

Tipo de artículo: Artículo original
Temática: Soluciones informáticas
Recibido: 12/11/19 | Aceptado: 22/01/2020 | Publicado: 02/02/2020

Solución informática para la selección del servidor web durante la migración a código abierto

Computer solution for the selection of the web server during the migration to open source

Nurisel Palma Pérez ^{1*}

¹ Centro de Software Libre. Universidad de las Ciencias Informáticas, Carretera a San Antonio Km 2 ½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370.

* Autor para correspondencia: npalma@uci.cu

Resumen

La presente investigación se centró en el objetivo de desarrollar una solución informática para aumentar la eficiencia en la selección de los servidores web Apache 2 y Nginx durante el proceso de migración a código abierto. Se aplicó el método Analítico-Sintético para el estudio de Apache 2 y Nginx y la determinación de las particularidades de cada uno. La realización de un estudio de caso aplicando la Norma Cubana ISO/IEC 25023:2017 y el método estadístico, permitió el procesamiento de la información recopilada acerca del comportamiento de Apache 2 y Nginx, esto sirvió de base para formular la teoría relacionada con la eficiencia de ambos servidores. A partir del tipo de contenido y del total de peticiones concurrentes se definieron 60 escenarios para la selección del servidor más eficiente, teniendo en cuenta todos los indicadores o cada medida específica: Rendimiento, Utilización de recursos y Capacidad. Las tres variantes a seleccionar son: Apache 2, Nginx o Nginx como proxy inverso de Apache 2. El resultado obtenido se materializa en la Herramienta para la Migración y Administración de Servicios Telemáticos, la cual posee un componente Web que permite la selección del servidor web que más se ajusta a la institución y la activación del módulo correspondiente para su administración. La solución se validó a través de un estudio de caso, del criterio de expertos en su variante Delphi y la técnica de Iadov. Finalmente la triangulación metodológica permitió confirmar el resultado satisfactorio de todos los métodos aplicados y el cumplimiento del objetivo planteado.

Palabras clave: Apache 2, eficiencia, Nginx, solución informática.

Abstract

The present research focused on the objective of developing a computer solution to increase efficiency in the selection of Apache 2 and Nginx web servers during the open source migration process. The Analytical-Synthetic method was applied for the study of Apache 2 and Nginx and the determination of the particularities of each one. The realization of a case study applying the Cuban Standard ISO/IEC 25023:2017 and the statistical method, allowed the processing of the information collected about the behavior of Apache 2 and Nginx, this served as the basis to formulate the

theory related to the efficiency of both servers. From the content type and the total of concurrent request, 60 scenarios were defined for the most efficient server selection, taking into account all the indicators or each specific measure: Performance, Resource utilization and Capacity. The three variants to select are: Apache 2, Nginx or Nginx as a reverse proxy of Apache 2. The result obtained is materialized in the Tool for Migration and Administration of Telematic Services, which has a Web component that allows the selection of the web server that more it adjusts to the institution and the activation of the corresponding module for its administration. The solution was validated through a case study, expert criteria in its Delphi variant and the Iadov technique. Finally, the methodological triangulation allowed to confirm the satisfactory result of all the applied methods and the fulfillment of the proposed objective.

Keywords: Apache 2, efficiency, Nginx, computer solution.

Introducción

Un servidor¹ web es una aplicación que responde a solicitudes provenientes de navegadores web, proporcionando recursos solicitados a través del protocolo HTTP (del inglés *Hypertext Transfer Protocol*) o de manera segura a través del protocolo HTTPS (del inglés *Hypertext Transfer Protocol Secure*) (MEJÍA, GONZÁLES y ESPAÑA, 2018). Actualmente son numerosos los servidores web, dentro de los que se encuentran Microsoft-IIS, Apache 2, Nginx, LiteSpeed, Google, Sun Java System y Lighttpd. Algunos son más usados que otros, la información sobre el uso de servidores web es proporcionada por algunos sitios como Netcraft (NETCRAFT, 2018) y Web Technology Surveys (W3TECHS, 2018), a partir de los reportes de ambos sitios en el mes de noviembre de 2018, se calculó la media del porcentaje de uso de cada uno de los cuatro servidores web que coinciden, determinándose Apache con 33,89%, Nginx con 29,61%, Microsoft-IIS con 24,79% y Google Servers con 1,23%, donde Apache y Nginx son los dos servidores web de código abierto más usados.

En Cuba también se evidencia el gran uso de estos dos servidores web de código abierto. Se realizó un análisis de los 40 principales sitios en Cuba de interés cultural y de entretenimiento, informativo, educativo e investigativo publicados por ETECSA y las tecnologías que estos emplean. Para analizar los sitios web se utilizó la herramienta multiplataforma Wappalizer, que permite identificar la tecnología utilizada en sitios web, incluidos los servidores (RAKHMAWATI, et al., 2018). Del análisis realizado se determinó que un 66,6% utiliza Apache 2, 27,3% Nginx y 6,1% Microsoft-IIS, evidenciándose que Apache 2 es notablemente el más usado.

En cuanto a la política integral para el perfeccionamiento de la informatización de la sociedad cubana, para lograr la soberanía tecnológica y desempeñar el proceso de migración, se destaca la Universidad de las Ciencias Informáticas (UCI) por desarrollar la distribución cubana de GNU/Linux Nova y llevar a cabo el proceso de migración a

¹ Software que realiza, controla o coordina un servicio o recurso (GILSTER, 2001)

tecnologías de software libre y plataformas de código abierto en diferentes instituciones cubanas (PÉREZ, 2015). La UCI ha realizado varios procesos de migración en diferentes entidades. La ejecución de los procesos de migración a código abierto a partir del año 2015 tiene como base la “Estrategia para la migración a aplicaciones de código abierto” elaborada por PÉREZ (2015).

La estrategia plantea que la migración de servicios telemáticos es el primer paso en el proceso de ejecución de la migración (PÉREZ, 2015). Dentro de estos servicios se encuentran los ofrecidos por los servidores web. Además como apoyo al proceso de migración se encuentra el libro “Buenas Prácticas para la Migración a Código Abierto”, el cual define que las alternativas libres a servidores web privativos para PYMES (Pequeñas y Medianas Empresas) y grandes empresas son Apache 2 y Nginx (PÉREZ, GARCÍA y GOÑI, 2015), sin embargo no se define cómo se debe realizar el proceso de selección de estos.

A partir de una entrevista realizada a los especialistas en servicios telemáticos pertenecientes a la UCI, se identificó que para la selección del servidor web a instalar, no existe una guía que oriente qué servidor se ajusta más a la institución. No se tiene un registro del estudio de servidores web migrados en procesos anteriores. Se evidencia la pérdida de tiempo y se duplican esfuerzos estudiando la alternativa a seleccionar. No se reutiliza el conocimiento para la selección del servidor web. Muchas veces se selecciona el servidor web más conocido aunque sea el menos eficiente para la institución. Dada la problemática existente, el objetivo general de la presente investigación está centrado en desarrollar una solución informática para aumentar la eficiencia en la selección de los servidores web Apache 2 y Nginx durante el proceso de migración a código abierto.

Materiales y métodos o Metodología computacional

Para el desarrollo de la investigación se utilizaron los métodos científicos y técnicas que a continuación se detallan.

- Analítico-Sintético: permitió el estudio de diferentes fuentes bibliográficas para extraer los elementos más importantes que se relacionan con los servidores web de código abierto Apache 2 y Nginx y a partir de los mismos determinar las particularidades de cada servidor.
- Hipotético deductivo: permitió verificar la hipótesis planteada en la investigación y establecer nuevas predicciones a partir de los conocimientos con los que ya se dispone.
- Modelación: permitió la creación de abstracciones con el objetivo de explicar la forma en la que cada servidor web procesa las peticiones que recibe. Se elaboraron representaciones explícitas de los tipos de arquitectura, la integración de Nginx como proxy inverso de Apache 2 y el funcionamiento del servidor web.

- Estadístico: permitió el procesamiento de la información recopilada acerca del comportamiento de Apache 2 y Nginx, facilitando las generalizaciones e interpretaciones de los datos. Se aplicó la estadística descriptiva utilizando una escala cuantitativa y la media como característica descriptiva, a los resultados de las pruebas de eficiencia aplicadas a los servidores web, con el objetivo de caracterizar los conjuntos de datos numéricos, poniendo de manifiesto de forma gráfica las propiedades de cada servidor.
- Entrevista: aplicada a los especialistas en servicios telemáticos del Centro de Software Libre para obtener información acerca de la experiencia en cuanto al proceso de selección de servidores web durante la migración a código abierto. Además se les aplicó a administradores de servidores web para conocer cómo se realiza actualmente el proceso de selección de estos en su institución.
- Encuesta: técnica de recopilación de información por medio de preguntas escritas organizadas en un formulario impreso, que se aplicó para determinar el nivel de satisfacción de los especialistas en servicios telemáticos con la herramienta informática como parte de la técnica de Iadov y en el marco del método Delphi para el criterio de expertos.

Se realizó un estudio sobre Apache 2 y Nginx, identificándose que Nginx se usa con frecuencia como un servidor proxy (SONI, 2016). El objetivo principal de configurar Nginx como un servidor de interfaz y darle a Apache una función de back-end, es mejorar la velocidad de publicación. Nginx tendría que diferenciar entre contenido estático y dinámico, en consecuencia, atender las solicitudes de archivos estáticos y reenviar las solicitudes de archivos dinámicos a un servidor back-end. Nginx actúa como un servidor proxy simple: recibe solicitudes HTTP del cliente (actuando como servidor HTTP) y las reenvía al servidor back-end (actuando como cliente HTTP) (NEDELUCU, 2015). Se determina que la configuración de Nginx como proxy inverso de Apache 2 es una buena solución para entornos donde se desee instalar Apache 2 y a la misma vez obtener las ventajas de Nginx para el manejo de peticiones. Además de analizar cada servidor web de forma independiente, la autora decide incluir la combinación de Nginx como proxy inverso de Apache 2 como alternativa para las instituciones cubanas.

A partir de una comparación realizada entre ambos servidores, se puede llegar a la conclusión que Apache 2 y Nginx tienen muchas características en común, sin embargo la diferencia de mayor impacto entre ambos servidores es la forma en que atienden las peticiones realizadas por el usuario². Esto proporciona quizás la diferencia más significativa en la forma en que responden a las diferentes condiciones del tráfico (CAMPOVERDE, HERNÁNDEZ y MAZÓN,

² Apache proporciona una arquitectura flexible basada en procesos y basada en hilos, mientras que Nginx presenta una potente arquitectura multiproceso e impulsada por eventos (NAM, 2017).

2015). Apache proporciona una arquitectura flexible basada en procesos y basada en hilos, mientras que Nginx presenta una potente arquitectura multiproceso e impulsada por eventos (NAM, 2017).

En Apache 2: cada solicitud es atendida por un hilo separado o proceso y utiliza sockets sincrónicos³ (NEDELUCU, 2015). Con la creación de la versión 2.0 se introducen los MPM (KABIR, 2003), que son responsables de conectar con los puertos de red de la PC servidora, aceptar las peticiones y generar los procesos hijos que se encargan de servirlos. Existen diferentes MPM, encontrándose entre ellos prefork, worker, event, threaded y perchild, cada uno basado en un funcionamiento diferente. Los sitios web que necesitan más que nada escalabilidad pueden usar un MPM hebrado como worker, mientras que los sitios web que requieran por encima de otras cosas estabilidad o compatibilidad con software antiguo pueden usar prefork (APACHE SOFTWARE FOUNDATION, 2017).

Arquitectura de Nginx: Nginx utiliza una arquitectura basada en eventos, una única instancia de Nginx consta de un proceso maestro y procesos de trabajo. Cada proceso de trabajo maneja conexiones múltiples, esto se logra ejecutando un bucle de eventos que extrae eventos que ocurrieron en sockets abiertos desde el sistema operativo. El sistema operativo es el encargado de distribuir las conexiones entrantes entre los procesos de trabajo de forma rotatoria (KHOLODKOV, 2015). Nginx hace uso de sockets asincrónicos⁴, no genera procesos tantas veces como recibe solicitudes. Un proceso por núcleo es suficiente para manejar miles de conexiones, lo que genera una carga de CPU y un consumo de memoria mucho más ligeros (NEDELUCU, 2015).

En cuanto al funcionamiento del servidor web, el otro elemento que distingue a ambos servidores es el soporte para protocolos basados en CGI⁵ (del inglés Common Gateway Interface), la forma en que procesan el contenido dinámico. A partir del estudio realizado, la autora elaboró el esquema que se aprecia en la Figura 1. Como se puede observar, el tiempo de respuesta y el consumo de recursos dependen de cómo el servidor maneje las peticiones que recibe y la forma en que procese el contenido a través de los protocolos basados en CGI. Por tanto la autora determina que el correcto funcionamiento del servidor web depende de la arquitectura, la cantidad de peticiones concurrentes y el tipo de contenido publicado.

³ Una comunicación es sincrónica cuando el envío y la recepción de información entre un remitente y un receptor son eventos simultáneos (CACCIAGRANO y CORRADINI, 2001).

⁴ Una comunicación es asíncrona cuando el envío y la recepción de información entre un remitente y un receptor no ocurren necesariamente en el mismo instante (CACCIAGRANO y CORRADINI, 2001).

⁵ Protocolo que describe la forma en que se intercambia la información entre el servidor web y la aplicación de puerta de enlace (PHP, Python, etc.) (NEDELUCU, 2015).

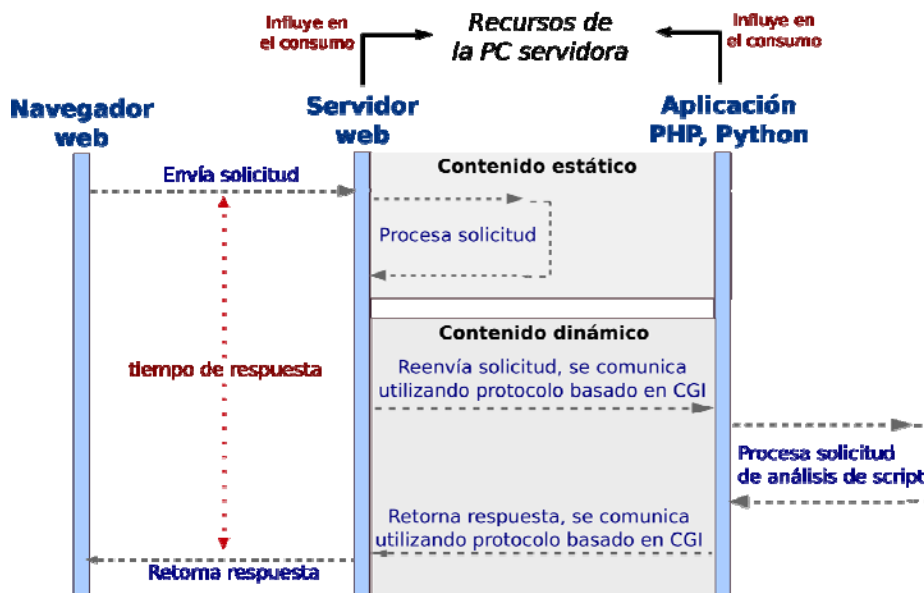


Figura 1: Esquema de funcionamiento del servidor web (Fuente: elaboración propia).

Resultados y discusión

La eficiencia en la selección de los servidores web Apache 2 y Nginx implica el desempeño adecuado del servidor web que se seleccione para un entorno determinado. Por tanto, se hace necesario el estudio y análisis de la eficiencia de ambos servidores en un entorno de prueba, por lo que se realizará un estudio de caso⁶.

Estudio de caso para determinar la eficiencia de Apache 2 y Nginx

El estudio tiene como objetivo definir la diferencia entre los servidores web Apache 2 y Nginx en cuanto a la eficiencia de desempeño. Para la medición de la eficiencia se emplean los indicadores de la Norma Cubana ISO/IEC 25023:2017 elaborada por la Oficina Nacional de Normalización⁷. Las medidas de eficiencia de desempeño se utilizan para evaluar el desempeño en relación con la cantidad de recursos utilizados en condiciones establecidas. Los recursos pueden incluir otros productos de software, la configuración de software y hardware del sistema y materiales.

⁶ Se refieren a estudios que al utilizar los procesos de investigación cuantitativa, cualitativa o mixta, analizan profundamente una unidad para responder al planteamiento del problema, probar hipótesis y desarrollar alguna teoría. En ocasiones utilizan la experimentación, se constituyen en estudios preexperimentales (HERNÁNDEZ, FERNÁNDEZ y BAPTISTA, 2006).

⁷ Organismo Nacional de Normalización de la República de Cuba y representa al país ante las organizaciones internacionales y regionales de normalización (OFICINA NACIONAL DE NORMALIZACIÓN, 2017).

Se ve fuertemente afectada y fluctúa dependiendo de las condiciones de uso, como la carga de datos de procesamiento, la frecuencia de uso, el número de sitios de conexión, entre otros. La eficiencia de desempeño se mide en cuanto a los siguientes parámetros: Rendimiento, Utilización de los recursos y Capacidad (OFICINA NACIONAL DE NORMALIZACIÓN, 2017).

La PC servidora se estudiará en tres escenarios de prueba diferentes en cuanto al tipo de contenido que puede ser: estático, dinámico con PHP o dinámico con Python. A su vez para el tipo de contenido estático existen dos escenarios en cuanto al servidor web instalado que puede ser Apache 2 o Nginx, por otra parte para tipo de contenido dinámico con ambos lenguajes de programación se encuentran tres escenarios: Apache 2, Nginx o Nginx funcionando como proxy inverso de Apache 2 (la autora de la investigación lo denominó Proxy). De lo anteriormente explicado se concluye que para cada uno de los 9 indicadores de la variable eficiencia de desempeño, se estudiarán ocho escenarios.

El estudio consistió en realizar con la herramienta *ab* (*Apache Benchmark*) un número determinado de peticiones con cierta concurrencia, desde una PC cliente al servidor web instalado en la PC servidora. En cada uno de los ocho escenarios se realizaron 10 observaciones, para todas con un total de 50 000 peticiones y cada una con concurrencias de 10, 100, 250, 500, 750, 1 000, 5 000, 10 000, 15 000 y 20 000 peticiones respectivamente. Además al mismo tiempo se realizó el monitoreo de los recursos de la PC servidora con la herramienta *dstat*. Para el cálculo de la eficiencia en cada indicador se definieron cinco escenarios en cuanto a la cantidad de peticiones concurrentes que pueden existir en las instituciones (denominada con la variable *p*), teniendo en cuenta las respuestas de los administradores de servidores web en una entrevista realizada. A continuación se describen cada uno de los escenarios.

- 1) **$p \leq 500$** : se refiere a una cantidad de peticiones concurrentes menor o igual que 500. Se incluyen cuatro observaciones con concurrencias 10, 100, 250 y 500 respectivamente.
- 2) **$500 < p \leq 1000$** : se refiere a una cantidad de peticiones concurrentes mayor que 500 y menor o igual que 1 000. Se incluyen dos observaciones con concurrencias 750 y 1 000 respectivamente.
- 3) **$1000 < p \leq 10000$** : se refiere a una cantidad de peticiones concurrentes mayor que 1 000 y menor o igual que 10 000. Se incluyen dos observaciones con concurrencias 5 000 y 10 000 respectivamente.
- 4) **$p > 10000$** : se refiere a una cantidad de peticiones concurrentes mayor que 10 000. Se incluyen dos observaciones con concurrencias 15 000 y 20 000 respectivamente.
- 5) **Todas**: se refiere a todas las concurrencias de peticiones incluyendo las 10 observaciones.

Para cada indicador se realizó el análisis correspondiente. En cuanto a la media de utilización del procesador (tiempo del procesador que se utiliza para ejecutar un determinado conjunto de tareas en comparación con el tiempo de operación), en la Figura 2 se aprecian cada una de las 10 observaciones. El tipo de contenido estático se expresa con columnas agrupadas, el dinámico con PHP mediante líneas y el dinámico con Python a través de líneas con marcadores. Para contenido estático los dos servidores web utilizan menos tiempo del procesador que para contenido dinámico. Nginx para todas las observaciones se mantiene como el servidor más eficiente, utilizando menor cantidad de tiempo que Apache. Para contenido dinámico con PHP y Python, Apache 2 utiliza considerablemente más tiempo que los demás servidores. Para Todas las observaciones en cada tipo de contenido el servidor web más eficiente es Nginx.

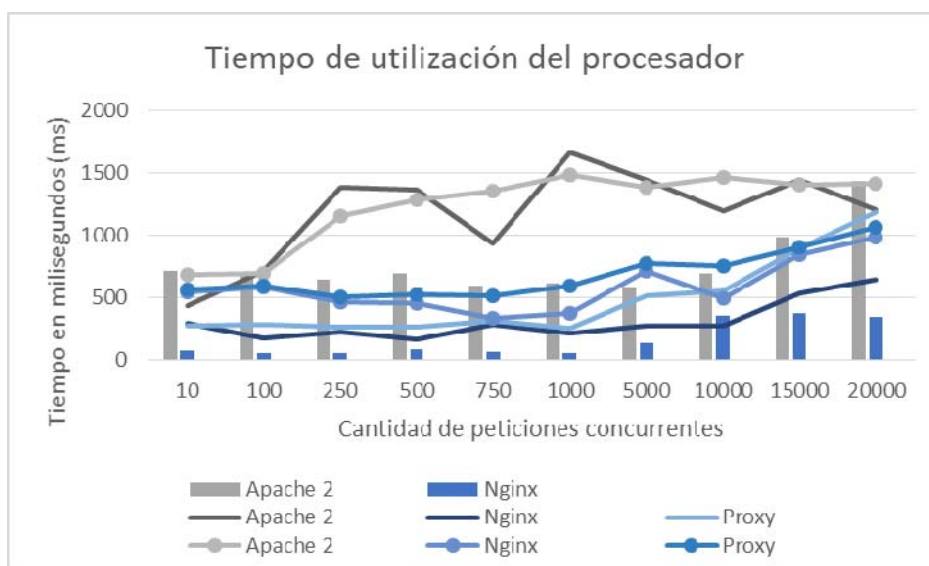


Figura 2: Tiempo de utilización del procesador (Fuente: elaboración propia).

A partir del análisis de la eficiencia calculada en cada indicador, con el objetivo de medir todos los indicadores en una misma escala, se asignó en cada escenario en cuanto a la concurrencia de peticiones y atendiendo al tipo de contenido, los valores 3, 2 y 1 a los servidores web en correspondencia con el orden de su eficiencia, de mayor a menor. Para casos de igual eficiencia se asignó el mismo valor. Para cada uno de los cinco escenarios en cuanto a la concurrencia de peticiones, se calculó la puntuación final para cada servidor web teniendo en cuenta todos los indicadores y además para cada conjunto de indicadores correspondientes a las tres medidas de la eficiencia que son Rendimiento, Utilización de los recursos y Capacidad. Finalmente se definieron 60 escenarios, como se aprecia en la Tabla 1.

Tabla 1: Escenarios definidos para la selección del servidor web (Fuente: elaboración propia).

Cantidad de peticiones concurrentes (p)	Indicadores	Tipo de contenido		
		Estático	Dinámico	
			PHP	Python
p≤500	Rendimiento	Nginx	Nginx	Nginx
	Recursos	Nginx	Nginx	Nginx
	Capacidad	Apache	Nginx	Proxy
	Todos	Nginx	Nginx	Nginx
500<p≤1000	Rendimiento	Nginx	Nginx	Proxy
	Recursos	Nginx	Nginx	Nginx
	Capacidad	Apache	Nginx	Nginx/Proxy
	Todos	Nginx	Nginx	Nginx
1000<p≤10000	Rendimiento	Apache	Proxy	Proxy
	Recursos	Nginx	Nginx	Nginx
	Capacidad	Nginx	Proxy	Nginx/Proxy
	Todos	Nginx	Proxy	Nginx
p>10000	Rendimiento	Nginx	Proxy	Proxy
	Recursos	Nginx	Nginx	Nginx
	Capacidad	Apache/Nginx	Nginx	Nginx/Proxy
	Todos	Nginx	Nginx	Nginx
Todas	Rendimiento	Nginx	Proxy	Proxy
	Recursos	Nginx	Nginx	Nginx
	Capacidad	Apache/Nginx	Nginx	Nginx/Proxy
	Todos	Nginx	Nginx	Nginx

Teniendo en cuenta todos los indicadores, el servidor web seleccionado en todos los escenarios es Nginx excepto para contenido dinámico con PHP y concurrencia de peticiones entre 1 000 y 10 000, que el seleccionado es Proxy. En cuanto al consumo de recursos el más eficiente en todos los casos es Nginx. Teniendo en cuenta el rendimiento y la capacidad, los tres servidores son elegidos respectivamente en algunos escenarios. Partiendo de la teoría definida, a continuación se desarrollará la solución como parte de la Herramienta para la Migración y Administración de Servicios Telemáticos.

Solución informática para la selección del servidor web

La solución informática de la presente investigación se sustenta en añadir a la Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST) un componente Web que permita seleccionar el servidor web que se ajuste a la institución, a partir del resultado obtenido con la aplicación del estudio de caso del epígrafe anterior.

Herramienta para la Migración y Administración de Servicios Telemáticos: en su versión 2.0 contiene varios módulos, dos de ellos corresponden a los servidores web Apache 2 y Nginx respectivamente. La herramienta permite la administración remota a través del protocolo SSH (*Secure SHell*) de diferentes PC servidoras. La instalación de HMAST en una PC diferente a la que está instalado el servidor, evita el consumo adicional de recursos. La característica de modularidad de la herramienta permite agregar más módulos, sin que esto afecte a los restantes, en caso que en las instituciones se necesite el empleo de otro servidor web. Existen dos funcionalidades que se aplican a todos los módulos: Aplicar cambios al servidor que permite guardar las configuraciones locales del servicio en la PC servidora y Descartar cambios locales que permite descartar las configuraciones locales, cambiando estas por las configuraciones de la PC servidora.

La herramienta presenta una arquitectura N-Capas orientada al dominio, distribuida en cinco componentes o paquetes: Presentación, Aplicación, Dominio, Persistencia e Infraestructura Transversal. La interacción entre los mismos se realiza a través de interfaces y utilizando inyección de dependencias (PALMA, 2013). La capa Dominio es el corazón del software, sus componentes implementan la funcionalidad principal del sistema y encapsulan toda la lógica de negocio relevante (DE LA TORRE, et al., 2010). En las capas de Presentación, Aplicación, Dominio y Persistencia se inserta un paquete llamado Web que a su vez contiene dos paquetes, Apache2 y Nginx, con todo lo referente a cada módulo respectivamente.

Componente Web: como solución de la presente investigación, en lugar de tener los módulos Apache 2 y Nginx aisladamente, se inserta en HMAST un componente Web que contiene dentro ambos módulos (ver Figura 3), que inicialmente están desactivados. Cuando se accede por primera vez al componente, se deben especificar los elementos necesarios para la selección del servidor web:

- Cantidad total de usuarios que acceden al contenido: (número entero positivo)
- Tipo de contenido: Estático/Dinámico
- En caso de ser dinámico especificar lenguaje de programación: PHP/Python
- Medida de la eficiencia que desea priorizar: Todos/Rendimiento/Utilización de los recursos/Capacidad

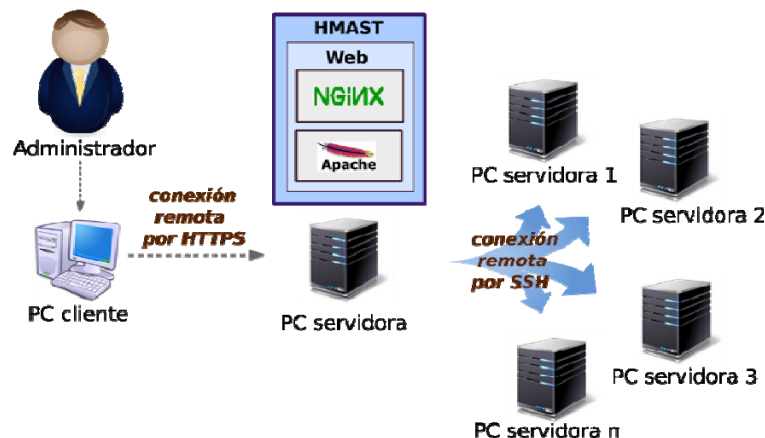


Figura 3: Esquema de funcionamiento de HMAST (Fuente: elaboración propia).

Posteriormente se presiona el botón *Mostrar resultado*, la herramienta tomando como referencia los 60 escenarios obtenidos, visualiza el nombre del servidor web más indicado según los elementos especificados anteriormente. Se muestra la opción *Instalar servidor web*, que permite instalar el servidor seleccionado o ambos servidores en caso de ser Proxy la opción resultante. En este último caso, el sistema instala primero Apache 2 y se habilita solo el puerto 8080 escuchando por 127.0.0.1 para posteriormente instalar Nginx y que no existan conflictos. El usuario puede desinstalar el servidor web en el momento deseado. A continuación se describen los módulos Apache 2 y Nginx.

- El módulo Apache 2 posee las funcionalidades para: iniciar, reiniciar, recargar y detener el servidor; activar el MPM a usar y configurar parámetros en cada uno (*prefork*, *worker* y *event*); especificar parámetros en el servidor principal; especificar parámetros en el manejo de conexiones; habilitar y deshabilitar los módulos a usar en Apache 2; modificar puertos para las conexiones y gestionar permisos de acceso a directorios y ficheros.
- El módulo Nginx cuenta con las funcionalidades necesarias para: iniciar, detener, reiniciar o recargar el servidor web Nginx; funcionar como proxy inverso; configurar las conexiones en el servidor; controlar el tráfico HTTP; adicionar, editar, eliminar, habilitar y deshabilitar los hosts virtuales; configurar los permisos de acceso y alias para cada host virtual (MOLINA, 2017).

Evaluación de la solución

La evaluación de la solución tiene la finalidad de verificar el cumplimiento del objetivo propuesto, que es aumentar la

eficiencia en la selección de los servidores web Apache 2 y Nginx durante el proceso de migración a código abierto. A continuación se describen los métodos aplicados.

Estudio de caso en ECOAIND3: se realizó un estudio de caso en la Empresa Constructora de Obras de Arquitectura e Industriales No. 3, Contingente “Ñico López”, que tuvo como objetivo comparar la eficiencia del servidor web seleccionado en dos proyectos de migración a código abierto: el proyecto de migración a aplicaciones de código abierto (2014 a 2015) y el proyecto de actualización a la migración de los servidores (2018). En el primero sin aplicar la solución se seleccionó Apache2 y en el segundo aplicándola, el servidor web seleccionado fue Nginx debido a que la empresa posee un total de 200 usuarios y el contenido publicado es estático (repositorio de Nova y la actualización del antivirus). Para la medición de la eficiencia se empleó la Norma Cubana ISO/IEC 25023:2017. Se realizaron cuatro observaciones, utilizando la herramienta *ab* mediante la cual se crearon un total de 200 peticiones con concurrencias 10, 100, 150 y 200 respectivamente, desde una PC cliente al servidor web instalado en la PC servidora. Al mismo tiempo se realizó el monitoreo de los recursos de la PC servidora con la herramienta *dstat*. Finalmente se realizó la comparación de la eficiencia de ambos servidores web en cuanto a todos los indicadores. Como se observa en la Tabla 2, se muestra en negrita el valor resultante del servidor más eficiente, donde en siete indicadores Nginx supera a Apache 2 se concluye que con la aplicación de la solución en ECOAIND3 se aumentó la eficiencia en la selección del servidor web.

Tabla 2: Cálculo de la eficiencia en ECOAIND3 (Fuente: elaboración propia).

Indicadores	Eficiencia de Apache 2	Eficiencia de Nginx
Tiempo medio de conclusión de un trabajo (ms)	562,250	190,250
Adecuación del tiempo de conclusión de un trabajo (ms)	2,811	0,951
Rendimiento medio	1,783	5,264
La media de utilización del procesador	0,012	0,005
La media de utilización de la memoria	0,275	0,126
La media del uso de los dispositivos de E/S	0,090	0,090
Utilización del ancho de banda	0,647	0,543
Capacidad de procesamiento de transacciones	2,864	2,921
Capacidad de acceso de usuario	200	200

Método Delphi y Modelo de Torgerson: para la selección de los expertos se aplicó una encuesta a 10 especialistas en servicios telemáticos, a partir de la información expresada en la misma se obtuvo el nivel de competencia de cada experto, donde el 90% posee un nivel de competencia Alto y un 10% un nivel Medio. Para la evaluación de la solución, se realizó el envío a los expertos de un documento que describe la solución desarrollada y una encuesta, donde se solicitó que evaluaran cinco aspectos relacionados con la misma. La evaluación de los aspectos se basó en cinco categorías: Muy Adecuado (MA), Bastante Adecuado (BA), Adecuado (A), Poco Adecuado (PA) y Nada Adecuado (NA). Posteriormente se emplea el Modelo de Torgerson. A partir de las valoraciones de los expertos se calculó la frecuencia absoluta por aspecto, la frecuencia acumulada por aspecto, la frecuencia acumulada relativa y los Puntos de Corte, que se ubicaron en una recta numérica, clasificando de esta forma cada aspecto sometido a evaluación. Se concluye que de los cinco aspectos, tres se evalúan de Muy Adecuado (60%) y dos de Bastante Adecuado (40%), sin mostrarse ninguno de los casos como Adecuado, Poco adecuado o Nada adecuado. Se demostró la existencia de un 100% de concordancia con la solución.

Técnica de Iadov: se realizó una encuesta a 10 especialistas en servicios telemáticos pertenecientes al Centro de Software Libre de la UCI, para determinar su grado de satisfacción con la herramienta propuesta a partir de su experiencia en la migración y administración de servicios telemáticos. De los 10 especialistas encuestados, 7 respondieron clara satisfacción, 2 más satisfechos que insatisfechos y 1 indefinido, resultando el índice de satisfacción grupal (ISG) igual a 0,8, como se encuentra entre 0,5 y 1, indica que los especialistas en servicios telemáticos presentan satisfacción con la solución informática desarrollada.

Triangulación metodológica: se define como el uso de más de dos métodos para estudiar el mismo fenómeno bajo investigación. Se basa en el uso de métodos y análisis de recolección de datos tanto cualitativos como cuantitativos en el estudio del mismo fenómeno (HUSSEIN, 2009). La realización de la triangulación metodológica permitió determinar que tanto los métodos cualitativos como cuantitativos aplicados para evaluar la solución, arrojaron el mismo resultado satisfactorio evidenciándose el cumplimiento del objetivo.

Conclusiones

Al finalizar la presente investigación se concluye lo siguiente:

- El análisis de las fuentes bibliográficas relacionadas con los servidores web Apache 2 y Nginx, permitió incluir como alternativa para las instituciones cubanas la configuración de Nginx como proxy inverso de

Apache 2 y determinar que el correcto funcionamiento del servidor web depende de la arquitectura del servidor, la cantidad de peticiones concurrentes y el tipo de contenido publicado.

- El empleo de un estudio de caso aplicando la Norma Cubana ISO/IEC 25023:2017 y calculando la eficiencia a partir del tipo de contenido y la cantidad de peticiones concurrentes, permitió el desarrollo de un componente Web en la Herramienta para la Migración y Administración de Servicios Telemáticos, que selecciona el servidor web más eficiente en cuanto a 60 escenarios definidos.
- La validación de la solución mediante un estudio de caso en ECOAIND3 analizando dos proyectos de migración; el empleo del criterio de expertos a través del método Delphi y el modelo de Torgerson; la aplicación de la técnica de Iadov para medir el grado de satisfacción con la solución; y la triangulación metodológica; permitió evidenciar el cumplimiento del objetivo de la presente investigación.

Agradecimientos

Al Centro de Software Libre de la Universidad de las Ciencias Informáticas y especialmente al equipo de desarrollo de la Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST).

Referencias

- APACHE SOFTWARE FOUNDATION. Multi-Processing Modules (MPMs). [En línea]. 2017. [Consultado el: 20 de octubre de 2017]. Disponible en: [<http://httpd.apache.org/docs/2.4/mpm.html>].
- CACCIAGRANO, Diletta; CORRADINI, Flavio. On synchronous and asynchronous communication paradigms. En: 7th Italian Conference, ICTCS. Theoretical Computer Science. Torino, Italy: Springer, 2001, p. 256-268.
- CAMPOVERDE, Ariel M.; HERNÁNDEZ, Dixys L.; MAZÓN, Bertha E. Cloud computing con herramientas open-source para Internet de las cosas. *Maskana*, 2015, 6 (Número Especial): p. 173-182.
- DE LA TORRE, César, et al. Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0 (BETA). España, Krasis Consulting, 2010. 433 p.
- GARCÍA, María; LENA, Francisco. Aplicación del método delphi en el diseño de una investigación cuantitativa sobre el fenómeno FABLAB. *Empiria - Revista de metodología de ciencias sociales*, 2018, 2 (40): p. 129-166.
- GILSTER, Ron. *PC Hardware: A Beginner's Guide*. India, McGraw-Hill Education, 2001. 674 p.
- HERNÁNDEZ, Roberto; FERNÁNDEZ, Carlos; BAPTISTA, Pilar. *Metodología de la investigación*. Cuarta Edición. México, McGraw-Hill, 2006. 882 p.

- HUSSEIN, Ashatu. The use of triangulation in social sciences research: Can qualitative and quantitative methods be combined? *Journal of comparative social work*, 2009, 4 (1): p. 1-8.
- KABIR, Mohammed. *La Biblia del Servidor Apache 2*. España, Anaya Multimedia, 2003. 845 p.
- KHOLODKOV, Valery. *Nginx Essentials*. Birmingham, Packt Publishing Ltd, 2015. 151 p.
- MEJÍA, José Teodoro; GONZÁLES, María Isabel; ESPAÑA, Angel Rafael. Programming Algorithms of load balancing with HA-Proxy in HTTP services. *Journal of Science and Research: Revista Ciencia e Investigación*, 2018, 3 (CITT2017): p. 100-105.
- MOLINA, Rachel. *Módulo para administrar el servidor web Nginx desde la Herramienta para la Migración y Administración de Servicios Telemáticos*. Tesis para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana, 2017.
- NAM, Van. Comparative Performance Evaluation of Web Servers. *VNU Journal of Science: Computer Science and Communication Engineering*, 2017, 31 (3): p. 28–34.
- NEDELCO, Clément. *Nginx HTTP Server*. Third Edition. Birmingham, Packt Publishing, 2015. 318 p.
- NETCRAFT. Most Reliable Hosting Company Sites in November 2018. [En línea]. 2018. [Consultado el: 6 de diciembre de 2018]. Disponible en: [<https://news.netcraft.com/archives/2018/>].
- OFICINA NACIONAL DE NORMALIZACIÓN. NORMA CUBANA NC ISO/IEC 25023:2017. INGENIERÍA DE SOFTWARE Y SISTEMAS – REQUISITOS DE LA CALIDAD Y EVALUACIÓN DE SOFTWARE Y SISTEMAS (SQuaRE) – MEDICIÓN DE LA CALIDAD DEL PRODUCTO DE SOFTWARE Y SISTEMA. 2017.
- PALMA, Nurisel. *Módulo para la administración de los servidores web en HMAST*. Tesis para optar por el título de Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, La Habana, 2013.
- PÉREZ, Yoandy; GARCÍA, Abel; GOÑI, Angel. *Buenas Prácticas para la Migración a Código Abierto*. La Habana, Ediciones Futuro, 2015. 106 p.
- PÉREZ, Yoandy. *Estrategia para la migración a aplicaciones de código abierto*. Tesis para optar por el título de Máster en Informática Aplicada, Universidad de las Ciencias Informáticas, La Habana, 2015.
- RAKHMAWATI, Nur Aini, et al. A survey of web Technologies in Indonesia Local Governments. *Jurnal Sisfo*, 2018, 7 (3): p. 213-222.
- SONI, Rahul. *Nginx: From Beginner to Pro*. New York, Apress, 2016. 255 p.
- W3TECHS. Usage of web servers for websites. [En línea]. 2018. [Consultado el: 6 de diciembre de 2018]. Disponible en: [https://w3techs.com/technologies/overview/web_server/all].