

IMPLEMENTACIÓN DE BALANCEO DE CARGA DE SERVIDORES WEB CON HAPROXY Y PRUEBAS DE CARGA CON ARTILLERY.

Juan Martin Vasquez Caicedo, 2185274
Jhon Sebastian Cortes Vasquez, 2205199
Felipe Chavez Romero, 2195109
Alejandro Giraldo 2161631,

Universidad Autónoma de Occidente
Santiago de Cali, Valle del Cauca
Facultad de Ingeniería
Servicios Telematicos
2023

IMPLEMENTACION DE BALANCEO DE CARGA DE SERVIDORES WEB CON HAPROXY Y PRUEBAS DE CARGA CON ARTILLERY.

Juan Martin Vasquez Caicedo, 2185274
Jhon Sebastian Cortes Vasquez, 2205199
Felipe Chavez Romero, 2195109
Alejandro Giraldo 2161631,

Informe Final de Proyecto

Docente
Oscar Hernan Mondragon

Universidad Autónoma de Occidente
Santiago de Cali, Valle del Cauca
2023

Resumen

El balanceo de carga de servidores web es una técnica utilizada para distribuir el tráfico de red entre varios servidores, con el objetivo de mejorar la escalabilidad, disponibilidad y rendimiento del sitio web. En este trabajo, se presenta la implementación de balanceo de carga utilizando HAProxy, un software de código abierto que se utiliza ampliamente en entornos de producción.

Para evaluar la efectividad del balanceo de carga, se realizaron pruebas de carga utilizando Artillery, una herramienta de pruebas de carga de código abierto. Se configuró un conjunto de pruebas de carga para simular múltiples usuarios accediendo al sitio web al mismo tiempo

Los resultados de las pruebas de carga demostraron que el balanceo de carga con HAProxy mejoró significativamente el rendimiento del sitio web. Se redujo el tiempo de respuesta y la tasa de errores, y se mejoró la utilización del servidor.

Índice de términos - Haproxy, balanceo, escalabilidad, disponibilidad.

INTRODUCCIÓN

Los sistemas de conectividad y comunicación tienen un papel fundamental, y su disponibilidad y rendimiento son cruciales para garantizar un funcionamiento fluido de los servicios y aplicaciones de TI. En este sentido, el monitoreo de infraestructura se convierte en una actividad clave para mantener la disponibilidad y el rendimiento óptimo de la red. En este informe se describe la implementación de un proyecto utilizando la herramienta HAProxy para el balanceo de carga de servidores web y pruebas de carga con Artillery.

El proyecto implicó la configuración de un entorno de red real utilizando Vagrant y VirtualBox, y la utilización de HAProxy para el balanceo de carga de servidores web. El objetivo fue distribuir el tráfico web entrante entre varios servidores, asegurando una alta disponibilidad y un rendimiento mejorado. Además, se realizaron pruebas de carga utilizando Artillery para simular un alto volumen de usuarios concurrentes accediendo a los servidores web.

PROBLEMÁTICA

En el contexto de servidores con gran flujo de tráfico, es común encontrar fallos y problemas de rendimiento que afectan Algunos de estos desafíos incluyen:

A. Alto tráfico de usuarios

En entornos de servidores web con un alto tráfico de usuarios, puede surgir el problema de la sobrecarga de servidores, lo que afecta el rendimiento y la capacidad de respuesta del sitio web. Si no se distribuye equitativamente la carga de trabajo, un único servidor puede estar sobrecargado, lo que resulta en tiempos de respuesta lentos y una mala experiencia para los usuarios. Esto puede ocasionar pérdida de clientes, disminución de la satisfacción del usuario y pérdida de oportunidades de negocio.

B. Disponibilidad de servicios

La disponibilidad de los servicios web es crucial para garantizar una experiencia continua para los usuarios. Cuando un servidor experimenta un fallo o se vuelve inaccesible, los usuarios pueden enfrentar interrupciones en el acceso a los servicios. Esto puede ocasionar pérdidas económicas, afectar la reputación de la empresa y generar una insatisfacción generalizada entre los usuarios. Es fundamental contar con una solución que permita redirigir el tráfico automáticamente hacia servidores de respaldo en caso de fallos, asegurando así una alta disponibilidad de los servicios.

C. Control de acceso y seguridad

La seguridad y el control de acceso a los servicios web son aspectos críticos en la protección de los datos y la privacidad de los usuarios. Sin un mecanismo adecuado, los servidores web pueden ser vulnerables a ataques maliciosos y accesos no autorizados. La implementación de un proxy inverso, como HAProxy, permite establecer reglas personalizadas para el enrutamiento y control de acceso, lo que garantiza que solo los usuarios autorizados puedan acceder a los servicios. Además, la capacidad de implementar SSL/TLS en HAProxy permite cifrar las comunicaciones, protegiendo la integridad y confidencialidad de los datos transmitidos.

OBJETIVOS

El objetivo principal de este proyecto es implementar un balanceador de carga con la herramienta Haproxy utilizando dos o más servidores apache y hacer pruebas de carga/estrés con artillery y sus herramientas Vagrant y VirtualBox.

El objetivo final es tener un balanceador de carga efectivo para poder sostener durante 10 minutos 450 personas sin picos de botella y con un más de 95% peticiones 200 resueltas de manera exitosa.

REQUERIMIENTOS

La implementación del proyecto se divide en tres requerimientos principales: montar una red de equipos virtuales mediante la herramienta vagrant, utilizar un software para el balanceo de caras , y mostrar si funcionan los algoritmos utilizados como puede ser Haproxy. Además, se realizan pruebas para verificar el monitoreo de equipos a través del sumary report de Artillery y la verificación de cambios en tiempo real que nos proporciona Haproxy como es Stadistic-Haproxy.

PRUEBAS MÍNIMAS ESPERADAS

Para este proyecto se han establecido al menos tres pruebas mínimas esperadas que deben realizarse para validar su funcionamiento.

- 1) Pruebas de carga variando los parámetros como los diferentes algoritmos (roud robin/source).
- 2) Probar con diferente cantidad de servidores web en el balanceador de carga.

3) Explorar las diferentes estadísticas brindadas en artillery como peticiones resueltas, tasa de llegada de peticiones, porcentaje de fallas en la atención de peticiones.

ALTERNATIVAS DE SOLUCIÓN

Artillery es una herramienta de prueba de carga y estrés de código abierto diseñado para evaluar el rendimiento y la escalabilidad de aplicaciones y servicios web. Permite simular el tráfico de usuarios reales y medir cómo responde el sistema bajo diferentes cargas. Algunas características clave de Artillery incluyen la capacidad de definir escenarios de prueba personalizados, generar altas cargas de trabajo, monitorear métricas en tiempo real y generar informes detallados de rendimiento.

Apache JMeter: Es una herramienta de prueba de carga y rendimiento ampliamente utilizada. Proporciona una interfaz gráfica de usuario para diseñar escenarios de prueba y ofrece una amplia gama de funciones y protocolos compatibles. JMeter es altamente extensible y permite simular una variedad de casos de uso y configuraciones complejas.

Gatling: Es otra herramienta de prueba de carga de código abierto escrita en Scala. Se enfoca en la simulación de usuarios concurrentes y ofrece una sintaxis de scripting basada en DSL para definir los escenarios de prueba. Gatling se destaca por su capacidad de escalar y gestionar altas cargas de trabajo, y proporciona métricas detalladas y visuales en tiempo real.

Locust: Es una herramienta de prueba de carga escrita en Python. Permite definir escenarios de prueba utilizando código Python y ofrece una interfaz web para controlar y supervisar las pruebas de carga.

ARTILLERY

Artillery es una poderosa herramienta de prueba de carga y rendimiento diseñada para evaluar el comportamiento y la capacidad de respuesta de aplicaciones y servicios web. Permite simular el tráfico de usuarios, generar cargas de trabajo significativas y medir cómo se desempeña una aplicación bajo diferentes condiciones de carga.

Las principales ventajas de Artillery son:

Flexibilidad en la definición de escenarios: Artillery ofrece una amplia flexibilidad para definir escenarios de prueba. Puedes especificar el número de usuarios virtuales, el patrón de tráfico, las solicitudes HTTP, las pausas y otros parámetros según tus necesidades. Esto te permite simular condiciones reales de uso y probar diferentes aspectos de tu aplicación.

Métricas de rendimiento detalladas: Durante las pruebas, Artillery recopila una variedad de métricas de rendimiento, como tiempos de respuesta, tasas de error, rendimiento por segundo, entre otros. Estas métricas te brindan una visión profunda del rendimiento de tu aplicación y te ayudan a identificar cuellos de botella, áreas de mejora y posibles problemas de escalabilidad.

Escalabilidad y rendimiento: Artillery está diseñado para gestionar altas cargas de trabajo y escenarios de

prueba concurrentes. Puede simular miles de usuarios simultáneos, generando una carga de tráfico significativa en tu aplicación. Esto te permite evaluar la capacidad de tu infraestructura y garantizar que tu aplicación pueda manejar el tráfico esperado.

Integración con sistemas de monitoreo: Artillery se puede integrar fácilmente con sistemas de monitoreo y generación de informes, como StatsD y Graphite. Esto te permite obtener una visualización más detallada de los resultados de las pruebas y comparar el rendimiento en diferentes escenarios.

Fácil de usar y configurar: Artillery ofrece una interfaz de línea de comandos intuitiva y una sintaxis simple para definir los escenarios de prueba. Esto facilita la configuración y ejecución de las pruebas, incluso para aquellos sin experiencia previa en pruebas de carga.

HERRAMIENTAS A USAR

1) **Vagrant:** Es una herramienta de código abierto para la creación y gestión de entornos de desarrollo virtualizados [2]. Permitirá crear una configuración de máquinas virtuales para simular una red de equipos.

2) **VirtualBox:** Es un software de virtualización de sistemas operativos de propósito general. Será utilizado para alojar las máquinas virtuales creadas por Vagrant.

3) **Artillery:** Es un software de prueba de carga y rendimiento diseñado para evaluar el comportamiento y la capacidad de respuesta de aplicaciones y servicios web.

4) **Apache:** Es una herramienta poderosa y altamente configurable que permite alojar y entregar contenido web a través del protocolo HTTP.

5) **Haproxy:** es una herramienta de balanceo de cargas y también sirve para llevar las estadísticas del servidor complementándose con Artillery y sus informes.

6) **Dns – Bind:** Se trata de un software de código abierto que permite la resolución de nombres de dominio en direcciones IP y viceversa

Cada herramienta tendrá una tarea importante dentro del proyecto.

ESTRUCTURA A IMPLEMENTAR

Una vez que las máquinas virtuales estén en funcionamiento, se implementará Haproxy como herramienta de balanceo de cargas para distribuir las peticiones en los servidores anteriormente creados y corriendo.

En la implementación, se definirán los diferentes algoritmos en el servidor de balanceo de cargas y la prioridad de carga puntos de monitoreo en las máquinas virtuales y se conformarán pruebas de carga/estrés para llenar de peticiones a nuestro balanceador y medir su nivel de rendimiento dependiendo del volumen de usuarios y encontrar el punto de se llevarán a cabo pruebas de carga constante en el sistema para observar cómo se congestiona y cómo se descongestiona para asegurarse de que el sistema de monitoreo funciona correctamente y se medirá el rendimiento de la red y la respuesta del sistema ante diferentes

cargas. Esto permitirá evaluar la capacidad del sistema y detectar posibles fallos o retardos.

DESARROLLO

Utilizando las herramientas de virtualización de VirtualBox y Vagrant, se realiza su configuración y aprovisionamiento. Incluyendo la instalación y configuración del software adicional como apache, vim y haproxy.

A. Configuración de Haproxy

En el servidor que se instalara Haproxy se debe instalar y su archivo estará en /etc/haproxy/haproxy.cfg, en este archivo se configura los default, los front-end, front-end stats y el backend. Haproxy su función es balancear cargas por lo tanto los otros servidores que serán balanceados deben estar en correcto funcionamiento.

B. Configuración de Artillery

Artillery se descarga mediante su página web oficial y se debe instalar Node.js y también habilitar los scripts en Windows cambiando los criterios en el mode, artillery utiliza extensión YAML, para utilizar Artillery con Haproxy solo debemos colocar la dirección ip de nuestro servidor Haproxy y ejecutamos las diferentes pruebas que necesitemos en el código, las cantidades de pruebas a realizar dependen del ejecutor y su imaginación, los criterios dependen de que tan escalable y disponible es el balanceador de cargas que necesite en ese momento.

C. Evaluación del rendimiento

Se analizan los resultados para evaluar el rendimiento del sistema y detectar posibles fallas o logros del sistema.

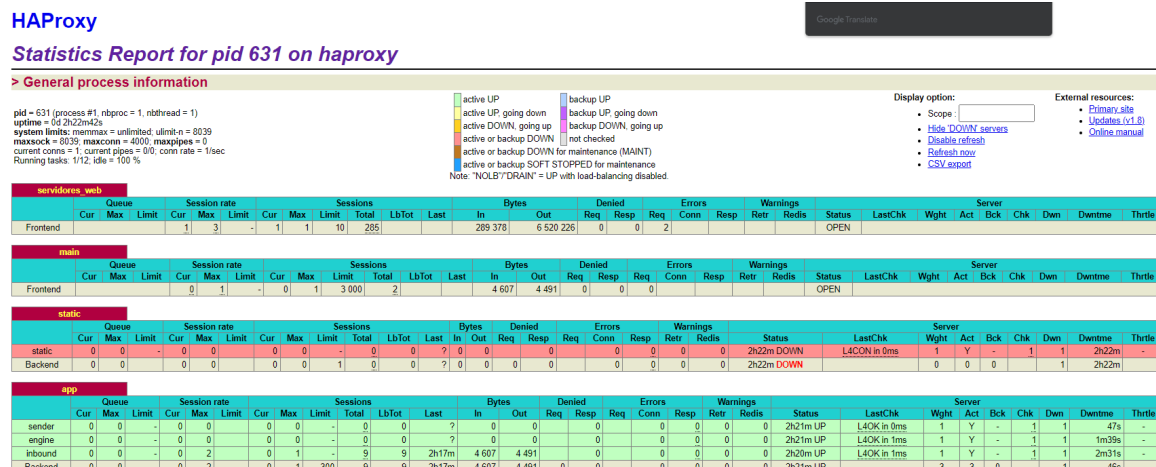


Fig. 1. Haproxy Statistics Report. Sirve para monitorear el flujo de peticiones y usuarios en los servidores.

RESULTADOS DE PRUEBA

Una vez que Artillery termine la prueba se desplegará un informe con diferentes peticiones, usuarios creados, fallidos, peticiones 600 aceptadas y muchas otras más.

```

-----
Summary report @ 14:59:18(-0500)
-----

http.codes.200: ..... 600
http.codes.404: ..... 600
http.downloaded_bytes: ..... 1785600
http.request_rate: ..... 20/sec
http.requests: ..... 1200
http.response_time:
  min: ..... 3
  max: ..... 33
  mean: ..... 6.3
  median: ..... 6
  p95: ..... 8.9
  p99: ..... 10.1
http.responses: ..... 1200
vusers.completed: ..... 600
vusers.created: ..... 600
vusers.created_by_name.0: ..... 600
vusers.failed: ..... 0
vusers.session_length:
  min: ..... 2011.9
  max: ..... 2066.4
  mean: ..... 2017.5
  median: ..... 2018.7
  p95: ..... 2018.7
  p99: ..... 2018.7

C:\compilaciones\Servicios_Telematicos\artillery\prueba de carga basica>

```

Fig. 2. Summary report. Sirve para registrar las tareas ejecutadas y la tasa de éxito/fallo, también su velocidad y mucho más.

CONCLUSIONES

Balanceo de carga efectivo: HAProxy ofrece una solución confiable y eficiente para distribuir la carga de trabajo entre varios servidores, lo que mejora el rendimiento y la disponibilidad de los servicios web. Permite escalabilidad y evita la sobrecarga de un único servidor.

Alta disponibilidad: HAProxy garantiza la continuidad del servicio al redirigir automáticamente el tráfico hacia servidores de respaldo en caso de fallos. Esto asegura una mayor disponibilidad y minimiza el impacto de los tiempos de inactividad no planificados.

Pruebas de carga eficientes: Artillery es una herramienta potente para realizar pruebas de carga y estrés en aplicaciones y servicios. Permite simular una gran cantidad de usuarios concurrentes y medir el rendimiento del sistema, identificando posibles cuellos de botella y problemas de escalabilidad.

Escalabilidad y rendimiento: Artillery es capaz de manejar escenarios de prueba de carga complejos y generar una carga significativa en los sistemas objetivos. Esto permite evaluar el rendimiento y la

capacidad de respuesta de las aplicaciones y servicios en condiciones reales.

Flexibilidad en la configuración de pruebas: Artillery ofrece opciones flexibles para definir los escenarios de prueba, incluyendo la posibilidad de personalizar las solicitudes HTTP, agregar lógica de negocios y establecer tasas de llegada de usuarios. Esto permite simular casos de uso específicos y adaptar las pruebas a las necesidades del proyecto.

Finalizando, el proyecto de implementación y balanceo de cargas mediante Haproxy y las pruebas mediante Artillery fue algo único debido a su complejidad y los conocimientos adquiridos, mantener un sistema de balanceo es bastante complejo y siempre se necesitan utilizar buenas prácticas para un correcto funcionamiento.

REFERENCIAS

[1]

1. [1] Artillery. (2023). "HTTP Reference." Retrieved [13/05/2023], from Artillery.io: <https://www.artillery.io/docs/guides/guides/http-reference>
2. [2] Vagrant. (2023). "Vagrant Documentation." Retrieved [13/05/2023], from Vagrant Up: <https://www.vagrantup.com/docs/>
3. [3] AprendeIT. (2023). "Enrutamiento y balanceo básico HTTP con HAProxy." Retrieved [12/05/2023], from AprendeIT: <https://aprendeit.com/enrutamiento-y-balanceo-basico-http-con-haproxy/>
4. [4] PTC. (2023). "HAProxy Example." Retrieved [16/05/2023], from PTC Support: <https://support.ptc.com/help/thingworx/platform/r9/es/index.html#page/ThingWorx/Help/ThingWorxHighAvailability/HAProxyExample.html>
5. [5] D. Kumar and M. Kumar. (2020). "An Empirical Study on Performance Analysis of HAProxy Load Balancer." In IEEE International Conference on Power, Energy and Environment (PEE).
6. [6] V. Kumar, S. Sharma, and S. Singh. (2019). "Performance Evaluation of HAProxy Load Balancer in Cloud Environment." In IEEE International Conference on Communication and Signal Processing (ICCSP).