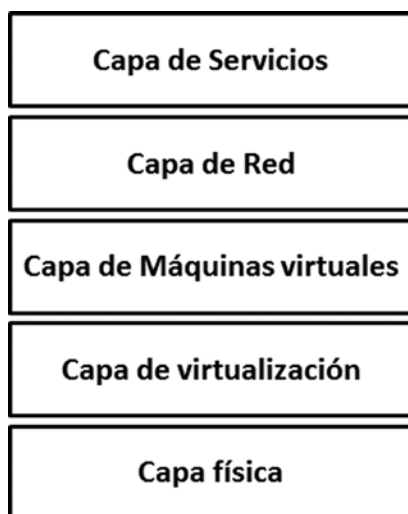


Fig. 3 Esquema de trabajo por capas

Capa física

La capa física del esquema de trabajo del proyecto de investigación consistió en un Servidor Dell Power Edge T100 equipado con un procesador Intel Xeon de cuatro núcleos de 2.4 GHz, 4 GB de memoria Ram y disco duro de 500 GB. Este equipo fue adquirido con el fin de implementar virtualización empresarial por hardware (paravirtualización) para el desarrollo de los proyectos del grupo de investigación en Redes, Información y Distribución - GRID, de la Universidad del Quindío.

Capa de virtualización

Para dar un adecuado aprovechamiento a los recursos de hardware disponibles, y además eliminar posibles errores presentes tanto en los medios físicos de transmisión de datos como en equipos activos, se implementó un ambiente virtual controlado para formar una red aislada. En la implementación del ambiente virtual se utilizó VMware ESXi, una versión gratuita de VMware, la cual permite dar los primeros pasos en virtualización empresarial de última generación. ESXi utiliza un hipervisor o monitor de máquinas virtuales, el cual es un componente de software que actúa directamente sobre el hardware para permitir la creación de máquinas virtuales sin necesidad de un sistema operativo host optimizando el hardware disponible. El hipervisor administra recursos hardware como memoria, procesador, disco y dispositivos periféricos, para ser asignados a las máquinas virtuales de acuerdo con las necesidades y según el caso [9].

Capa de máquinas virtuales

Se utilizaron tres máquinas virtuales (Fig. 4) en las cuales se configuró el sistema operativo GNU/Linux Debian Lenny (Versión 5). La selección de este sistema operativo se hizo de manera arbitraria, debido a que podría haberse utilizado cualquier otra distribución. Sin embargo, se eligió por ser una distribución altamente reconocida por su estabilidad y aplicación en la configuración de servidores, además de tener una gran comunidad de desarrolladores alrededor del mundo. Las tres máquinas tenían la misma configuración: Procesador 2 GHz, memoria Ram: 1 GB, disco duro: 20 GB. En cuanto a la configuración de la red, tanto el cliente como el servidor web contaban con una tarjeta de red, mientras que el servidor proxy caché tenía dos.

Capa de red

La capa de red del esquema de trabajo corresponde a la topología de red empleada para el desarrollo del proyecto. En este caso, las tres máquinas virtuales estaban configuradas en dos segmentos de red di-

ferentes. El primer segmento representaba la red local o corporativa y el segundo segmento representaba una red externa en la cual se encontraba el servidor web. Para la conexión de los equipos se utilizaron switches virtuales proporcionados por *VMware ESXi*. Es de resaltar que en esta topología no se presentaban mensajes en difusión que pudieran alterar el rendimiento de la red, ni se consideraron los retrasos cada vez que un paquete pasaba a través de un dispositivo activo local. Por otra parte, a pesar de simular el paso entre dos redes diferentes, no se contaba con enrutadores intermedios que pudieran aumentar el retardo extremo a extremo. Tampoco se tuvo en cuenta el retraso ocasionado por el acceso a la red externa ni los retrasos producidos por el proceso de resolución de nombres de dominio, lo cual estaba fuera del alcance del proyecto. La Fig. 5 muestra la topología de red.

Capa de servicios

En la capa de servicios se adicionaron roles específicos a cada una de las máquinas virtuales creadas para desempeñar las funciones necesarias para la realización de las



Fig. 4 Capa de máquinas virtuales

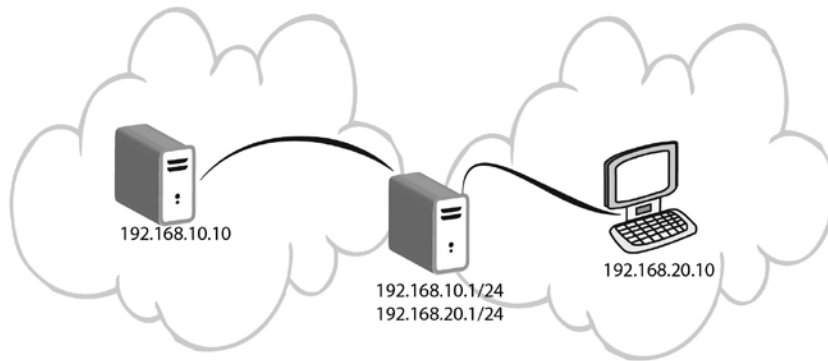


Fig. 5 Topología de red

pruebas. De esta manera, se implementó un servidor web, un servidor proxy caché y un cliente web, tal como se puede apreciar en la Fig. 6.

AUTOMATIZACIÓN DEL PROCESO DE PRUEBAS

Para la ejecución de una prueba era necesario invertir una cantidad de tiempo considerable, debido a que se trataba de un proceso dispendioso, que usualmente era realizado en forma manual. Esta situación limitaba la adquisición de información determinante para el estudio de las configuraciones del servidor proxy caché. Además, podía inducir al error humano en la ejecución de estas actividades.

Por lo anterior, y con el fin de aumentar la cantidad de información recolectada para la realización de los análisis de una manera ágil y fácilmente repetible y precisa, se diseñó un ambiente de pruebas automatizado para su ejecución sobre los servicios ilustrados en la Fig. 6. Este ambiente consistía en tres componentes: un generador de solicitudes que funcionaba en el cliente; un cargador automático de archivos de configuración que se ejecutaba en el mismo equipo donde corría el servidor proxy caché; y una zona compartida comunicada por la red local, la cual era utilizada para sincronización a partir de archivos compartidos. Para evitar la intervención de un usuario se diseñó un procesamiento de archivos en lote, a par-

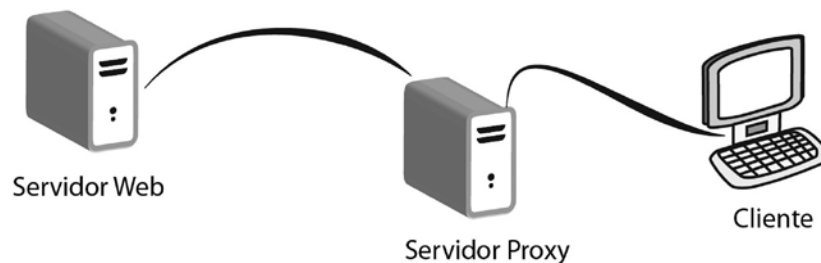


Fig. 6. Capa de servicios

tir de los cuales se definieron: los archivos disponibles para descargar, la cantidad de veces que debía ser descargado un archivo y los archivos de configuración del servidor *Squid* que serían cargados para la ejecución de cada prueba. Por otra parte, el registro de actividades en bitácoras (archivos log) permitía obtener la información necesaria para calcular los tiempos promedio necesarios para los análisis de cada archivo de configuración. El ambiente automatizado de pruebas se desarrolló a partir de la combinación de herramientas disponibles en el sistema operativo, shell scripts y aplicaciones software construidas a la medida en lenguaje C.

CONCLUSIONES, APORTE Y TRABAJO FUTURO

La configuración por defecto de *Squid* no es apropiada para todos los ambientes de producción. Es necesario estudiar las necesidades de la organización y, con base en ellas, realizar los ajustes respectivos, los cuales pueden ser tan sencillos como modificar un parámetro de configuración, o tan complejos como recompilar el software y/o el kernel del sistema operativo (si éste lo permite); utilizar varias estructuras de archivos para alojar los cachés y tal vez ajustar una variedad de parámetros en el archivo de configuración, sin dejar de lado la adecuada selección del hardware para la instalación del servicio.

El comportamiento de una configuración es altamente dependiente de los ambientes de ejecución particulares y lo que puede ser

conveniente en un escenario puede ser inconveniente en otro. Determinar los valores de los parámetros de configuración adecuados para un ambiente específico es una tarea no trivial; por lo tanto, las pruebas realizadas en el proyecto de investigación sirven de referencia para que los administradores de servidores *Squid* realicen ajustes a sus archivos de configuración.

El ambiente automatizado desarrollado dentro de este proyecto es un valor agregado para los administradores de infraestructura de redes, ya que puede ser aplicado a servicios similares en múltiples plataformas, siempre y cuando su comportamiento esté basado en parámetros especificados en un archivo de configuración.

La infraestructura para virtualización empresarial fue probada y funciona eficientemente, lo que permite aprovechar este recurso en futuros proyectos de investigación.

Como trabajo futuro, quedan muchas ideas para profundizar, entre las que se destacan la automatización de la asignación de los valores de los parámetros de configuración mediante la aplicación de inteligencia artificial; el uso de cachés cooperativos (distribuidos y jerárquicos); el uso de servidores proxy caché sobre plataformas de red basadas en IPv6; implementar un robot multi-hilo que simule ambientes con acceso concurrente al servidor proxy caché; y realizar estudios acerca de configuraciones propias del sistema operativo y de hardware que afecten el rendimiento del servidor proxy caché.

REFERENCIAS

- [1] Apache. *The Apache Software Foundation*. [Online] Disponible en: <http://www.apache.org/>, 2010.
- [2] R. Ayani, “Cost-based Proxy Caching”. *Proceedings of the International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, pp. 218-222. Wuxi, 2002.
- [3] T. Berners-Lee, *RFC 1945. Hypertext Transfer Protocol -- HTTP/1.0*, 1996.
- [4] R. Cáceres *et al.*, “Web Proxy Caching: The devil is in the details”. *Workshop on Internet Server Performance*. Madison, Wisconsin USA, 1998.
- [5] B. Davison, *A survey of proxy cache evaluation techniques*, 1999.
- [6] B. Davison, “A Web Caching Primer”. *IEEE Internet Computing*, Volume 5, Number 4, 38-45, 2001.
- [7] R. Fielding, *RFC 2616. Hypertext Transfer Protocol -- HTTP/1.1*, 1999.
- [8] H. Flynn, *Designing And Building Enterprise DMZs*. Syngress Media Inc., 2006.
- [9] B. Goldworm and A. Skamarock, *Blade Servers and Virtualization: Transforming Enterprise Computing While Cutting Costs*. Willey Publishing Inc., 2007.
- [10] C. E. Gómez and L. E. Sepúlveda, *Análisis de configuraciones de servidores proxy caché*, Informe Final. Universidad del Quindío, 2010.
- [11] C. E. Gómez and L. E. Sepúlveda, “Mejorando la calidad de servicio a través de configuraciones adecuadas de servidores proxy caché”. *Primer Seminario Taller Latinoamericano de Instrumentación, Control y Telecomunicaciones SICOTEL*. Universidad del Quindío, 2012.
- [12] J. Kurose, *Computer Networking: A top-down approach*, 5th Edition. Massachusetts: Addison Wesley, 2010.
- [13] Mozilla, Firefox. [Online] Disponible en: <http://www.mozilla.org/firefox/>, 2010.
- [14] S. Ríos, *Metodología de la Investigación Tecnológica*. Ediciones Hispanoamericanas Ltda., 1998.
- [15] D. Wesseles, *Squid: The definitive guide*. O’Reilly & Associates, Inc., 2004.
- [16] D. Zeng, “Efficient Web Content Delivery Using Proxy Caching Techniques”. *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews*, 2004.