Content Delivery Network (CDN)

Daniela Tomás
up202004946@edu.fc.up.pt

Diogo Nunes
up202007895@edu.fc.up.pt

João Veloso
up202005801@edu.fc.up.pt

***Abstract*—This report describes the initial phase of our project, which is setting up a Content Delivery Network (CDN) for web servers hosted on Google Cloud Platform (GCP) by using both DNS-based and Anycast-based CDN approaches.**

***Index Terms*—Content Delivery Network (CDN), Nginx, Google Cloud (GCP), DNS-based CDN, Anycast-based CDN**

I. Introduction

As the internet continued to evolve in both the quantity and size of content hosted on it, as well as the number of users accessing it, all the while the network and server capacity didn't grow in hand, this led to servers and the internet itself being overwhelmed by traffic, sometimes from physically far-away locations, which have well-known effects on latency and general experience. CDNs try to solve this by replicating content hosted on web servers in physically distinct locations strategically chosen to be closer to where a high volume of requests comes from, tackling both problems of distance and server load by having requests travel much shorter distances and taking workload away from the origin server into the replicas. This report includes a description of the technology, project objectives, tool and component identification, proof of feasibility, and a work plan for the remaining work.

II. Description of the Technology

In this section, we will look deeper into the underlying CDN architectures, [4], including DNS and Anycast based CDN, among others.

A. Content Delivery Networks (CDNs)

CDNs have changed the delivery of web content by providing faster and more reliable access to online resources. Essentially, CDN is a distributed network of servers that are placed strategically throughout different regions of the world and aims to improve online content performance by circumventing congested paths [3] and enabling users to get content from the closest server instead of having to fetch it from the original server.

1) CDN architecture [1], [4]: A CDN architecture involves several components, including content-delivery, request-routing, distribution, and accounting. First, the content-delivery component includes an origin server to store the original content and respective replica servers to provide copies of the content to end-users. Second, the request-routing system controls traffic from the user to relevant replica servers based on factors like proximity and server load and also maintains the content updated. The request-routing system works with the distribution system, which routes content from the origin server to the replica servers and ensures consistency across the network. And last is the accounting system, which tracks user access and usage patterns for billing purposes.

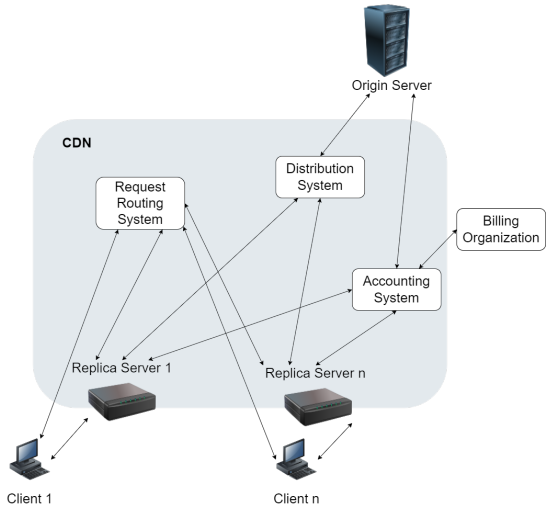

Fig. 1. CDN general architecture.

B. HTTP Redirection CDN

In this type of CDN, requests are made directly to the origin server, which then redirects the client to a new URL through HTTP. Since requests still have to initially travel all the way to the origin server, this makes it fairly inefficient, as the initial request still incurs a high latency penalty even if subsequent data benefits from being fetched from a closer server.

C. DNS-based CDN

DNS-based CDNs may differ in implementation specifics, but the base concept remains the same. In essence, a DNS server that is able to map the IP address on the client to a general geographical location with a precision level of up to a city answers with the IP of a replica server of the requested domain name based on the criteria of physical distance, server load, and possibly others. This has the much added benefit of not involving the origin server in the exchange, which may potentially be far away, assuming that the replica server already has the requested content available, which can increase the performance drastically, as if the origin server were much closer to the client. The major drawback of the DNS based CDN ends up being the DNS itself, as it can be costly to setup and maintain.

D. Anycast-based CDN

Anycast is a strategy in which the same IP address is announced in different geographical locations. Then, theoretically, packets make it to the closest presence of this address by means of the Border Gateway Protocol (BGP), with the help of Anycast-aware routers. This has a big advantage against DNS-based CDNs as there is no need for the potentially complex DNS sub-system, reducing costs, complexity, and centralization, which in turn makes the system more robust, but it also comes with disadvantages such as not being able to take server load into account, the inability of BGP to react to changes in network quality, and, according to previous studies [2], sometimes packets are routed to sub-optimal replica servers.

E. Peer-to-Peer CDN

Much like other Peer-to-Peer (P2P) technologies, such as torrents, content is distributed in a non-centralized manner, meaning there isn't a server to request content from; instead, it is fetched from other peers that either introduced the content to the network or have previously requested it themselves, distributing it to other nodes thereafter. Due to the decentralized nature of P2P technologies, this type is much more fault tolerant, as long as at least one member still has a given content, it continues to be available in the network, but it also shares the normal drawbacks of other P2P technologies, such as making it difficult to coordinate in large scale, high management complexity, as well as having security and legal/copyright infringement risks.

III. Project objectives

Primarily to build a DNS-based CDN for a web server, consisting of at least two replica servers with considerable geographical distance between them. Evaluating the performance differences between direct access to the origin and access to a closer replica with content already stored as well as not yet stored.

Time permitting, build an anycast-based CDN and compare performance to a DNS-based CDN.

IV. Tools and components identification

All of the project components are on VMs, hosted by Google Cloud Platform, the web server used is on F5s Nginx, DNS server is BIND and cache nodes are running varnish

A. Google Cloud Platform

Google´s cloud platform is an extremely versatile, powerful, and featured-packed tool that will be used to host all the necessary virtual machines. It was chosen for its high availability, fast performance, and worldwide presence, allowing us to deploy the whole CDN infrastructure all over the world, which enables us to have much more realistic test conditions.

B. Nginx

Nginx is a web server that also includes, among other functionality, the ability to act as a reverse proxy and was chosen by us to act as both the origin and replica servers due to its ease of use and performance.

C. BIND

BIND (*Berkeley Internet Name Domain*) is the one of the most used DNS software in the world and was choosen for its ease-of-use aswell as for its community GeoIP databases, which will prove vital in our DNS based CDN.

D. Varnish

Varnish is a reverse caching proxy software that will fetch content from our origin server and serve it to clients that are redirected to it. It was chosen for first and foremost designed as an HTTP accelarator contrary to competing software which has other, unnecessary in our use case, functionality.

V. Proof of feasibility

At the time of delivery, we have sucessfully deployed a simple CDN for our website hosted on www.tar-cdn.cloud . It consists of 2 DNS servers in a master-slave configuration running BIND nameserver, 1 cache node replicating content of the origin server with the use of varnish, located in europe, and a server running nginx located in japan. Requests incoming from european countries are routed to our cache node, while others are routed to the japan server.

VI. Work Plan for the remaining work

In the future there could be several ways to expand on this work such as:

- Adding support for video or big downloads.
- Expanding the overall infrastructure with more PoP.
- Experimenting with Anycast.

All of these will require some previous study to check if they are viable or not given the available time and resources.

As well as test performance of CDN vs No CDN.

VII. Work done

Like was stated previously, the project is composed by 5 nodes, 2 DNS nodes in a master-slave configuration, 2 caching nodes and 1 server nodes.

DNS nodes run BIND DNS server software and contain the GeoIP information for all the countries in an ACL , which allows us to route requests to our pretended caching nodes.

Caching nodes run varnish sofware, and periodically poll the server to check for updated content.

Server node runs NGINX and hosts a simple site just to serve as an example. Site is generated using a SSG (static site generator) using a tool called "hugo".

VIII. Conclusion

In conclusion, CDNs proved to be a vital part of the internet in its earlier exploding-growth days, as they helped alleviate congestion and enhance the experience in a way that is mostly to the average user and are to this day still a cornerstone for any website that expects high amounts of worldwide traffic.

REFERENCES

[1] Rajkumar Buyya, Mukaddim Pathan, and Athena Vakali. *Content delivery networks*, volume 9. Springer Science & Business Media, 2008.

[2] Matt Calder, Ashley Flavel, Ethan Katz-Bassett, Ratul Mahajan, and Jitendra Padhye. Analyzing the performance of an anycast cdn. In *Proceedings of the 2015 Internet Measurement Conference*, IMC '15, page 531–537, New York, NY, USA, 2015. Association for Computing Machinery.

[3] B Molina Moreno, CE Palau Salvador, M Esteve Domingo, I Alonso Peña, and V Ruiz Extremera. On content delivery network implementation. *Computer Communications*, 29(12):2396–2412, 2006.

[4] Gang Peng. Cdn: Content distribution network. *arXiv preprint cs/0411069*, 2004.