Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

Tipo de artículo: Artículo original
Temática: Software Libre
Recibido: 04/04/2020 | Aceptado: 06/06/2020 | Publicado: 01/09/2020

Eficiencia de los servidores web Apache 2 y Nginx: un estudio de caso

Efficiency of Apache 2 and Nginx web servers: a case study

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 definir la eficiencia de los servidores web Apache 2 y Nginx utilizando los indicadores de eficiencia de la Norma Cubana ISO/IEC 25023:2017 en un estudio de caso. El estudio consistió en realizar con la herramienta Apache Benchmark un número determinado de peticiones con cierta concurrencia, desde una computadora cliente al servidor web instalado en la computadora servidora. Además al mismo tiempo se realizó el monitoreo de los recursos de la computadora servidora con la herramienta dstat. Los servidores web utilizados fueron Apache 2, Nginx y Nginx funcionando como proxy inverso de Apache 2. Se crearon diferentes escenarios de prueba variando la cantidad de peticiones concurrentes y el tipo de contenido publicado: estático, dinámico con PHP y dinámico con Python. Posteriormente se aplicó la medición de nueve indicadores de eficiencia según la Norma Cubana ISO/IEC 25023:2017: tiempo medio de conclusión de un trabajo, adecuación del tiempo de conclusión de un trabajo, rendimiento medio, media de utilización del procesador, media de utilización de la memoria, media del uso de los dispositivos de entrada/salida, utilización del ancho de banda, capacidad de procesamiento de transacciones y capacidad de acceso de usuario. Finalmente se determinó que Nginx es el servidor web más eficiente en cuanto a la utilización de los recursos para los tres tipos de contenido analizados. De los 27 escenarios analizados en cuanto a todas las observaciones, resultó ser Nginx el más eficiente en 18 de ellos, Apache en 1 y Proxy en 8.

Palabras clave: Apache 2, eficiencia, Nginx, estudio de caso.

Abstract
The present investigation focused on the objective of defining the efficiency of Apache 2 and Nginx web servers using the efficiency indicators of the Cuban Standard ISO / IEC 25023: 2017 in a case study. The study consisted of making a certain number of requests with certain concurrence with the Apache Benchmark tool, from a client computer to the web server installed on the server computer. In addition, at the same time, the server computer resources were monitored with the dstat tool. The web servers used were Apache 2, Nginx and Nginx working as Apache 2's reverse

proxy. Different test scenarios were created, varying the number of concurrent requests and the type of content published: static, dynamic with PHP and dynamic with Python. Subsequently, the measurement of nine efficiency indicators was applied according to the Cuban Standard ISO / IEC 25023: 2017: average time of completion of a job, adequacy of the time of completion of a job, average performance, average processor utilization, average utilization memory, average input / output device usage, bandwidth usage, transaction processing capacity, and user access capacity. Finally, it was determined that Nginx is the most efficient web server in terms of resource utilization for the three types of content analyzed. Of the 27 scenarios analyzed for all observations, Nginx turned out to be the most efficient in 18 of them, Apache in 1 and Proxy in 8.

Keywords: *Apache 2, efficiency, Nginx, case study.*

Introducción

Internet ha creado un nuevo escenario en el que las relaciones personales cobran protagonismo. Las nuevas plataformas y herramientas colaborativas han producido un cambio desde una web 1.0 basada en páginas estáticas, meramente informativas, sin capacidad de generar una participación del usuario, hacia una web dinámica donde se produce una interrelación que genera una suma de conocimientos y/o experiencias. La web 2.0 o web social son personas colaborando, compartiendo y participando en un canal multidireccional abierto que permite lograr la máxima interacción entre los usuarios y les ofrece nuevas posibilidades de colaboración, expresión y participación (Schiavi, 2013). Las redes sociales son una de las actividades en línea más populares, conectarse con familiares y amigos, expresar opiniones, entretenimiento y compras en línea, se encuentran entre las razones más populares para el uso de Internet (Statista, 2018).

Los servidores web desempeñan un papel dominante en la infraestructura de estos servicios. Su tarea principal es recibir y procesar solicitudes de clientes que exigen objetos específicos de los servidores y devolverlos mediante respuestas relacionadas. Los mensajes de solicitud y respuesta asociados se transportan mediante el Protocolo de Transferencia de Hipertexto (HTTP, del inglés Hypertext Transfer Protocol) o el Protocolo de Transferencia de Hipertexto Seguro (HTTPS, del inglés Hypertext Transfer Protocol Secure) mediante conexiones TCP (del inglés Transmission Control Protocol) entre los clientes y el servidor. El rendimiento resultante de la prestación de servicios depende crucialmente del procesamiento adecuado y eficiente de estas tareas (Van, Krieger y Chakka, 2008).

La arquitectura de software de un servidor web proporciona un servicio que establece cómo un usuario accede a través del protocolo HTTP a los archivos de datos almacenados en el servidor web. Existen tres arquitecturas: basadas en procesos, basadas en hilos y basadas en eventos. En las dos primeras la E/S (Entrada/Salida) se bloquea durante el manejo de la solicitud, por tanto el consumo de recursos se incrementa linealmente de acuerdo con el número

creciente de hilos o procesos. La tercera es mucho más flexible que las anteriores, con E/S sin bloqueo y manejo asíncrono de solicitudes (Nam, 2017).

En los servidores web si bien las características y los asuntos relacionados con la comunidad son importantes en general, el aspecto que puede marcar la diferencia es el rendimiento (Nedelcu, 2015). Una acción computacional típica en un entorno de servicio web es una solicitud. Cada solicitud requiere recursos, como mínimo consume memoria y ciclos de la unidad central de procesamiento (CPU, por sus siglas en inglés Central Processor Unit), pero con frecuencia también lee y escribe algunos datos en otros dispositivos, como unidades de disco, lo que introduce un mayor costo de E/S. El agotamiento de cualquier recurso requerido para atender una solicitud lleva a la creación del llamado cuello de botella de rendimiento (Ligus, 2013). La prestación exitosa de los servicios de un servidor web depende de su rendimiento, el cual es la proporción entre el producto o el resultado obtenido y los medios utilizados (Real Academia Española, 2018).

Actualmente son numerosos los servidores web, sin embargo Apache 2 y Nginx son los dos servidores web de código abierto más usados (Netcraft, 2018; W3Techs, 2018). Nginx se caracteriza por poseer una arquitectura orientada a eventos para el manejo de peticiones, presentar un sistema de módulos estático, funciona como proxy inverso y posee soporte para hosts virtuales basados en IP (del inglés *Internet Protocol*) y/o hosts virtuales basados en nombre. Por otra parte Apache 2 se caracteriza por la introducción de Módulos de Multiprocesamiento (MPM, del inglés Multiprocessing Modules) en el manejo de peticiones, un sistema de módulos dinámico, funciona como servidor proxy caché y posee soporte para hosts virtuales basados en IP y/o hosts virtuales basados en nombre. Además se puede configurar Nginx como proxy inverso de Apache 2. 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) (Nedelcu, 2015).

En Cuba también se evidencia el gran uso de Apache 2 y Nginx. En el proceso de migración a software libre y aplicaciones de código abierto, constituyen las alternativas libres a servidores web privativos para PYMES (Pequeñas y Medianas Empresas) y grandes empresas (Pérez, García y Goñi, 2015), sin embargo no está definido cuál de los dos es más eficiente. A partir de los procesos de migración realizaos por la Universidad de las Ciencias Informáticas en entidades como la Empresa Constructora de Obras de Arquitectura e Industriales No.3 (ECOAIND3) y el diagnóstico en 10 ministerios con vísperas a una futura migración, se determinó que las PC servidoras[1] en ocasiones presentan

[1] Computadoras en red que realizan una tarea especial para atender las necesidades de recursos de las estaciones de trabajo (clientes) en una red (Gilster, 2001).

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

bajas prestaciones en cuanto a los recursos de hardware. Por tal motivo se necesita la selección del servidor web más eficiente en cuanto al entorno empresarial cubano.

La eficiencia es una de las características del software incluidas en la Norma Cubana ISO/IEC 25023:2017 (Oficina Nacional de Normalización, 2017), la cual abarca nueve indicadores y se mide a partir de tres subcaracterísticas: rendimiento, utilización de los recursos y capacidad. El objetivo general de la presente investigación está centrado en definir la eficiencia de los servidores web Apache 2 y Nginx utilizando los indicadores de eficiencia de la Norma Cubana ISO/IEC 25023:2017 en un estudio de caso.

Materiales y métodos o Metodología computacional

La eficiencia de los servidores web Apache 2 y Nginx implica el desempeño adecuado de los mismos. 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 caso. Los estudios de casos 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. La unidad o caso investigado puede tratarse de un individuo, una familia, un objeto, un sistema, una organización, etc. (Hernández, Fernández y Baptista, 2006). A continuación se definen elementos que caracterizan el estudio de caso a realizar.

Objetivo del estudio: definir la diferencia entre los servidores web Apache 2 y Nginx en cuanto a la eficiencia de desempeño, teniendo en cuenta los indicadores de la Norma Cubana ISO/IEC 25023:2017.

Unidad de análisis: eficiencia de Apache 2 y Nginx.

Descripción: se realiza un estudio preexperimental, que consiste en administrar un tratamiento a un grupo y después aplicar una medición de una o más variables, para observar cuál es el nivel del grupo en estas variables (Hernández, Fernández y Baptista, 2006). El estudio se basa en los elementos de los que depende el funcionamiento del servidor web: arquitectura del servidor (se evidencia en el servidor web en cuestión), la cantidad de peticiones concurrentes y el tipo de contenido publicado. 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 nueve 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. Las características de ambas PC se aprecian en la Tabla 1.

Tabla 1: Características de las dos PC empleadas en el estudio de caso (Fuente: elaboración propia).

Identificador	**RAM**	**CPU**	**HDD**	**Sistema Operativo**
PC-Servidora	8 GB	Intel Core i3-2120 de 3.30 GHz y 4 procesadores	1024 GB	Nova Servidores 6.0
PC-Cliente	4 GB	Intel Core i3-2120 de 3.30 GHz y 4 procesadores	1024 GB	Nova Escritorio 6.0

Para la ejecución del proceso anteriormente descrito se emplearon las herramientas ab y dstat.

- **ab** es una herramienta para la evaluación comparativa del servidor web HTTP (Kunda, Chihana y Sinyinda, 2017). Está diseñada para dar una impresión de cómo funciona el servidor web, mostrando especialmente cuántas solicitudes por segundo puede servir. Mediante la misma se realizan un total de peticiones n, con una concurrencia c (donde el valor máximo que acepta es 20 000) a una URL determinada.
- **dstat** es una herramienta versátil para generar estadísticas de recursos del sistema, es un reemplazo versátil para iostat, ifstat y vmstat. Dstat supera algunas de las limitaciones y agrega algunas características adicionales. Permite ver todos los recursos del sistema al instante, también proporciona inteligentemente la información más detallada en columnas e indica claramente en qué magnitud y unidad se muestra la salida. Dstat permite que los datos se escriban directamente en un archivo CSV para ser importados y utilizados por OpenOffice, Gnumeric o Excel para crear gráficos. Dstat combina las funciones de varias herramientas en un único paquete (Höbel, 2013).

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 describe cada uno.

- **p≤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.

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

- **500<p≤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.
- **1000<p≤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.
- **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.
- **Todas:** se refiere a todas las concurrencias de peticiones incluyendo las 10 observaciones.

Resultados y discusión

A continuación se describen los resultados del estudio, para cada uno de los nueve indicadores analizados.

Tiempo medio de conclusión de un trabajo

El tiempo medio de conclusión de un trabajo se calcula como la media del tiempo de respuesta a una petición (en milisegundos), para una muestra de 10 observaciones. Este elemento se determinó con la salida de la herramienta ab para el parámetro *Time per request*, que expresa el tiempo en milisegundos (ms). Apache 2 y Nginx se demoran más tiempo para responder cuando es tipo de contenido dinámico, que para estático. Para contenido estático y dinámico con PHP, en todos los servidores se mantiene el tiempo de respuesta estable y aumenta el valor en las últimas observaciones. Para contenido dinámico con Python el tiempo de respuesta para los tres servidores es notablemente mucho mayor que para los restantes tipos de contenido. El cálculo de la eficiencia para este indicador se muestra en la Tabla 2, donde para todas las observaciones con contenido estático el servidor web más eficiente es Nginx y para ambos casos de contenido dinámico es Proxy. Para cada escenario de concurrencia aparece en negrita el valor resultante del servidor más eficiente, un valor menor es mejor.

Tabla 2: Eficiencia del tiempo medio de conclusión de un trabajo (Fuente: elaboración propia).

Cantidad de Peticiones Concurrentes (p)	Tipo de contenido							
	Estático		Dinámico					
			PHP			Python		
	Apache	Nginx	Apache	Nginx	Proxy	Apache	Nginx	Proxy
p≤500	0,212	**0,210**	0,604	**0,597**	0,601	4,482	**3,937**	3,987
500<p≤ 1000	0,217	**0,214**	0,637	**0,599**	0,610	6,167	5,732	**5,571**
1000<p≤10000	**0,460**	0,494	0,871	0,760	**0,643**	6,719	6,180	**5,932**

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

p>10000	0,817	**0,653**	2,049	1,688	**1,091**	7,756	7,211	**6,464**
Todas	0,383	**0,356**	0,953	0,848	**0,709**	5,921	5,400	**5,188**

Adecuación del tiempo de conclusión de un trabajo

La adecuación del tiempo de conclusión de un trabajo se refiere a qué tan bien el tiempo de respuesta cumple los objetivos especificados. Se calcula como el cociente entre el tiempo medio de conclusión de un trabajo medido por el indicador anterior y el tiempo de respuesta especificado, que es 200 ms. El cálculo de la eficiencia para este indicador se muestra en la Tabla 3, donde para todas las observaciones con contenido estático el servidor web más eficiente es Nginx y para ambos casos de contenido dinámico es Proxy. Para los tres tipos de contenido los servidores web disminuyen su eficiencia con el aumento de la concurrencia de peticiones. Para cada escenario de concurrencia, se muestra en negrita el valor resultante del servidor más eficiente, un valor menor es mejor.

Tabla 3: Eficiencia de la adecuación del tiempo de conclusión de un trabajo (Fuente: elaboración propia).

Cantidad de Peticiones Concurrentes (p)	**Tipo de contenido**							
	Estático		**Dinámico**					
			PHP			**Python**		
	Apache	**Nginx**	**Apache**	**Nginx**	**Proxy**	**Apache**	**Nginx**	**Proxy**
p≤500	0,00106	**0,00105**	0,00302	**0,00298**	0,00300	0,02241	**0,01969**	0,01994
500<p≤ 1000	0,00109	**0,00107**	0,00319	**0,00299**	0,00305	0,03083	0,02866	**0,02785**
1000<p≤10000	**0,00230**	0,00247	0,00435	0,00380	**0,00322**	0,03359	0,03090	**0,02966**
p>10000	0,00409	**0,00326**	0,01024	0,00844	**0,00545**	0,03878	0,03606	**0,03232**
Todas	0,00192	**0,00178**	0,00476	0,00424	**0,00354**	0,02961	0,02700	**0,02594**

Rendimiento medio

El rendimiento medio se refiere a la cantidad de peticiones que el servidor web puede atender en un segundo. Se calcula como se muestra en la ecuación I.

(I) $$X = \sum_{i=1}^{n} (A_i / B_i)/n$$

A_i= Número de trabajos completados durante la i-ésimo tiempo de observación

B_i= i-ésimo período de tiempo de observación

n = Número de observaciones

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

El número de peticiones se determinó con la salida de la herramienta *ab* para el parámetro *Completed requests* (cuando se completaron todas las peticiones) o *Total of requests completed* (cuando no se completaron todas). El tiempo de la observación se determinó con la misma herramienta para el parámetro *Time taken for tests*, que expresa el tiempo en segundos (s). En la Figura 1 se muestra la gráfica con la cantidad de peticiones completadas por segundo, para 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 poseen un mayor rendimiento que para contenido dinámico, sin embargo en las últimas observaciones disminuyen la cantidad de peticiones completadas. Para contenido dinámico los tres servidores web poseen un mayor rendimiento con PHP con respecto a Python, en las últimas observaciones la cantidad de peticiones completadas decrece.

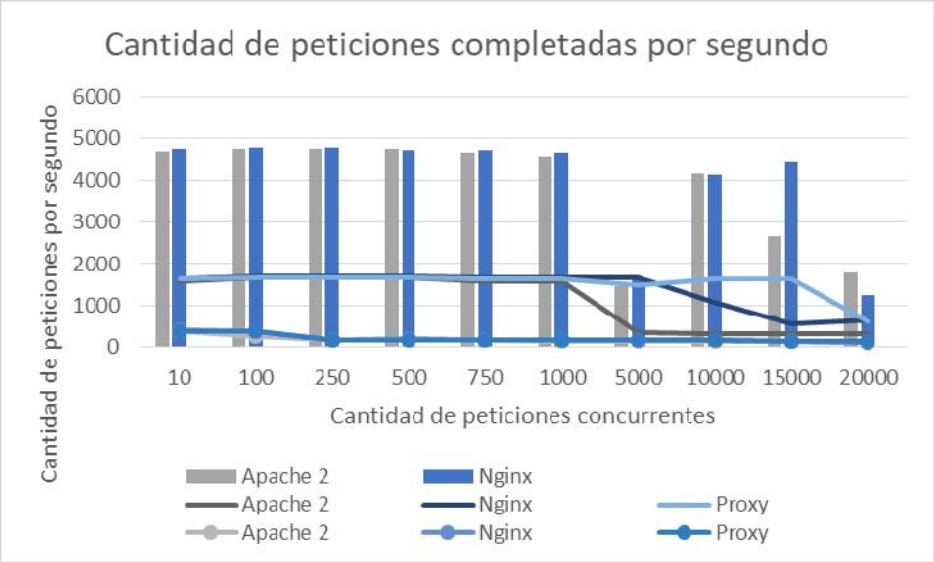

Figura 1: Cantidad de peticiones completadas por segundo (Fuente: elaboración propia).

El cálculo de la eficiencia para este indicador se muestra en la Tabla 4, donde para todas las observaciones con contenido estático el servidor web más eficiente es Nginx y para ambos casos de contenido dinámico es Proxy. Para cada escenario de concurrencia aparece en negrita el valor del servidor más eficiente, un valor mayor es mejor.

Tabla 4: Eficiencia del rendimiento medio (Fuente: elaboración propia).

	Tipo de contenido

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

Cantidad de Peticiones Concurrentes (p)	Estático		Dinámico					
			PHP			Python		
	Apache	Nginx	Apache	Nginx	Proxy	Apache	Nginx	Proxy
p≤500	4729,3	**4764,6**	1655,5	**1676,0**	1665,1	251,2	**291,0**	288,0
500<p≤ 1000	4607,7	**4688,8**	1569,4	**1671,7**	1640,5	162,3	174,6	**179,6**
1000<p≤10000	2813,0	**2894,8**	344,0	1377,9	**1560,5**	148,9	161,8	**168,6**
p>10000	2219,3	**2846,0**	315,5	596,2	**1136,8**	129,5	139,9	**154,8**
Todas	3819,7	**3991,8**	1108,0	1399,5	**1533,6**	188,6	211,7	**215,8**

La media de utilización del procesador

La media de utilización del procesador se refiere al tiempo del procesador que se utiliza para ejecutar un determinado conjunto de tareas en comparación con el tiempo de operación. Se calcula como se muestra en la ecuación II.

(II) $$X = \sum_{i=1}^{n} (A_i/B_i)/n$$

A_i= Tiempo del procesador realmente utilizado para ejecutar un conjunto dado de tareas en la observación i

B_i= Tiempo de operación para realizar las tareas en la observación i

n = Número de observaciones

El tiempo del procesador realmente utilizado se determinó con la opción --top-cputime-avg de dstat y el tiempo de operación con la opción Time taken for tests de ab. 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. El cálculo de la eficiencia para este indicador se muestra en la Tabla 5, donde para todas las observaciones en cada tipo de contenido el servidor web más eficiente es Nginx. Para cada escenario de concurrencia aparece en negrita el valor resultante del servidor más eficiente, un valor menor es mejor.

Tabla 5: Eficiencia de la media de utilización del procesador (Fuente: elaboración propia).

Cantidad de Peticiones Concurrentes (p)	Tipo de contenido							
	Estático		Dinámico					
			PHP			Python		
	Apache	Nginx	Apache	Nginx	Proxy	Apache	Nginx	Proxy

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

p≤500	0,0649	**0,0195**	0,0325	**0,0072**	0,0092	0,0044	**0,0031**	0,0032
500<p≤ 1000	0,0559	**0,0061**	0,0408	0,0084	**0,0048**	0,0048	**0,0013**	0,0020
1000<p≤10000	0,0375	**0,0116**	0,0338	**0,0076**	0,0168	0,0048	**0,0020**	0,0026
p>10000	0,0349	**0,0199**	0,5057	**0,0071**	0,0221	0,0050	**0,0027**	0,0032
Todas	0,0516	**0,0153**	0,1291	**0,0075**	0,0124	0,0047	**0,0024**	0,0028

La media de utilización de la memoria

En cuanto a la utilización de la memoria, durante la ejecución de la prueba no se usó la swap, la capacidad de consumo reportada fue de 0 MB tanto para contenido estático como dinámico con PHP y Python. Este indicador se refiere a la Memoria de Acceso Aleatorio (RAM, del inglés Random Access Memory) y la eficiencia se calcula como se muestra en la ecuación III.

(III) $$X = \sum_{i=1}^{n} (A_i / B_i)/n$$

A_i = Tamaño de la memoria realmente utilizada para realizar un determinado conjunto de tareas para el procesamiento de la i-ésima muestra

B_i = Tamaño de memoria disponible para realizar las tareas durante el procesamiento de la i-ésima muestra

n = Número de muestras procesadas

La memoria realmente utilizada se obtuvo a través de la opción –m de dstat, específicamente la columna para la memoria usada y la memoria disponible es la capacidad total de RAM de la PC servidora (8192 MB). Para obtener el consumo de memoria en cada observación, se calculó la media de todos los números durante el tiempo de la prueba, obteniendo un valor promedio para cada servidor en dependencia del tipo de contenido y de la cantidad de peticiones concurrentes. Para contenido estático, los dos servidores web consumen menos RAM que para contenido dinámico. Apache 2 es el servidor que más memoria utiliza, además el consumo aumenta más a mayor concurrencia de peticiones. Para contenido dinámico los tres servidores web utilizan más memoria con Python que con PHP. En ambos casos Apache utiliza notablemente más memoria que los demás servidores. El cálculo de la eficiencia para este indicador se muestra en la Tabla 6, donde para todas las observaciones con cada tipo de contenido el servidor web más eficiente es Nginx. Para cada escenario de concurrencia aparece en negrita el valor resultante del servidor más eficiente, un valor menor es mejor.

Tabla 6: Eficiencia de la media de utilización de la memoria (Fuente: elaboración propia).

	Tipo de contenido

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

Cantidad de Peticiones Concurrentes (p)	Estático		Dinámico					
			PHP			Python		
	Apache	Nginx	Apache	Nginx	Proxy	Apache	Nginx	Proxy
p≤500	0,059	**0,034**	0,083	**0,036**	0,042	0,090	**0,040**	0,056
500<p≤ 1000	0,061	**0,035**	0,144	**0,038**	0,047	0,157	**0,041**	0,061
1000<p≤10000	0,097	**0,043**	0,146	**0,055**	0,066	0,160	**0,056**	0,081
p>10000	0,149	**0,051**	0,135	**0,074**	0,116	0,150	**0,077**	0,131
Todas	0,085	**0,039**	0,118	**0,048**	0,063	0,129	**0,051**	0,077

La media del uso de los dispositivos de entrada/salida (E/S)

La media del uso de los dispositivos de entrada/salida se refiere al período de tiempo durante el cual un sistema o un dispositivo están realmente funcionando, en este caso el disco duro. Se calcula como se muestra en la ecuación IV.

(IV) $$X = \sum_{i=1}^{n} (A_i / B_i)/n$$

A_i = Duración del tiempo ocupado de los dispositivos de E/S para realizar un conjunto dado de tareas para la i-ésima observación

B_i = Duración de las operaciones de E/S para realizar las tareas para la i-ésima observación

n = Número de observaciones

Se calculó a partir de la opción --disk-util de dstat, que devuelve la utilización del disco en porciento. Aplicando la regla de tres se sustituyó en la fórmula el cociente entre el porciento y 100, en lugar del cociente entre el tiempo ocupado de los dispositivos y la duración de las operaciones de E/S. En los tres tipos de contenido el valor para cada servidor web se mantiene variable. El cálculo de la eficiencia para este indicador se muestra en la Tabla 7, donde para todas las observaciones en cada tipo de contenido el servidor web más eficiente es Nginx. Para cada escenario de concurrencia aparece en negrita el valor resultante del servidor más eficiente, un valor menor es mejor.

Tabla 7: Eficiencia del uso de los dispositivos de E/S (Fuente: elaboración propia).

Cantidad de Peticiones Concurrentes (p)	Tipo de contenido							
	Estático		Dinámico					
			PHP			Python		
	Apache	Nginx	Apache	Nginx	Proxy	Apache	Nginx	Proxy
p≤500	**0,01633**	0,02033	0,01503	**0,01490**	0,01633	0,01733	0,01798	**0,01728**

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

500<p≤ 1000	**0,01600**	0,02000	**0,01370**	0,01435	0,01880	0,01835	**0,00865**	0,01915
1000<p≤10000	0,02250	**0,01150**	0,01840	0,01600	**0,01340**	0,01965	**0,01545**	0,02120
p>10000	0,01870	**0,01500**	0,02070	**0,01770**	0,02230	0,02050	**0,01800**	0,02110
Todas	0,01797	**0,01743**	0,01657	**0,01557**	0,01743	0,01863	**0,01561**	0,01920

Utilización del ancho de banda

La utilización del ancho de banda se refiere a qué proporción del ancho de banda disponible se utiliza para realizar un conjunto dado de tareas. Se calcula como se muestra en la ecuación V.

(V) $X = A/B$ A = Ancho de banda de transmisión real medido a lo largo del tiempo para realizar un conjunto dado de tareas

B = Capacidad de ancho de banda disponible para realizar un determinado conjunto de tareas

El ancho de banda de transmisión real se obtuvo con la opción –n de dstat y la capacidad de ancho de banda disponible es 10 Mbps. Para contenido estático se utiliza más ancho de banda que para contenido dinámico, solo decrece en las últimas observaciones. Los tres servidores web realizan menor uso del ancho de banda para contenido dinámico con Python que cuando es con PHP. En la mayoría de las observaciones Nginx realiza menor consumo que los restantes servidores web. El cálculo de la eficiencia para este indicador se muestra en la Tabla 8, donde para Todas las observaciones en cada tipo de contenido el servidor web más eficiente es Nginx. Para cada escenario de concurrencia aparece en negrita el valor resultante del servidor más eficiente, un valor menor es mejor.

Tabla 8: Eficiencia de la utilización del ancho de banda (Fuente: elaboración propia).

Cantidad de Peticiones Concurrentes (p)	**Tipo de contenido**							
	Estático		**Dinámico**					
			PHP			**Python**		
	Apache	**Nginx**	**Apache**	**Nginx**	**Proxy**	**Apache**	**Nginx**	**Proxy**
p≤500	2,834	**2,742**	1,748	**1,619**	1,660	0,211	**0,208**	0,218
500<p≤ 1000	2,764	**2,695**	1,665	**1,411**	1,624	0,131	**0,074**	0,134
1000<p≤10000	1,763	**1,603**	1,547	**1,262**	1,434	0,140	**0,067**	0,116
p>10000	**1,845**	1,940	1,476	**1,195**	1,468	0,153	**0,070**	0,118
Todas	2,408	**2,345**	1,637	**1,421**	1,569	0,170	**0,125**	0,161

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

Capacidad de procesamiento de transacciones

La capacidad de procesamiento de transacciones se refiere a las transferencias realizadas en un segundo. Se calcula como se muestra en la ecuación VI.

(VI) $X = A/B$ A = Número de transacciones realizadas durante el tiempo de observación

B = Duración de la observación

El tiempo de la observación se obtuvo con la opción Time taken for tests de ab y la cantidad de transacciones con la salida de la herramienta dstat para la opción --disk-tps, que devuelve las transacciones de disco por segundo (tps). En los tres tipos de contenido el valor para cada servidor web se mantiene variable. El cálculo de la eficiencia para este indicador se muestra en la Tabla 9, donde para todas las observaciones con contenido estático el servidor web más eficiente es Apache 2 y para contenido dinámico con ambos lenguajes de programación es Nginx. Para cada escenario de concurrencia aparece en negrita el valor resultante del servidor más eficiente, un valor menor es mejor.

Tabla 9: Eficiencia de la capacidad de procesamiento de transacciones (Fuente: elaboración propia).

Cantidad de Peticiones Concurrentes (p)	Tipo de contenido							
	Estático		Dinámico					
			PHP			Python		
	Apache	Nginx	Apache	Nginx	Proxy	Apache	Nginx	Proxy
p≤500	**1,65**	2,92	2,63	**1,67**	3,35	2,55	2,18	**2,16**
500<p≤ 1000	**1,33**	1,67	2,43	**1,88**	2,19	2,74	**2,00**	2,71
1000<p≤10000	2,56	**1,35**	3,14	2,42	**1,55**	3,16	**2,00**	2,86
p>10000	**2,60**	4,19	3,09	**2,70**	4,45	3,40	**2,53**	3,11
Todas	**1,96**	2,61	2,78	**2,06**	2,98	2,92	**2,18**	2,60

Capacidad de acceso de usuario

La capacidad de acceso de usuario se refiere a la capacidad máxima de usuarios que acceden en un momento determinado. Se calcula como se muestra en la ecuación VII.

(VII) $X = \sum_{i=1}^{n} A_i/n$ A_i = Número máximo de usuarios que pueden acceder simultáneamente al sistema en la i-ésima observación

n = Número de observaciones

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 13, No. 9, Mes Septiembre, 2020
ISSN: 2306-2495| RNPS: 2343
Pág. 97-112
http://publicaciones.uci.cu

Para contenido estático, Apache 2 y Nginx presentan mayor capacidad de acceso de usuario que para contenido dinámico. Para contenido dinámico con ambos lenguajes de programación los tres servidores web disminuyen la capacidad de acceso de usuario en las últimas observaciones, sin embrago con Python presentan menor capacidad que con PHP. El cálculo de la eficiencia para este indicador se muestra en la Tabla 10, donde para todas las observaciones con contenido estático el servidor más eficiente es Nginx y para contenido dinámico con ambos lenguajes de programación es Proxy. Para cada escenario de concurrencia aparece en negrita el valor resultante del servidor más eficiente, un valor mayor es mejor, en este caso existen escenarios con más de un servidor con la misma eficiencia.

Tabla 10: Eficiencia para la capacidad de acceso de usuario (Fuente: elaboración propia).

Cantidad de Peticiones Concurrentes (p)	**Tipo de contenido**							
	Estático		**Dinámico**					
			PHP			**Python**		
	Apache	**Nginx**	**Apache**	**Nginx**	**Proxy**	**Apache**	**Nginx**	**Proxy**
p≤500	**50000,0**	**50000,0**	**50000,0**	**50000,0**	**50000,0**	49369,3	49438,3	**50000,0**
500<p≤ 1000	**50000,0**	**50000,0**	**50000,0**	**50000,0**	**50000,0**	47907,0	48481,0	**50000,0**
1000<p≤10000	**50000,0**	**50000,0**	45138,5	**50000,0**	**50000,0**	43817,0	49847,5	**49938,5**
p>10000	49998,5	**50000,0**	37044,0	38832,5	**49984,5**	36411,5	46890,0	**47879,0**
Todas	49999,7	**50000,0**	46436,5	47766,5	**49996,9**	45374,8	48819,0	**49563,5**

Conclusiones

El cálculo de la eficiencia de los servidores web Apache 2, Nginx y Nginx como proxy inverso de Apache 2 (denominado Proxy), a partir del tipo de contenido y la cantidad de peticiones concurrentes, permitió concluir lo siguiente:

- En cuanto al rendimiento los servidores web más eficientes son Nginx con contenido estático y Proxy con contenido dinámico con ambos lenguajes de programación PHP y Python.
- En cuanto a la utilización de los recursos el servidor web más eficiente es Nginx para los tres tipos de contenido analizados, siendo este servidor web la opción indicada a seleccionar en instituciones cubanas donde las PC servidoras presentan bajas prestaciones en cuanto a los recursos de hardware.
- En cuanto a la capacidad, Nginx se destaca en tres escenarios, Proxy en dos y Apache en uno.

- De los 27 escenarios analizados en cuanto a todas las observaciones, resultó ser Nginx el más eficiente en 18 de ellos, Apache en 1 y Proxy en 8, siendo Nginx el servidor web más eficiente en la mayor cantidad de escenarios.

Referencias

- Gilster, R. PC Hardware: A Beginner's Guide. India, Tata McGraw-Hill Education, 2001. 674 p.

- Höbel, V. Paquete Estadístico: ajuste del sistema con herramientas de diagnóstico. Linux magazine, 2013, (92): p. 50-53.

- Kunda, D.; Chihana, S.; Sinyinda, M. Web Server Performance of Apache and Nginx: A Systematic Literature Review. Computer Engineering and Intelligent Systems, 2017, 8 (2): p. 43-52.

- Ligus, S. Effective Monitoring and Alerting. Sebastopol, O'Reilly Media, Inc., 2013. 164 p.

- Nam, V. Comparative Performance Evaluation of Web Servers. VNU Journal of Science: Computer Science and Communication Engineering, 2017, 31 (3): p. 28–34.

- Nedelcu, C. 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.

- Pérez, Y.; García, A.; Goñi, A. Buenas Prácticas para la Migración a Código Abierto. La Habana, Ediciones Fututo, 2015. 106 p.

- Real Academia Española. Diccionario de la lengua española. [En línea]. 2018. [Consultado el: 10 de marzo de 2018]. Disponible en: [http://dle.rae.es].

- Hernández, R.; Fernández, C.; Baptista, P. Metodología de la investigación. Cuarta Edición. México, McGraw-Hill, 2006. 882 p.

- Schiavi, P. La protección de los datos personales en las redes sociales. A&C-Revista de Direito Administrativo & Constitucional, 2013, 13 (52): p. 145-178.

- Statista. Number of internet users worldwide from 2005 to 2017 (in millions). [En línea]. 2018. [Consultado el: 13 de diciembre de 2018]. Disponible en: [https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/].

- Van, T.; Krieger, U.; Chakka, R. Performance modeling of an apache web server with a dynamic pool of service processes. Telecommunication Systems, 2008, 39 (2): p. 117-129.

- 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].