

Implementación de un balanceador de carga

Implementation of a balancer of load

Néstor Javier Rodríguez Castellanos¹
Hugo Alejandro Rodríguez Castiblanco²

Recibido: 17/05/2014 - Aceptado: 18/08/2014

Cómo citar este artículo: N. Rodríguez y H. Rodríguez “Implementación de un balanceador de carga”, *IngEam*, vol. 2, n.º 2, pp. 73-89, 2015

Resumen

El propósito de este artículo es exponer cómo se ejecuta la implementación de un balanceador de carga por medio de la herramienta HAProxy y servidores backend apache tomcat. La metodología utilizada para lograr la simulación se utilizó 4 máquinas virtuales en Virtualbox con sistemas de Centos 7. Por último, se concluye que la utilización de un balanceador de carga refuerzan la fundamentación teórica de la misma, pues su aplicabilidad y aporta a la experiencia para futuras implementaciones en los diseños de las arquitecturas cuando se requiera alta disponibilidad y mejorar el rendimiento de una aplicación con la distribución de cargas, también se ayuda a los arquitectos en la implementación de diseños de arquitecturas con menores costos y logran cumplir los requerimientos en este caso de disponibilidad y rendimiento.

Palabras clave: implementación de un balanceador de carga, herramienta HAProxy, servidores Backed Apache Tomcat

Abstract

The purpose of this article is to expose how to execute the implementation of a balancer of load by means of the tool HAProxy and servers backend apache tomcat. The utilized methodology to achieve the simulation utilized 4 virtual machines in Virtualbox with systems of Centos 7. Finally it is concluded that the utilization of a balancer of load reinforces the theoretical foundation of the same, because his applicability contributes to the experience for future implementations in the designs of architectures when high availability be required and improving the performance of an application with the distribution of loads,

¹ Ingeniero Telemático de la Universidad Distrital. Especialista en arquitectura empresarial de software Pontificia Universidad Javeriana. Correo electrónico: stonerjrc@gmail.com

² Ingeniero de sistemas de la Universidad del Quindío. Especialista en arquitectura empresarial de software Pontificia Universidad Javeriana. Correo electrónico: harcalejo@gmail.com

also the architects in the implementation of designs of architectures with minor costs are helped and they manage to meet the requests, in this case of availability and performance.

Key words: Implementation of a balancer of load, tool HAProxy, servers Backed Apache Tomcat

Introducción

Dentro del marco de referencias de las nuevas tecnologías para arquitectura de software se encuentran los balanceadores de carga, componentes que pueden ser de tipo hardware o software. Su principal función en el contexto de aplicaciones empresariales es cubrir requerimientos de alta disponibilidad y mejoras de desempeño. Como ejercicio práctico en este trabajo se documenta la implementación de un balanceador de carga, en este caso el HAProxy, una herramienta open source para el balanceo por software. Adicionalmente, se utilizan servidores backend apache Tomcat como contenedores de la página de prueba que responderá a las peticiones de los usuarios.

Para lograr una simulación de la infraestructura existen 4 máquinas virtuales en VirtualBox con sistemas operativos CentOS 7, una encargada de recibir las peticiones y enrutarlas con la implementación de HAProxy a las otras 3 máquinas que contienen los servidores Apache Tomcat con la página replicada. Con esto se busca fortalecer los conceptos vistos de virtualización y balanceo de carga vistos en la asignatura de Nuevas Tecnologías para Arquitectura de Software y tener experiencias prácticas que permitan proyectar este conocimiento de las organizaciones.

74

1. Arquitectura solución de la implementación del balanceador

A continuación se muestra la configuración en un modelo conceptual en donde se describen los componentes que interactúan en la solución:

Balanceador de carga HaProxy: Es una aplicación gratuita y de código abierto Linux, ofrece alta disponibilidad, balanceo de carga. Proxy para TCP y aplicaciones HTTP. Fue especialmente diseñado para sitios web de muy alto tráfico [1].

Apache Tomcat: es un servidor web con soporte de servlets y JSPs, puede funcionar como servidor web por sí mismo y es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad [2].

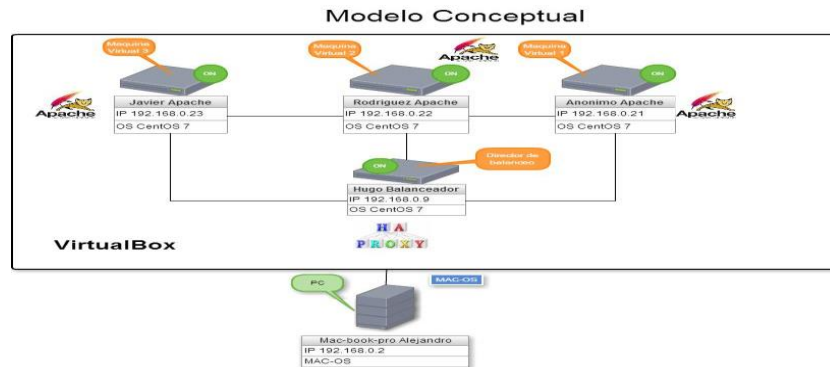


Ilustración 1. Modelo Conceptual

2. Proceso de instalación de las Máquinas Virtuales

Este proceso de instalación se realiza sobre una máquina virtual siguiendo los puntos que se enumeran a continuación. Luego, se procede a la clonación de este ambiente para los demás servidores necesarios:

- a) Descarga del Software VirtualBox (<https://www.virtualbox.org/>) Versión 5.0.0
- b) Instalación de software VirtualBox en Máquina MAC
- c) Descarga ISO CentOS <https://www.centos.org/download/>
- d) Ejecutar Máquina virtual
- e) Asignación de nombre a la maquina virtual
- f) Instalación de sistema operativo en la máquina virtual
- g) Finalización de instalación de sistema operativo
- h) Configuración de dirección IP

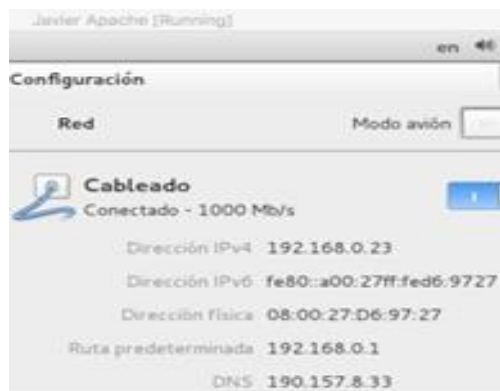


Ilustración 2. Configuración IP maquina Javier Apach



Ilustración 3. Configuración IP maquina Rodríguez Apache

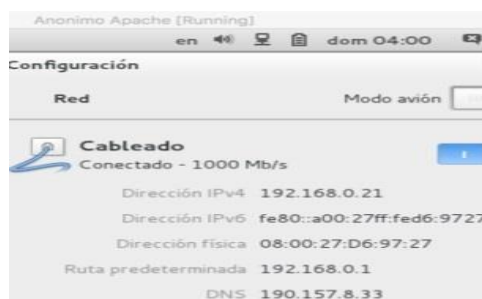


Ilustración 4. Configuración IP maquina Anónimo Apache

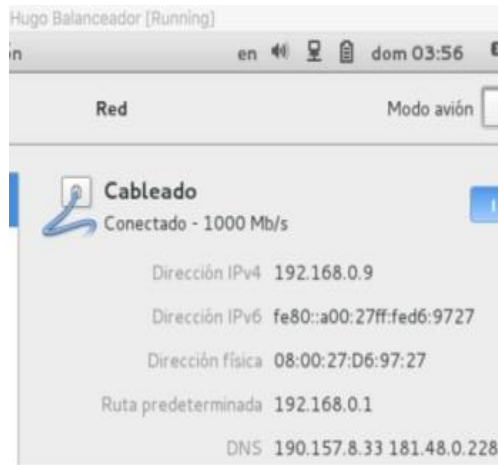


Ilustración 5. Configuración IP maquina Balanceador Apache

Tabla 1. Configuración Máquinas Virtuales

Función Máquina virtual	Nombre de la maquina	Sistema operativo	IP asignada
Balanceador	Hugo Balanceador	CentOS 7	192.168.0.9
Servidor Apache	Javier Apache	CentOS 7	192.168.0.23
Servidor Apache	Rodríguez apache	CentOS 7	192.168.0.22
Servidor Apache	Anónimo Apache	CentOS 7	192.168.0.21

77

Configuración final de las máquinas virtual

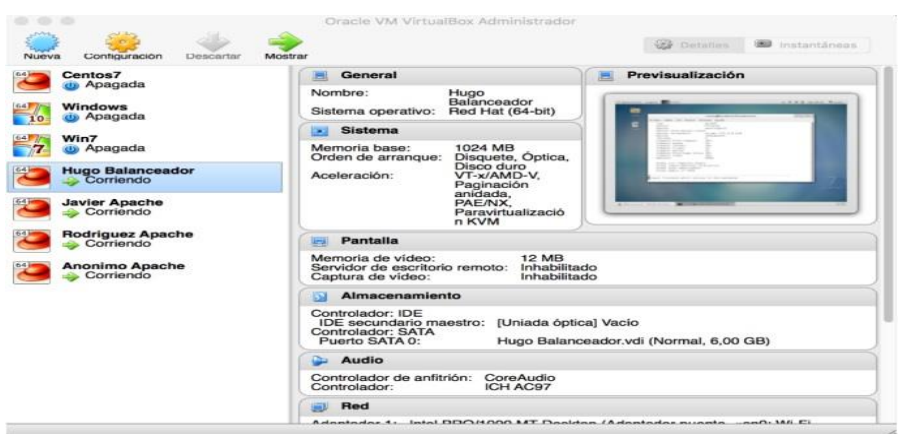


Ilustración 6. Configuración Máquinas Virtuales

3. Procedimiento instalación software apache en máquinas virtuales [3]

a) Instalación apache web server sobre sistema operativo centOS7

```
sudo yum clean all
```

```
sudo yum -y update
```

```
sudo yum -y install httpd
```

b) Permitir que apache funcione a través del firewall

```
sudo firewall-cmd --permanent --add-port=80/tcp
```

```
sudo firewall-cmd --permanent --add-port = 443 / tcp
```

c) Configurar apache para que Iniciar desde el Boot(arranque)

```
sudo systemctl enable httpd -- habilitar apache
```

```
sudo systemctl status httpd -- status apache
```

```
sudo systemctl stop httpd --detener apache
```

d) Despliegue de aplicación HTML.

Para cada una de las máquinas virtuales configuradas se realiza la página en HTML con su respectivo nombre en el Title del head para diferenciar que servidor se está atendiendo.

```
oracle@localhost:/var/www/html
Archivo Editar Ver Buscar Terminal Ayuda
<html>
  <head>
    <title>Javier Apache</title>
  </head>
  <body>
    <h1>===== BIENVENIDO =====</h1>
  </body>
</html>
~
```

Ilustración 7. Construcción página web

4. Instalación de HaProxy en la máquina virtual Hugo Balanceador [4]

- a) HaProxy no está disponible dentro de los repositorios por defecto del CentOS 7 por lo tanto se debe realizar la instalación por línea de comandos.

79

```
wget http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm -- Descargar el repositorio RPM EPEL.

-ivh rpm EPEL-release-6.8.noarch.rpm --Instalar el repositorio RPM EPEL para agregar su archivo de configuración Yum a su servidor.

yum install haproxy -- Instale HAProxy
```

- b) Configuración HaProxy [5]

Se debe realizar una configuración en el archivo HaProxy.cfg ubicado en la siguiente ruta:

```
vi /etc/haproxy/haproxy.cfg
```

```
[root@localhost oracle]# vi /etc/haproxy/haproxy.cfg
[root@localhost oracle]# sudo haproxy -f /etc/haproxy/haproxy.cfg
[root@localhost oracle]# sudo haproxy -f /etc/haproxy/haproxy.cfg
[root@localhost oracle]# vi /etc/haproxy/haproxy.cfg
```

Ilustración 8. Ubicación Archivo HaProxy.cfg

Los valores que debe contener son los siguientes de acuerdo con las asignaciones de IP y nombres de las maquinas, además se realiza la configuración para que aparezca el tablero de control o reporte estadísticos de HAProxy, con los comandos stats. Que se muestran a continuación en la ilustración 4.

La configuración inicial donde se definen tiempos de time out y opciones, pues las configuraciones que trae por defecto el HAProxy en su instalación. Esto esta seguido por configuración de stats donde se define el puerto, la uri y credenciales para acceder a su informe estadístico de peticiones [4].

Seguido de esto se encuentra la configuración del frontend, donde se define el tipo de entrada, en el caso http y el puerto por el que atenderá estas peticiones con la definición. Por último, se define el algoritmo y los servidores a los que se deben redireccionar las peticiones según el funcionamiento del algoritmo seleccionado.

80

```
option http-server-close
option forwardfor      except 127.0.0.0/8
option                 redispatch
retries                3
timeout http-request   10s
timeout queue          1m
timeout connect        10s
timeout client         1m
timeout server         1m
timeout http-keep-alive 10s
timeout check          10s
maxconn                3000

listen stats *:9090
stats uri /haproxy_stats
stats realm HAProxy\ Statistics
stats auth admin:admin
stats admin if TRUE

#-----
# main frontend which proxys to the backends
#-----
frontend http-in
    bind *:80
    default_backend app
#-----
# round robin balancing between the various backends
#-----
backend app
    balance roundrobin
    server Anonimo 192.168.0.21:8080 check
    server Rodriguez 192.168.0.22:8080 check
    server Javier 192.168.0.23:8080 check
```

Ilustración 9. Información de configuración archivo HaProxy

- c) Visualización de reporte de estadísticos de HAProxy

Esta visualización se realiza con todos los servidores de backend apagados, por eso en la tabla de app se pueden observar sus nombres y todos ellos en rojo.

The screenshot shows the HAProxy web interface. The 'stats' section displays overall statistics for the Frontend and Backend. The 'http-in' section shows request statistics. The 'app' section lists individual servers with their status (DOWN), last check time, and weight. A legend at the top explains the color coding for server states.

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntms	Thrtle
Frontend	1 1 -	1 1 3 000	4	1 364	21 332	0 0	1	0	0	0	0	0	0	0	0
Backend	0 0	0 1	0 1 300	1	0 0	0 0	0	0s	1 364	21 332	0 0	0 0	0 0	1	0

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntms	Thrtle
Frontend	0 0	0 0	0 0 3 000	0	0 0	0 0	0	0	0	0	0	0	0	0	0

Queue	Session rate	Sessions	Bytes	Denied	Errors	Warnings	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntms	Thrtle
Anonimo	0 0	0 0	0 0	0 0	0 0	0 0	27s DOWN	* L4TOUT in 2001ms	1	Y	-	1	1	27s	-
Rodriguez	0 0	0 0	0 0	0 0	0 0	0 0	26s DOWN	* L4TOUT in 2003ms	1	Y	-	1	1	26s	-
Javier	0 0	0 0	0 0	0 0	0 0	0 0	26s DOWN	L4TOUT in 2002ms	1	Y	-	1	1	26s	-
Backend	0 0	0 0	0 0 300	0	0 0	0 0	26s DOWN		0	0	0	0	1	26s	-

Ilustración 10. Información reporte estadístico de HAProxy

5. Pruebas de funcionalidad

- a) La primera prueba de funcionalidad a presentar es realizada tomando el balanceador de carga (Hugo Balanceador) y el servidor apache tomado de la Máquina virtual Javier apache con IP 192.168.0.23.

The screenshot shows the Oracle VM VirtualBox Administrator interface. On the left, a list of virtual machines is shown, with 'Hugo Balanceador' and 'Javier Apache' highlighted. The main window displays the configuration for 'Javier Apache', including system details like memory (1024 MB), disk, and hardware settings.

Ilustración 11. Interacción máquinas virtuales

En la siguiente ilustración se puede ver el cuadro de estadísticas de HAProxy, donde

se refleja que el servidor apache alojado en la máquina de Javier Apache está atendiendo las peticiones de HAProxy.

HAProxy version 1.5.4, released 2014/09/02

Statistics Report for pid 4221

> General process information

pid = 4221 (process #1, rbrproc = 1)
 uptime = 0s 0m49m08s
 system limits: memmax = unlimited, ulimit-n = 8035
 maxsock = 8035; maxconn = 4000; maxpipes = 0
 current conns = 6; current pipes = 0; conn rate = 81/sec
 Running tasks: 3/14, idle = 95 %

active UP backup UP
 active UP, going down backup UP, going down
 active DOWN, going up backup DOWN, going up
 active or backup DOWN not checked
 active or backup DOWN for maintenance (MAINT)
 active or backup SOFT STOPPED for maintenance
 Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
 • Scope
 • Hide "DOWN" servers
 • Refresh row
 • CSV export

External resource:
 • Primary all
 • Updates 0
 • Online ma

stats		Queue		Session rate			Sessions			Bytes		Denied		Errors			Warnings			Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	
Frontend	0	0	0	1	2	-	1	1	3 000	6			2 449	42 464	0	0	0	0	0	0	0	0	OPEN		0	0	0	0	0	0
Backend	0	0	0	0	2	-	0	1	300	3	0	0s	2 449	42 464	0	0	0	0	0	0	0	0	OPEN		0	0	0	0	0	0

http-in		Queue		Session rate			Sessions			Bytes		Denied		Errors			Warnings			Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	
Frontend	80	264	-	5	5	3 000	1 611			1 611			268 202	627 946	0	0	0	0	0	0	0	0	OPEN							

app		Queue		Session rate			Sessions			Bytes		Denied		Errors			Warnings			Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	
Anonimo	0	0	-	0	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	0	49m6s DOWN	L4TOUR in 2001ms	1	Y	-	1	1	49m6s
Rodriguez	0	0	-	0	0	0	0	0	-	0	0	?	0	0	0	0	0	0	0	0	0	0	49m6s DOWN	L4TOUR in 2002ms	1	Y	-	1	1	49m6s
Javier	0	0	-	80	264	-	4	5	-	1 611	1 611	0s	268 369	628 337	0	0	0	0	0	0	0	0	12m25s UP	L4OK in 1178ms	1	Y	-	6	2	20m55s
Backend	0	0	-	80	264	-	4	5	300	1 611	1 611	0s	268 369	628 337	0	0	0	0	0	0	0	0	12m25s UP		1	1	0	2	20m55s	

Ilustración 12 reporte de estadísticas acceso máquina virtual Javier Apache

Ejecución de comando tcpdump -n port 8080, que permite mostrar las peticiones TCP enviadas por la IP 192.168.0.9 y son atendidas por el puerto 8080 de la máquina 192.168.0.9

```

oracle@localhost:
Archivo Editar Ver Buscar Terminal Ayuda
IP 192.168.0.9.34021 > 192.168.0.23.webcache: Flags [R.],
nop,TS val 3194235 ecr 1877231, length 0
IP 192.168.0.9.34023 > 192.168.0.23.webcache: Flags [.],
877232, length 0
IP 192.168.0.9.34023 > 192.168.0.23.webcache: Flags [P.],
194235 ecr 1877232, length 197
IP 192.168.0.23.webcache > 192.168.0.9.34023: Flags [.],
3194235, length 0
IP 192.168.0.9.34024 > 192.168.0.23.webcache: Flags [S],
S val 3194237 ecr 0,nop,wscale 7, length 0
IP 192.168.0.23.webcache > 192.168.0.9.34024: Flags [S.],
mss 1460,sackOK,TS val 1877234 ecr 3194237,nop,wscale 7,
IP 192.168.0.9.34024 > 192.168.0.23.webcache: Flags [.],
877234, length 0
Val 3194120 ecr 0,nop,wscale 7, length 0
IP 192.168.0.23.webcache > 192.168.0.9.33921: Flags [S.],
s 1460,sackOK,TS val 1877117 ecr 3194120,nop,wscale 7, l
IP 192.168.0.23.webcache > 192.168.0.9.33920: Flags [P.],
1877117 ecr 3194120, length 391
IP 192.168.0.9.33921 > 192.168.0.23.webcache: Flags [.],
877117, length 0
IP 192.168.0.9.33920 > 192.168.0.23.webcache: Flags [.],
1877117, length 0
IP 192.168.0.23.webcache > 192.168.0.9.33920: Flags [F.],

```

Ilustración 13. Análisis de tráfico Balanceador - Maquina Javier Apache

A continuación se muestra la página de respuesta haciendo llamado a la aplicación expuesta en el servidor apache Javier Apache, además la ejecución de un comando que realiza 5 peticiones concurrentes durante 120 segundos a la Ip 192.168.0.9 por el puerto 80, que corresponde a la ubicación y configuración de HAProxy: “Siege”.

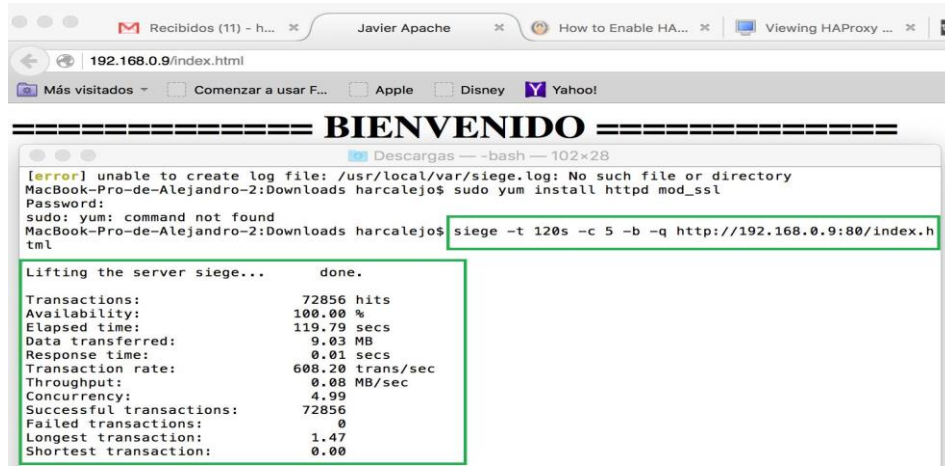


Ilustración 14. Comprobación Rendimiento y petición atendida por la máquina Javier Apache

- b) La siguiente prueba se realiza con la máquina virtual Javier apache IP 192.168.0.23 y Rodríguez Apache 192.168.0.22 encendidas.

83

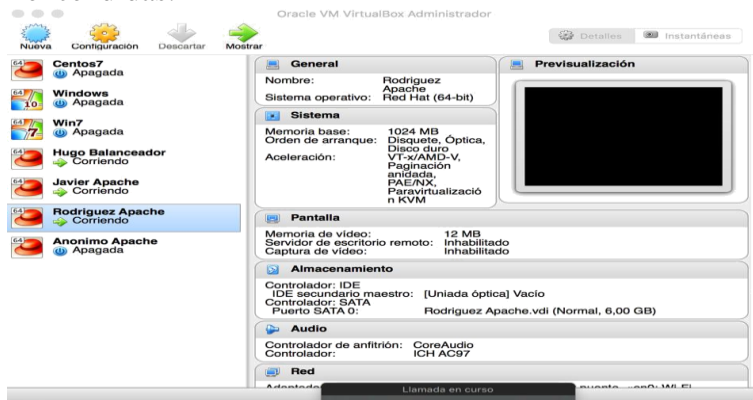
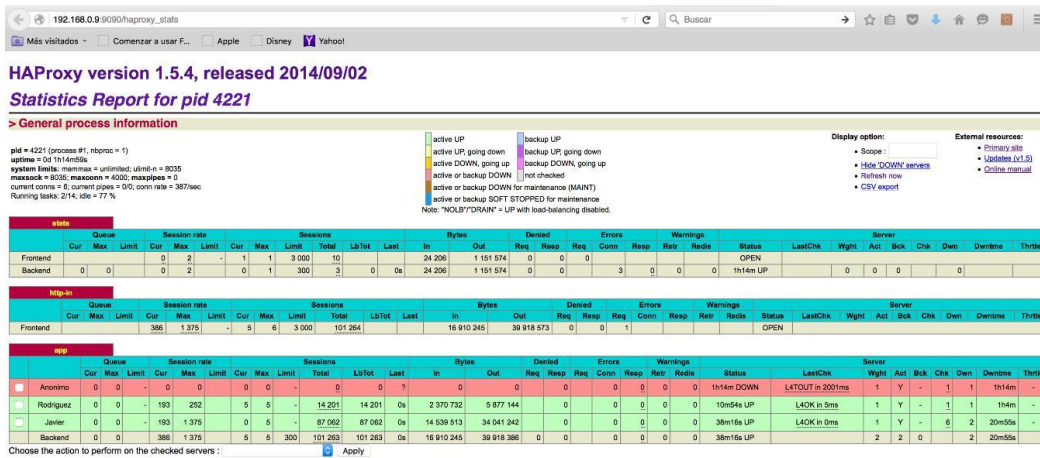


Ilustración 15. Interacción Máquinas Virtuales

En la ilustración 16 se puede ver el cuadro de estadísticas de HAProxy, donde se refleja que los servidores apache alojados en las máquinas Javier Apache y Rodríguez Apache están atendiendo las peticiones de HAProxy.



84

Ilustración 16. Reporte de estadísticas acceso máquina virtual Javier Apache y Rodríguez Apache

Ejecución de comando tcpdump -n port 8080, que permite mostrar las peticiones TCP enviadas por la IP 192.168.0.9 y son atendidas por el puerto 8080 de la máquinas 192.168.0.22 y la 192.168.0.23

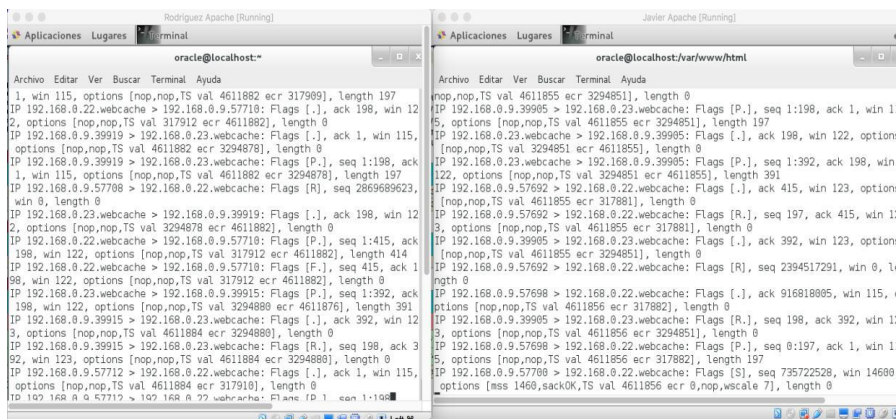


Ilustración 17. Análisis de tráfico Balanceador - Maquina Javier Apache y Rodríguez Apache

A continuación se muestran las páginas de respuesta haciendo el llamado a la aplicación alojada en los servidores apache Javier Apache y Rodríguez Apache, además la ejecución de un comando que realiza 5 peticiones concurrentes durante 240 segundos a la Ip 192.168.0.9 por el puerto 80 que corresponde a la ubicación y configuración de HAProxy: "Siege".



85

Ilustración 18. Peticiones al balanceador atendidas por los servidores Javier Apache y Rodríguez Apache.


```

[MacBook-Pro-de-Alejandro-2:Downloads hrcalejo$ siege -t 240s -c 5 -b -q http://192.168.0.9:80/index.html
Lifting the server siege...      done.

Transactions:          69208 hits
Availability:          100.00 %
Elapsed time:          239.40 secs
Data transferred:     8.68 MB
Response time:         0.02 secs
Transaction rate:     289.09 trans/sec
Throughput:            0.04 MB/sec
Concurrency:           4.99
Successful transactions: 69208
Failed transactions:   0
Longest transaction:  2.28
Shortest transaction:  0.00
    
```

Ilustración 19. Comprobación Rendimiento.

c) La prueba siguiente se realiza con las tres máquinas virtuales encendidas.

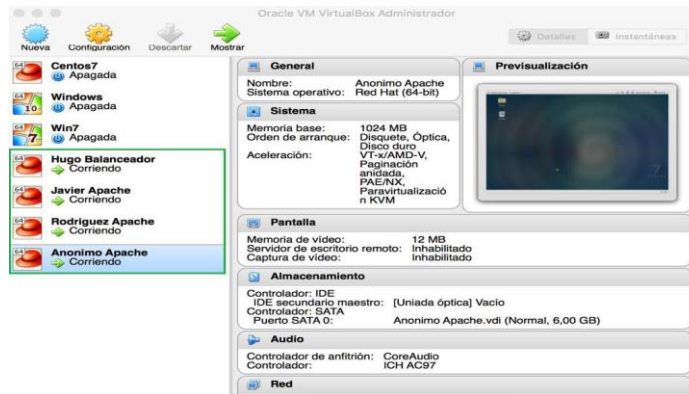


Ilustración 20. Interacción Máquinas Virtuales

La ilustración 21 muestra las estadísticas de HAProxy donde se refleja que los servidores apache alojados en las máquinas Javier Apache, anónimo Apache y Rodríguez Apache están atendiendo las peticiones de HAProxy.

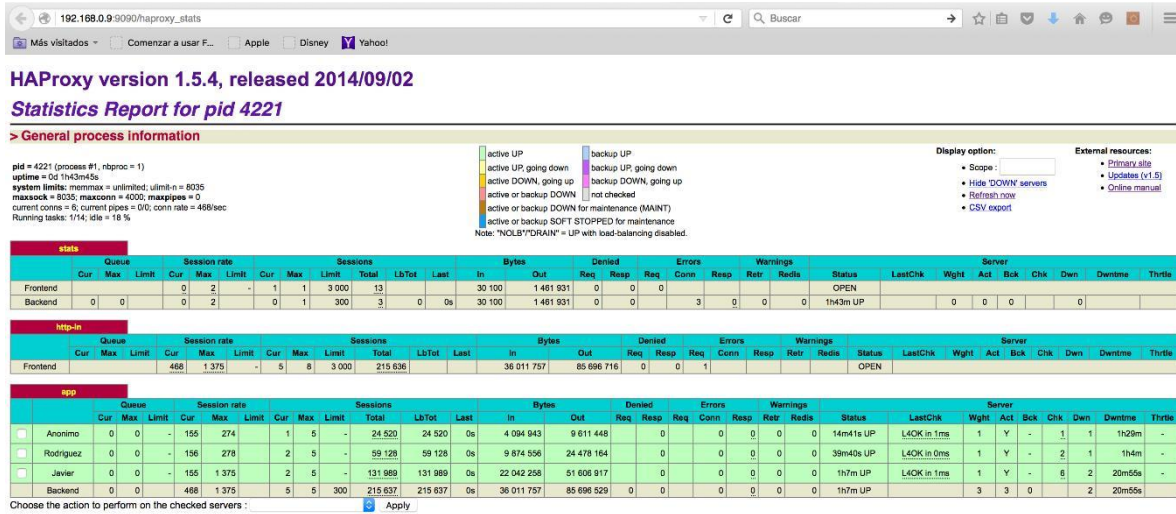


Ilustración 21. Reporte de estadísticas acceso a todas las máquinas virtuales

87

Ejecución de comando tcpdump -n port 8080, que permite mostrar las peticiones TCP enviadas por la IP 192.168.0.9 y son atendidas por el puerto 8080 de la máquinas 192.168.0.22, la 192.168.0.23 y 192.168.0.21

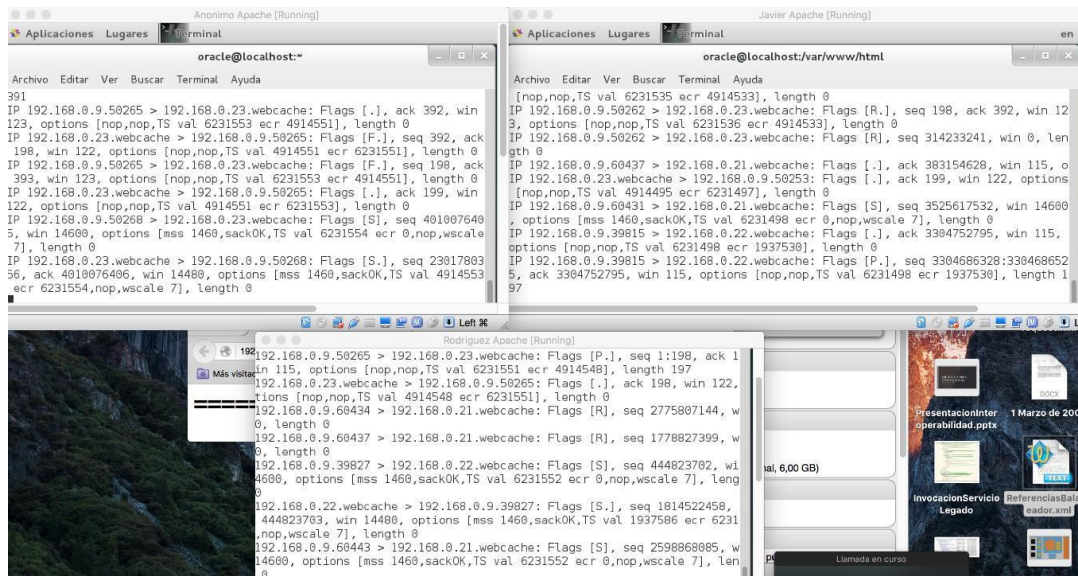
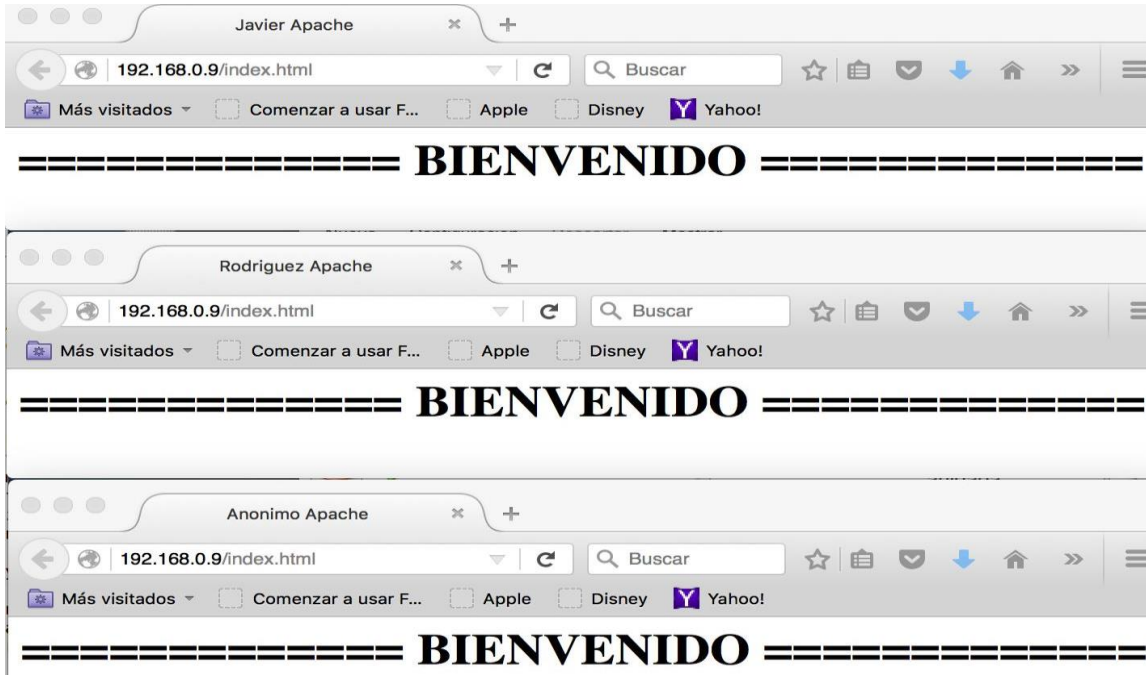


Ilustración 22. Análisis de tráfico Balanceador - todas las máquinas virtuales configuradas

A continuación, se muestran las páginas de respuesta haciendo el llamado a la aplicación alojada en los servidores apache Javier Apache, Rodríguez Apache y Anónimo Apache, además la ejecución de un comando que realiza 5 peticiones concurrentes durante 240 segundos a la Ip 192.168.0.9 por el puerto 80 que corresponde a la ubicación y configuración de HAProxy: “Siege”



88

Ilustración 23. Peticiones al balanceador atendidas por los servidores Javier Apache, Rodríguez Apache y Anónimo Apache

```
MacBook-Pro-de-Alejandro-2:Downloads harcalejo$ siege -t 240s -c 5 -b -q http://192.168.0.9:80/index.html
Lifting the server siege...      done.

Transactions:          94984 hits
Availability:          100.00 %
Elapsed time:          239.81 secs
Data transferred:     11.90 MB
Response time:         0.01 secs
Transaction rate:     396.08 trans/sec
Throughput:            0.05 MB/sec
Concurrency:           4.99
Successful transactions: 94984
Failed transactions:   0
Longest transaction:   3.02
Shortest transaction:  0.00
```

Ilustración 24. Comprobación Rendimiento.

Conclusiones

La utilización de un balanceador de carga y la realización de este trabajo refuerzan la fundamentación teórica de la misma, ya que se ve su aplicabilidad y aporta a la experiencia para futuras implementaciones en los diseños de las arquitecturas cuando se requiera alta disponibilidad y mejorar el rendimiento de una aplicación con la distribución de cargas de trabajo. Con la exploración de conceptos de nuevas herramientas de software libre, ayuda a los arquitectos en la implementación de diseños de arquitecturas con menores costos y logran cumplir los requerimientos en este caso de disponibilidad y rendimiento.

Con la ejecución de HAProxy y los comandos para pruebas de carga Siegue, se logra ver en diferentes escenarios como las disponibilidad de la página, la cual se mantenía en un 100 %, a pesar de realizar baja de servidores. De manera que la decisión de usar HAProxy fue una decisión acertada, ya que permite con sus propias configuraciones ver como mantenía actualizado el estado de sus servidores backend y los estadísticos de peticiones que a ellos se enviaban.

La instalación y configuración de HAProxy permite ver la simplicidad en la implantación de estas herramientas con las que se puede solucionar problemas complejos de disponibilidad y rendimiento en las aplicaciones, aunque para este caso no fue necesario, pues de hecho HAProxy de forma sencilla ofrece la posibilidad de configurar escenarios donde se pueda trabajar con soluciones que requieren mantener sesiones durante toda la interacción con esta. Por último, este tipo de laboratorios son de grandes interés y producen gran satisfacción, ya que permiten conocer diferentes herramientas o soluciones que existen en el entorno y atacar problemas que se manifiestan en diferentes proyectos y organizaciones.

89

Referencias bibliográficas

- [1] W. Tarreu, HAProxy Configuration Manual [En línea]. 2015, 13 de noviembre. Disponible en: <http://www.haproxy.org/download/1.4/doc/configuration.txt>
- [2] The Apache, Apache Tomcat [En línea]. 2015, 13 de octubre. Disponible en: <http://tomcat.apache.org/>
- [3] Liquiweb, How to Install Apache on CentOS 7 [En línea]. 2015, 13 de noviembre. Disponible en <http://www.liquidweb.com/kb/how-to-install-apache-on-centos-7/>
- [4] Serverlab, Deploying an HAProxy Load Balancer on CentOS 6 [En línea]. 2015, 14 de noviembre. Disponible en: <http://www.serverlab.ca/tutorials/linux/network-services/deploying-an-haproxy-load-balancer-on-centos-6/>
- [5] Networking Howtos, Viewing HAProxy Statistics, [En línea]. 2015, 14 de noviembre. Disponible en: <http://www.networkinghowtos.com/howto/viewing-haproxy-statistics/>