

## **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 Telemáticos  
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

La técnica de balanceo de carga de servidores web es ampliamente utilizada para distribuir el tráfico de red entre diversos servidores con el objetivo principal de mejorar la escalabilidad, disponibilidad y rendimiento de una aplicación web. Este trabajo presenta la implementación del balanceo de carga mediante el uso de HAProxy, un software de código abierto ampliamente adoptado en entornos industriales y empresariales productivos.

Para evaluar la eficacia del balanceo de carga, se llevaron a cabo diversas pruebas de carga utilizando Artillery, una herramienta de pruebas de carga de código abierto. Se configuraron escenarios de prueba que incluían situaciones de carga y estrés para simular múltiples usuarios virtuales que accedían al sitio web y ejecutaban diversas funciones de manera simultánea.

Los resultados de las pruebas de carga revelaron que la implementación del balanceo de carga con HAProxy tuvo un impacto significativo en el rendimiento del sitio web. Se utilizó una variedad de algoritmos, como Round Robin y Source, en las pruebas, lo que resultó en una reducción del tiempo de respuesta, una disminución de la tasa de errores y una mejora en la utilización del servidor.

## INTRODUCCIÓN

Los sistemas que permiten la conectividad y la comunicación desempeñan un papel esencial, y su capacidad de estar disponibles y funcionar de manera eficiente son cruciales para asegurar un funcionamiento ininterrumpido de los servicios y aplicaciones de tecnología de la información. En este contexto, la supervisión de la infraestructura se convierte en una actividad esencial para mantener la disponibilidad y el rendimiento óptimo de la red. Este informe describe la ejecución de un proyecto que involucra el uso de la herramienta HAProxy para distribuir la carga de trabajo entre servidores web y realizar pruebas de carga utilizando Artillery.

El proyecto consistió en configurar un entorno de red real mediante el uso de Vagrant y VirtualBox, y emplear HAProxy para distribuir la carga de tráfico

web entre varios servidores, garantizando así una alta disponibilidad y un mejor rendimiento. Además, se llevaron a cabo pruebas de carga mediante Artillery con el fin de simular un elevado número de usuarios concurrentes que acceden a los servidores web.

## PROBLEMÁTICA

En entornos donde los servidores manejan un alto volumen de tráfico, es habitual detectar fallos y dificultades en el rendimiento que pueden afectar el funcionamiento de los sistemas. Algunos de los obstáculos comunes abarcan:

### ALTO TRÁFICO DE USUARIOS

Cuando se opera un servidor web en un entorno con una gran afluencia de usuarios, es común que surja el desafío de la sobrecarga del servidor. Este problema impacta negativamente en el rendimiento y la capacidad de respuesta del sitio web. Si la distribución de la carga de trabajo no es uniforme, un solo servidor puede verse sobrecargado, lo que conlleva a tiempos de respuesta más lentos y una experiencia insatisfactoria para los usuarios. Estas situaciones pueden tener consecuencias como la pérdida de clientes, la disminución de la satisfacción de los usuarios y la oportunidad de negocio desaprovechada.

### 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.

## CONTROL DE ACCESO Y SEGURIDAD

La protección de datos y la preservación de la privacidad de los usuarios son aspectos críticos en la seguridad y control de acceso a los servicios web. Sin un mecanismo adecuado, los servidores web pueden quedar expuestos a ataques maliciosos y accesos no autorizados. La implementación de un proxy inverso, como HAProxy, permite establecer reglas personalizadas para dirigir y gestionar el acceso, asegurando que solo los usuarios con autorización puedan ingresar a los servicios. Además, la capacidad de habilitar SSL/TLS en HAProxy permite cifrar las comunicaciones, lo que contribuye a salvaguardar la confidencialidad e integridad de los datos transmitidos.

## PROPÓSITO

El propósito principal de este proyecto radica en la instalación de un balanceador de carga a través de la aplicación **HAProxy**, empleando dos servidores Apache o más, y llevar a cabo evaluaciones de carga y estrés mediante Artillery, respaldado por las herramientas **Vagrant** y **VirtualBox**.

El resultado deseado es contar con un balanceador de carga eficaz que pueda mantener a 450 personas sin enfrentar problemas significativos durante un período de 10 minutos, con una tasa de éxito superior al 95% al procesar solicitudes **HTTP** con el código de respuesta 200 de manera satisfactoria.

## REQUERIMIENTOS

La ejecución del proyecto se desglosa en varios requisitos esenciales: establecer una red de máquinas virtuales utilizando la herramienta Vagrant, emplear un software para llevar a cabo el balanceo de carga y evaluar la eficacia de los algoritmos utilizados, como por ejemplo HAProxy. Además, se efectúan pruebas para confirmar el seguimiento de los equipos a través del informe resumido de Artillery y la observación en tiempo real de los cambios proporcionados por HAProxy.

## PRUEBAS MÍNIMAS ESPERADAS

Para este proyecto, se han definido al menos tres pruebas esenciales que deben llevarse a cabo para validar su funcionamiento.

Realizar pruebas de carga con la variación de parámetros, incluyendo la utilización de distintos algoritmos como Round Robin, Least Connections y Source.

Evaluar el rendimiento con una variedad de configuraciones de servidores web en el balanceador de carga.

Explorar diversas métricas y estadísticas proporcionadas por Artillery, como el número de peticiones resueltas, la tasa de llegada de peticiones y el porcentaje de fallos en la atención de las solicitudes.

## ALTERNATIVAS DE SOLUCIÓN

### ELASTIC LOAD BALANCING (AWS)

Elastic Load Balancing es una solución de equilibrio de carga en la nube que mejora la escalabilidad, disponibilidad y seguridad de las aplicaciones en Amazon Web Services (AWS). Es especialmente útil para aquellas aplicaciones que se ejecutan en la nube de AWS y requieren distribuir el tráfico de red entre múltiples instancias para garantizar un rendimiento óptimo y alta disponibilidad.

Este servicio distribuye automáticamente el tráfico entrante entre varios destinos, como instancias EC2, contenedores y direcciones IP en una o varias zonas de disponibilidad. Monitorea el estado de los destinos registrados y enruta el tráfico solo hacia los destinos en buen estado. Elastic Load Balancing escala automáticamente la capacidad del balanceador de carga en respuesta a cambios en el tráfico entrante.

## AZURE LOAD BALANCER

Azure Load Balancer es un servicio de equilibrio de carga proporcionado por Microsoft Azure, la plataforma de nube de Microsoft. Su función principal es distribuir el tráfico de red de manera equitativa entre múltiples máquinas virtuales (VM) o instancias de aplicaciones dentro de una red virtual de Azure.

Este servicio distribuye solicitudes de tráfico entrante, como solicitudes HTTP, HTTPS, TCP o UDP, entre varios destinos, como VM o instancias de aplicaciones, para evitar la sobrecarga de un solo servidor o instancia.

Ventajas destacadas de Azure Load Balancer incluyen alta disponibilidad y redundancia al enrutar automáticamente las solicitudes hacia servidores o instancias de aplicaciones en funcionamiento, así como la capacidad de realizar equilibrio de carga a nivel de red (Capa 4) y a nivel de aplicación (Capa 7).

## GOOGLE CLOUD LOAD BALANCING

Google Cloud Load Balancing es un servicio completamente gestionado proporcionado por Google Cloud Platform (GCP) que permite distribuir el tráfico de red a través de múltiples instancias o servicios para garantizar la alta disponibilidad, escalabilidad y confiabilidad de aplicaciones.

Ofrece varios tipos de balanceadores de carga, como el Balanceador de Carga Global de Google Cloud, diseñado para distribuir el tráfico en múltiples regiones, y el Balanceador de Carga Regional de Google Cloud, que se enfoca en la distribución de tráfico dentro de una sola región, asegurando alta disponibilidad.

Además, Google Cloud Load Balancing ofrece servicios como Google Cloud Traffic director, Proxy SSL de Google Cloud, y otras opciones diseñadas para satisfacer diferentes casos de uso y mejorar la seguridad y el rendimiento de las aplicaciones.

## ARTILLERY

Artillery es una robusta herramienta de evaluación de carga y rendimiento diseñada para analizar el comportamiento y la capacidad de respuesta de aplicaciones y servicios web. Su utilidad radica en la simulación del tráfico de usuarios, generación de cargas significativas y medición del rendimiento de una aplicación en diversas condiciones de carga.

Las ventajas clave de Artillery son las siguientes:

**Flexibilidad en los escenarios:** Artillery proporciona una flexibilidad amplia para establecer escenarios de prueba. Es posible especificar el número de usuarios virtuales, el patrón de tráfico, solicitudes HTTP, pausas y otros parámetros de acuerdo con las necesidades específicas. Esto permite simular situaciones de uso reales y evaluar múltiples aspectos de la aplicación.

**Métricas que visualizan el rendimiento:** En el transcurso de las pruebas, Artillery recolecta diversas métricas de rendimiento, tales como tiempos de respuesta, tasas de error, rendimiento por segundo, entre otras. Estas métricas brindan una visión profunda del desempeño de la aplicación y son útiles para identificar cuellos de botella, áreas de mejora y posibles problemas de escalabilidad.

**Escalabilidad y rendimiento:** Artillery está diseñado para manejar cargas de trabajo elevadas y pruebas simultáneas. Puede simular miles de usuarios concurrentes, lo que genera una carga de tráfico considerable en la aplicación. Esto posibilita evaluar la capacidad de la infraestructura y garantizar que la aplicación sea capaz de manejar el tráfico previsto.

**Integración con sistemas de monitoreo:** Artillery se integra de forma sencilla con sistemas de monitoreo y generación de informes, como StatsD y Graphite. Esto permite obtener visualizaciones más detalladas de los resultados de las pruebas y comparar el rendimiento en diferentes escenarios.

**Facilidad de uso y configuración:** Artillery ofrece una interfaz de línea de comandos intuitiva y una sintaxis sencilla para definir los escenarios de prueba. Esto simplifica la configuración y ejecución de las pruebas, incluso para aquellos sin experiencia previa en pruebas de carga.

## APLICATIVOS REQUERIDOS EN EL DESARROLLO

**Vagrant:** Se trata de una herramienta de código abierto que posibilita la creación y administración de entornos de desarrollo virtualizados.

Su función principal consiste en establecer una configuración de máquinas virtuales con el propósito de simular una red de sistemas.

**VirtualBox:** Un software de virtualización de sistemas operativos de uso general que se empleará para hospedar las máquinas virtuales creadas mediante Vagrant.

**Artillery:** Un software destinado a la evaluación del rendimiento y carga diseñado para analizar el comportamiento y la capacidad de respuesta de aplicaciones y servicios web.

**Apache:** Una herramienta altamente configurable y poderosa utilizada para el alojamiento y entrega de contenido web a través del protocolo HTTP.

**HAProxy:** Una herramienta de equilibrio de cargas que, en este contexto, también se empleará para recopilar estadísticas del servidor, complementando así las capacidades de Artillery y sus informes.

**DNS – Bind:** Un software de código abierto que facilita la resolución de nombres de dominio en direcciones IP y viceversa. Cada una de estas herramientas desempeñará un papel fundamental dentro del proyecto.

**Node.js** es un entorno de tiempo de ejecución de código abierto que permite a los desarrolladores ejecutar JavaScript en el lado

del servidor. A diferencia del JavaScript tradicional, que se ejecuta en el navegador del cliente.

## IMPLEMENTACIÓN Y DESARROLLO

Se planea implementar implica el despliegue de máquinas virtuales, seguido por la configuración de HAProxy como herramienta de equilibrio de carga. Esta tarea tiene como objetivo distribuir las solicitudes entre los servidores previamente creados y en funcionamiento.

En esta implementación, se establecerán algoritmos que son empleados en el servidor de equilibrio de carga y se definirá la prioridad de los puntos de monitoreo en las máquinas virtuales. A continuación, se llevarán a cabo pruebas de carga y estrés para inundar el balanceador de solicitudes, con el fin de evaluar su rendimiento en función del número de usuarios y determinar su capacidad de respuesta. También se realizarán pruebas continuas en el sistema para observar cómo se congestionan y des congestionan, asegurando que el sistema de monitoreo esté funcionando correctamente. Se medirá el rendimiento de la red y la respuesta del sistema en distintos niveles de carga. Esto permitirá evaluar la capacidad del sistema y detectar cualquier posible fallo en el sistema evaluado.

## DESARROLLO

A través de las herramientas de virtualización proporcionadas por VirtualBox y Vagrant, se procede a configurar el entorno en donde se emplearon 4 servidores virtuales con las siguientes IP:

10.85.32.191 – haproxy, 10.85.32.172 – sender, 10.85.32.225 – engine, 10.85.32.46 – inbound

En donde se procedió a instalarse aplicaciones para adecuar los servidores, como vim, HAProxy, httpd, net-tools, bind-utils, desactivar selinux, también el firewall, ajustar el named.conf y archivos de zona para la parte de dns

## AJUSTE Y CONFIGURACIÓN DE HAProxy

En el servidor donde HAProxy se implementará, se requerirá su instalación, y el archivo de configuración se encontrará en la ruta /etc/haproxy/haproxy.cfg. En este archivo se



Para utilizar Artillery en conjunto con HAProxy, simplemente se debe indicar la url en nuestro caso fue la siguiente ya que se le configuro un dns `ww2.haproxy.com` y su dirección ip: `10.85.32.191` es de nuestro servidor HAProxy en el código y ejecutar las pruebas necesarias.

La cantidad de pruebas a realizar estará sujeta a la decisión del usuario y su inventiva, y los criterios empleados estarán vinculados a los requerimientos específicos de escalabilidad y disponibilidad del balanceador de cargas en ese momento. Esto forma parte de la evaluación del rendimiento. Se analizan los resultados para evaluar el rendimiento del sistema y detectar posibles fallas o logros del sistema.

Artillery utiliza archivos de extensión YAML e

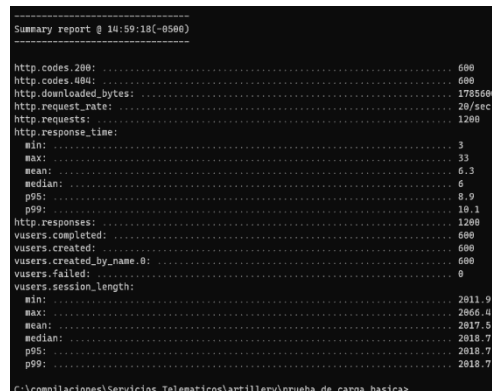


Fig. 2. Summary report. Sirve para registrar las tareas ejecutadas y la tasa de éxito/fallo, también su velocidad y mucho más.

## CONCLUSIONES

Lograr un equilibrio de carga eficiente es una de las ventajas clave de HAProxy. Proporciona una solución confiable y efectiva para distribuir la carga de trabajo entre múltiples servidores, lo que tiene un impacto positivo en el rendimiento y la disponibilidad de los servicios web. Esto se traduce en la capacidad de escalar y evitar la sobrecarga de un servidor individual.

La alta disponibilidad es otro punto fuerte de HAProxy, ya que garantiza la continuidad del

servicio redirigiendo automáticamente el tráfico a servidores de respaldo en caso de fallos. Esto no solo mejora la disponibilidad, sino que también minimiza los efectos de posibles tiempos de inactividad imprevistos.

En cuanto a las pruebas de carga, Artillery se destaca como una herramienta potente y eficiente. Permite simular un gran número de usuarios concurrentes, lo que facilita la medición del rendimiento del sistema y la identificación de posibles cuellos de botella y problemas de escalabilidad.

La escalabilidad y el rendimiento son aspectos clave de Artillery, ya que puede manejar escenarios de prueba de carga complejos y generar cargas significativas en los sistemas objetivos. Esto brinda la capacidad de evaluar el rendimiento y la capacidad de respuesta de las aplicaciones y servicios en condiciones del mundo real. Artillery también se distingue por su flexibilidad en la configuración de pruebas. Ofrece opciones versátiles para definir escenarios de prueba, lo que incluye la personalización de solicitudes HTTP, la incorporación de lógica empresarial y la configuración de tasas de llegada de usuarios. Esto permite simular casos de uso específicos y adaptar las pruebas según las necesidades del proyecto.

En resumen, la implementación del proyecto de balanceo de cargas con HAProxy y las pruebas realizadas mediante Artillery representan un desafío significativo debido a su complejidad. Adquirir los conocimientos necesarios y aplicar buenas prácticas son elementos fundamentales para garantizar un funcionamiento adecuado en un sistema de balanceo de carga.

## REFERENCIAS

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).