

Comparativa de métodos de asignación de servidores CDN a clientes

Bastían Rivas C.
Ing. Civil Electrónica
UTFSM
Valparaíso, Chile
bastian.rivas@usm.cl

Abstract—Las CDN son una parte vital del Internet al acercar el contenido al usuario y hacer más eficiente la entrega del mismo. En este paper se presenta un análisis de las técnicas más usadas para la asignación de servidores de contenido a los usuarios, enfatizando en los tiempos de espera y los anchos de banda disponibles para las tareas de los clientes. Se hace una comparación entre estas técnicas y se identifican sus falencias, como también se hacen experimentos con redes ya existentes en distintas zonas geográficas. Los hallazgos hechos dan pistas a posibles mejoras que se le pueden hacer a los esquemas ya existentes y a la evolución que pueden tener.

Index Terms—CDN, bandwidth, anycast, fastroute, DNS

I. INTRODUCCIÓN

Las redes de distribución de contenido (CDN, por sus siglas en inglés) se han vuelto una parte fundamental de los servicios entregados a través de Internet. Ofrecen numerosas ventajas en cuanto a la calidad, robustez, disponibilidad y evitar tener un único punto de fallo que pueda traer como consecuencia interrupciones totales del servicio, por medio de replicar el contenido que se desea brindar en un servidor cercano al cliente. [1]. Sin embargo, los grandes volúmenes de demanda que experimentan los servicios ofrecidos actualmente han generado que la calidad del servicio no sea óptima. Las exigencias en cuanto a retardos y ancho de banda motivan a que busquen distintas formas de optimizar los procesos que se ejecutan al hacer requerimientos.

Teniendo en cuenta la gran cantidad de clientes que van requiriendo contenido, se necesita ver cómo optimizar la asignación de servidores a clientes de tal forma que se les brinde el mejor servicio posible. Actualmente hay dos grandes esquemas de selección. Por un lado, se usa el apoyo de un servidor Domain Name Server (DNS) propietario que tenga previamente registradas las IPs de los servidores CDN para que escoja al servidor de contenido con menor Round-Trip Time (RTT), tal como lo hacen servicios como el de Akamai y Amazon Web Services (AWS) [2]. Por otro lado, existen las implementaciones que se apoyan con Anycast [3], usándolo por sí solo o incluyendo alguna modificación que cubra alguna necesidad no considerada en la versión más pura. Para este segundo punto se destacan servicios como Cloudflare que usa Anycast puro y el motor de búsqueda Bing de Microsoft, que usa una técnica basada en Anycast que jerarquiza servidores llamada FastRoute [4].

Para enfrentar el problema de la optimización de asignación de servidores a clientes se propone comparar estas tres técnicas más populares en cuanto a los tiempos de retardo que se perciben y el ancho de banda con el que se puede trabajar, para luego identificar posibles puntos de mejora.

II. TRABAJOS RELACIONADOS

A lo largo de los últimos años, se han hecho varias propuestas para mejorar el desempeño de las redes de contenido. En [2] se presenta la evolución que han presentado los servicios brindados por CDNs, particularmente introduciendo lo que se ha hecho en métodos de asignación basados en DNS. También menciona los servicios que lo usan como Akamai, Limelight y Amazon, las desventajas que pueden tener como la baja en el desempeño al usar un DNS remoto y los posibles arreglos que pueden tener.

Por otra parte, en [3] se presenta Anycast como una de las técnicas de ruteo con bastante adopción para servicios de DNS y CDN, así como su forma de operar, su dependencia en los algoritmos usados en BGPs y propuestas de métricas que se puedan aplicar para cuantificar el desempeño de esta técnica.

Adicionalmente a lo anterior, en [4] se presenta FastRoute como una alternativa basada en Anycast. A diferencia de la variante pura, en esta se toma en cuenta la carga que perciben los servidores y se les da jerarquía. Las capas externas de servidores son las primeras en atender los requerimientos de clientes, siendo capaces de redirigir la carga hacia servidores en capas más internas a medida que se van saturando.

Finalmente, en [5] se presenta un sistema llamado *Polygon*, que cambia varios aspectos en la forma de operar respecto a los métodos anteriormente mencionados. Usando máquinas adicionales, hace que el servicio sea consciente no solo de la carga de los servidores, sino que agrega el tipo de recurso que el cliente va solicitando. Plantea que los sistemas anteriormente mencionados ignoran información importante para definir qué servidor asociar. Según la necesidad del cliente, ya sea en la descarga de archivos grandes o en la transmisión de información con baja latencia, se escoge el servidor que disponga de más CPU, retardo o ancho de banda según el caso, con lo que se puede optimizar aún más el trabajo hecho con CDNs.

III. PROBLEMA IDENTIFICADO

Considerando la alta demanda que enfrentan los servicios ofrecidos en Internet, se identifica la necesidad de optimizar la elección del mejor servidor CDN para cada cliente. Esto implica ofrecer un servicio que logre baja latencia y buen ancho de banda, con lo que se comparan los servicios con los esquemas más usados comúnmente y se concluye respecto a las pruebas hechas.

IV. ESTRATEGIA Y ARQUITECTURA

El trabajo relacionado a este problema puede hacerse de 2 formas. Por una parte, se puede simular usando herramientas como Mininet o GNS3, armar la red CDN con sus servidores y DNS y hacer las consultas pertinentes. Por otra parte, se pueden hacer varias consultas a los servicios ya existentes en distintas zonas geográficas y se pueden analizar los resultados. En ambos casos los resultados obtenidos se representan gráficamente usando retardo y ancho de banda como métricas de interés.

En un principio se consideró simular, pero dados los altos requerimientos computacionales se opta por hacer pruebas sobre redes ya existentes. Para este trabajo en particular, se decide usar la segunda alternativa implementando máquinas virtuales dentro de Microsoft Azure en África, Asia, Australia, Europa y América. Las máquinas ejecutan un código en Python con el que se hacen 200 requerimientos hacia los servicios de AWS (DNS), Cloudflare (Anycast) y Bing (FastRoute) y registran los tiempos de llegada de respuestas.

A. Preparación de las máquinas virtuales

Las pruebas se hacen usando máquinas virtuales Ubuntu 22.04.5 LTS (GNU/Linux 6.5.0-1025-azure x86_64), con 1 vCPU y 1 GiB de memoria RAM. Para configurarlas e ir cargando los scripts se les habilita el puerto SSH (22) y se usa como terminal la aplicación MobaXterm desde un computador con Windows 11. Cada una de estas máquinas se encuentra corriendo en distintas zonas geográficas para tener en cuenta la ejecución de los métodos de asignación en distintos entornos.

B. Dependencias

Para asegurar el correcto funcionamiento de los scripts de prueba, se sugiere instalar Python 3 con el siguiente comando:

```
sudo apt install python-is-python3
```

Se recomienda verificar que exista la librería `requests` para poder hacer los requerimientos. El código utilizado se encuentra disponible en <https://github.com/BRivsC/IPD438/>.

C. Ejecución del experimento

Una vez instalado lo anterior, se carga el programa `test_CDNs.py` y se ejecuta con el comando:

```
python test_CDNs.py
```

En específico, este archivo usa las funciones provistas por la librería `requests` para hacer requerimientos a las URLs correspondientes a los servicios, registrando los tiempos que

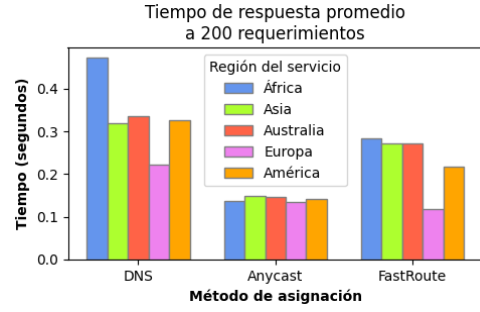


Fig. 1. Resultados del tiempo de respuesta promedio luego de hacer 200 requerimientos

demoró en recibir una respuesta, contabilizando el número de errores y estimando el ancho de banda en base a la ecuación 1.

$$BW = \frac{Largo\ Respuesta}{T_{final} - T_{inicial}} \quad (1)$$

Los resultados se van generando y almacenando en un archivo de registro `cdn_performance.log`, que puede ser descargado a través de la misma aplicación de terminal para su posterior análisis. Este archivo va registrando los tiempos de todos los requerimientos, la cantidad de errores y la estimación del ancho de banda en cada caso, además de ir calculando los valores promedio para cada servicio. Finalmente, se descargan los registros correspondientes a cada máquina virtual y se procesan para poder visualizarlos.

V. RESULTADOS

Los experimentos hechos con los 200 requerimientos determinaron que, en promedio, el método de asignación con mayor tiempo de retardo fue el hecho con servicio DNS, tal como se puede observar en la figura 1. Adicionalmente puede notarse que los tiempos de retardo en el caso africano fueron incluso mayores, lo que puede ser explicado por la menor densidad de equipos de red disponibles y la asociación con servidores más lejanos como consecuencia. Por otra parte, Anycast obtuvo menores tiempo de respuesta promedio en todas las regiones. Esto puede deberse a que se escogieron servidores con menor distancia en términos de saltos por routers.

En cuanto al desempeño obtenido con FastRoute se puede notar que obtuvo ciertas mejoras en comparación a DNS, sobre todo en África. Esta mejora puede asociarse a la distribución de carga hecha en los servidores africanos, haciendo que los más ociosos puedan atender los requerimientos que los otros no pueden tomar.

Analizando la dispersión de los tiempos que se demoran en responder los servidores de contenido respectivos obtenida de la figura 2, se obtiene que en la gran mayoría de los casos los servicios asignando con DNS son los que experimentan tiempos más variables. También se puede obtener que FastRoute arrojó en su mayoría los resultados más consistentes. Esto puede atribuirse a la carga que experimentan los servidores y a la distancia real que existe entre cliente y servidos. Como FastRoute sí toma en cuenta la carga de cada uno de los

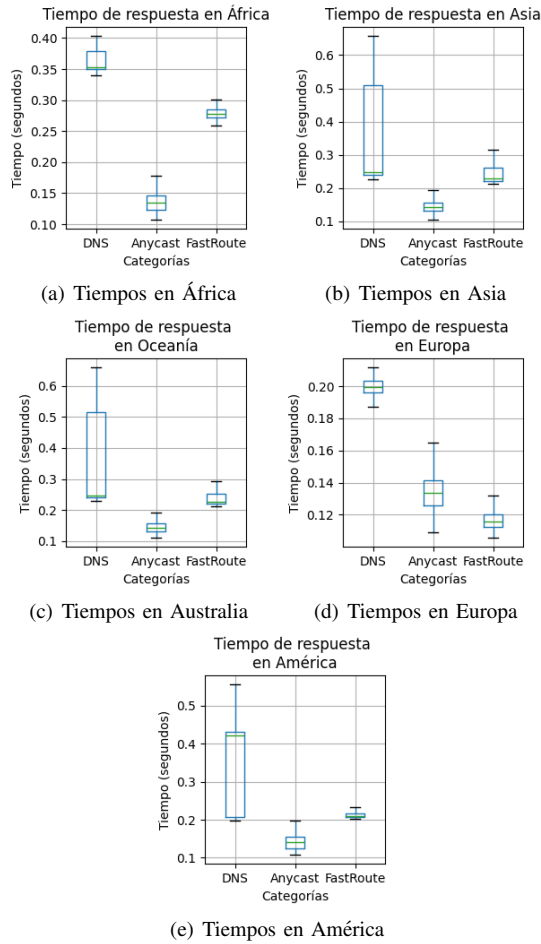


Fig. 2. Distribución de los tiempos en cada zona

servidores, ya se tiene una idea más clara de qué servidor asignarle al cliente.

Desde el punto de vista del ancho de banda utilizado para recibir los requerimientos, se obtiene que Anycast es capaz de llegar a valores más altos. Este resultado puede explicarse con que Anycast no requiere consumir tanto ancho de banda disponible en coordinar con varios equipos, mientras que DNS depende del mismo servidor DNS y FastRoute pide una comunicación constante entre los servidores del mismo nodo para efectuar el balanceo y distribución de carga entre las capas de servidores.

VI. CONCLUSIONES

La creación y establecimiento de una red de distribución de contenido con buena calidad de servicio es un desafío con bastantes variables a tomar en cuenta. La distancia en red, la calidad de la infraestructura y los niveles de exigencia que enfrentan los servidores son algunos puntos que pueden variar muchísimo la calidad que se le brinda a los clientes. Para asignar de la mejor forma posible, se usan comúnmente los métodos presentados cuya eficacia varía según las condiciones en las que se hagan los requerimientos.

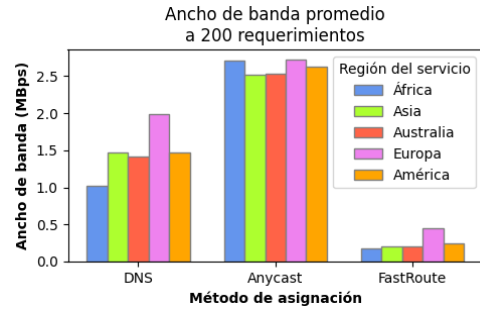


Fig. 3. Resultados del ancho de banda estimado luego de hacer 200 requerimientos

Los métodos con DNS demostraron funcionar mejor en zonas donde la red existente es sólida y poblada de infraestructura, pero presenta un espacio importante donde se pueden aplicar mejoras. Se debe tomar en cuenta, por ejemplo, que los equipos usando DNS remoto van a verse asociados a servidores cercanos al mismo DNS pero no necesariamente más cerca del cliente. El uso de Anycast presenta una forma distinta de trabajar e introduce ciertas mejoras, llegando a otorgar los menores tiempos y más altos anchos de banda. Dada la cantidad de requerimientos hechos, los servicios usando este esquema pudieron funcionar sin problema, pero queda por ensayar cómo sería su desempeño en situaciones donde se le exige más a los servidores.

Siguiendo con la línea de Anycast, FastRoute toma más en consideración la carga impuesta a los servidores. Esto permitió que los resultados obtenidos sean mucho menos dispersos que los resultados de los otros métodos, lo que sugiere que la carga hecha por clientes ajenos a estos experimentos fue compensada para que los recursos destinados a las máquinas virtuales usadas sean reservados de forma más eficiente.

Los anchos de banda utilizados por cada método dan una pista de cómo se asignan los recursos de la red para su funcionamiento interno y la comunicación con el cliente. Específicamente, en los resultados de la figura 3 se obtuvo que el ancho de banda utilizado para responderle al cliente fue inversamente proporcional a la complejidad de los pasos que deben seguir los esquemas.

VII. TRABAJO FUTURO

Aunque los resultados obtenidos mostraron el desempeño de redes ya existentes implementadas en sistemas funcionando, existe espacio para hacer más pruebas en otros entornos distintos a los usados.

En primer lugar, se puede implementar una red CDN en un simulador como Mininet o GNS3 usando un computador que disponga de suficiente potencia o un servicio de Cloud Computing que permita implementar varias máquinas que se comuniquen entre sí. Esto permitiría eliminar las incertidumbres generadas por intermitencias del servicio o sobrecargas según periodos del día donde más se hagan requerimientos por parte de otros usuarios.

En segundo lugar, está la opción de implementar las técnicas anteriormente mencionadas con el protocolo QUIC en lugar de TCP para aprovechar los handshakes de 0-RTT y analizar cuánta mejora se obtiene, tal como se propone en [5].

En tercer lugar, se plantea utilizar alguna técnica más nueva como Polygon [5], que agrega sensibilidad hacia los recursos que se le deben asignar a los clientes y qué servidores disponen de más de esos recursos. La implementación de este sistema podría hacerse con simuladores o en alguna red pequeña que disponga de suficientes máquinas.

En conjunto, el análisis y experimentación propuesta en estos puntos puede conducir a mejoras en el servicio de las redes de distribución de contenido, aprovechando de manera más óptima los recursos disponibles y entregando una mejor calidad de servicio a los usuarios que las utilicen.

REFERENCIAS

- [1] J. Kurose and K. Ross, "Computer networks: A top down approach featuring the internet," 2010.
- [2] Z. Wang, J. Huang, and S. Rose, "Evolution and challenges of dns-based cdns," *Digital Communications and Networks*, vol. 4, no. 4, pp. 235–243, 2018.
- [3] J. M. Evang, A. Gomola, and T. Čičić, "Anycast metrics and performance tuning," in *2024 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–6, 2024.
- [4] A. Flavel, P. Mani, D. Maltz, N. Holt, J. Liu, Y. Chen, and O. Surmachev, "FastRoute: A scalable Load-Aware anycast routing architecture for modern CDNs," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, (Oakland, CA), pp. 381–394, USENIX Association, May 2015.
- [5] M. Zhou, T. Guo, Y. Chen, Y. Li, M. Niu, X. Wang, and P. Hui, "Polygon: A quic-based cdn server selection system supporting multiple resource demands," *IEEE/ACM Transactions on Networking*, pp. 1–15, 2024.