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

Introduction

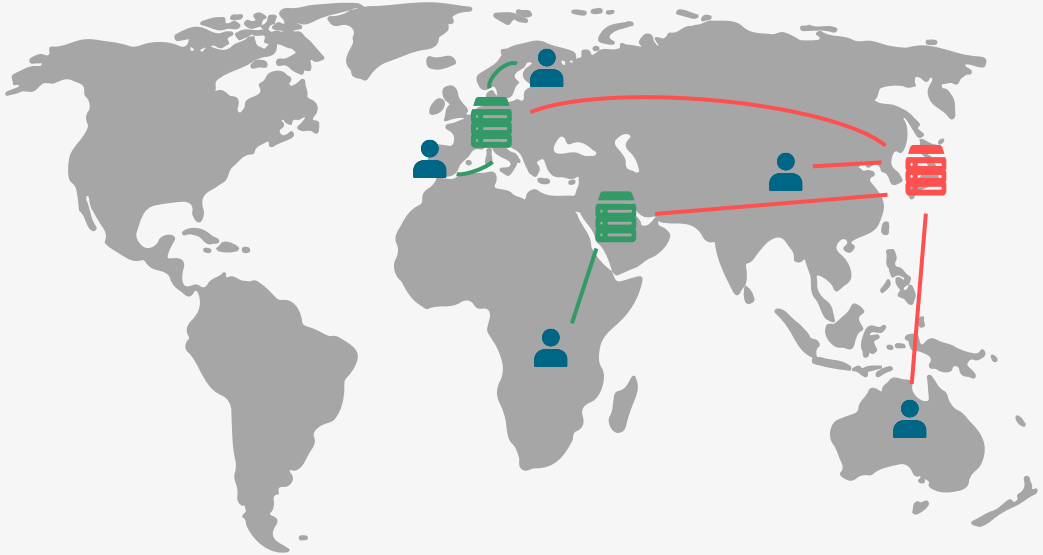
- The internet's growth in content and users has led to server overload and latency issues.
- CDNs aim to address this by replicating content in distinct locations, reducing server load and distance.
- The project aims to set up a CDN for web servers on Google Cloud Platform (GCP) to address these challenges.

CDN Overview

Origin Server

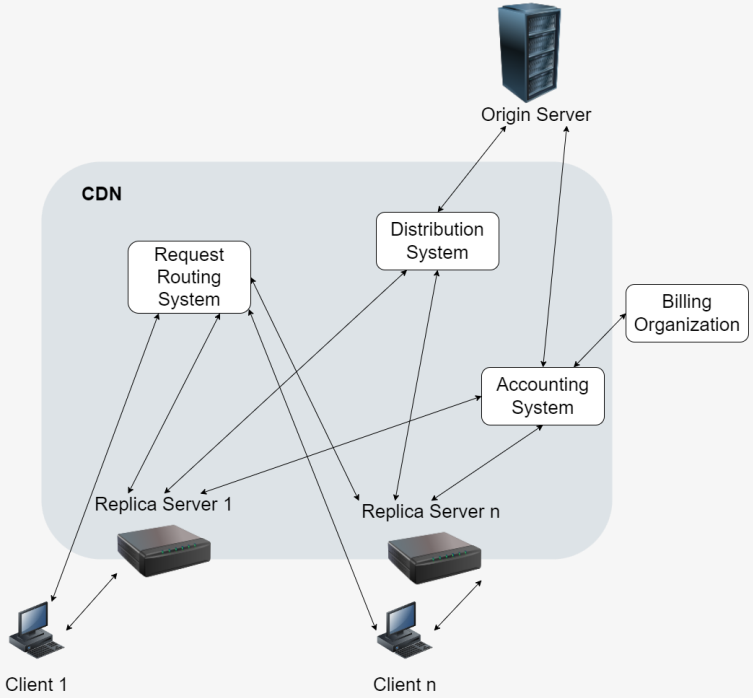
Edge Server

End User



CDN General Architecture

- CDN architecture includes four components :
  - Content-delivery
  - Request-routing
  - Distribution
  - Accounting.



Types of CDN

HTTP Redirection CDN

Requests are made directly to the origin server, redirecting the client to new URLs via HTTP, but incur high latency penalties, despite data fetching from closer servers.

DNS-based CDN

Routes requests to replica servers based on proximity and load, improving performance by avoiding origin server involvement.

Anycast-based CDN

Routes traffic to the closest server using BGP, reducing costs and complexity but may lead to sub-optimal routing.

Peer-to-Peer CDN

Distribute content non-centralized, fault-tolerant, and share drawbacks like coordination, high management complexity, and security risks.

Project Objectives

- Build a DNS-based CDN with at least two replica servers.
- Evaluate performance differences between direct access and CDN access.
- Explore Anycast-based CDN if time allows.

Tools and Components

- **Google Cloud Platform (GCP)** for hosting VMs.
- **NGINX** for web server functionality and reverse proxy.
- **BIND** for DNS services and GeoIP databases.
- **Varnish** for caching and content delivery.

Final Project Version

- **2 DNS Nodes (both in Europe):**
  - Configuration: Master-slave configuration using BIND DNS server software.
  - Functionality: Handle DNS resolution and route requests to appropriate caching nodes based on GeoIP information.
- **2 Caching Nodes (1 in Europe, 1 in Middle East):**
  - Configuration: Runs Varnish software.
  - Functionality: Fetch and cache content from the origin server periodically.
- **1 Server Node (in Asia):**
  - Configuration: Runs NGINX and hosts a simple static site using SSG using a tool called “hugo”.
  - Functionality: Hosts a static site as an example contente.

Performance Testing

- **Methodology:**
  - Tool Used: wrk (HTTP benchmarking tool)
  - Configuration: 400 simultaneous connections, 12 threads, 30 seconds per run.
- **Results:**

Location	Average Latency	Requests	Data Transmitted
Asia	254ms	46354	1.13GB
Europe	169ms	70325	1.72GB
Change	**-33.4%**	**+51.7%**	**+52.2%**

Conclusion

- In general, we think our project was well executed within what we had at our disposal, and the only thing we would have liked to perform was the implementation of an Anycast-based CDN so we could perform more comparisons with our DNS-based CDN, but unfortunately that wasn’t possible.

- 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

- Rajkumar Buyya, Mukaddim Pathan, and Athena Vakali. Content delivery networks, volume 9. Springer Science & Business Media, 2008.
- 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.
- 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.
- Gang Peng. Cdn: Content distribution network. arXiv preprint cs/0411069, 2004.