Evaluation of CDN's Design

Let's evaluate our proposed CDN.

We'll cover the following

Evaluation

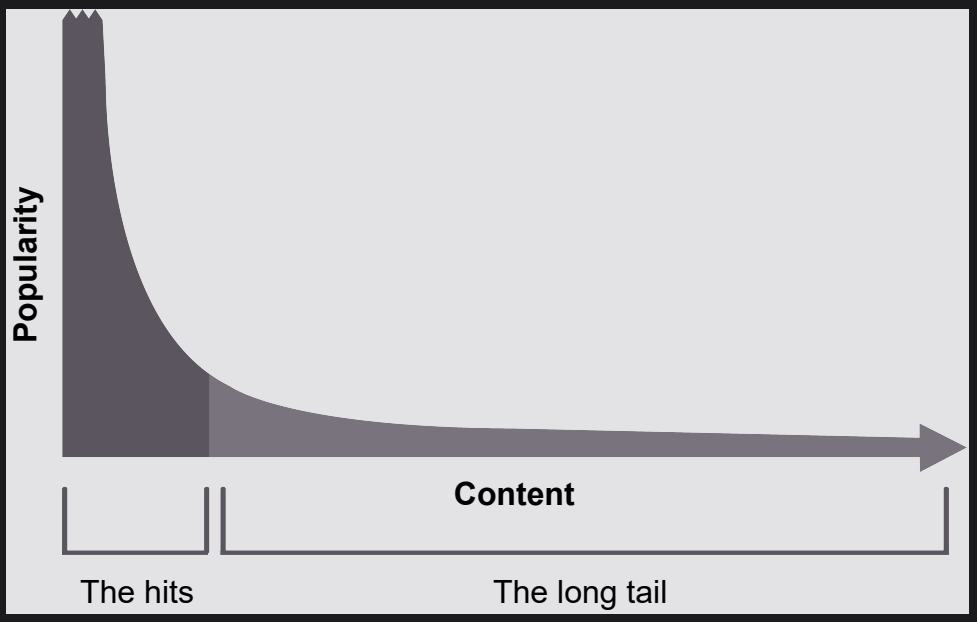
Here, we see how our design fulfills the requirements we discussed in the previous lessons. Our main requirements are high performance, availability, scalability, reliability, and security. Let’s discuss them all one by one.

Performance

CDN achieves high performance by minimizing latency. Some of the key design decisions that minimize latency are as follows:

- Proxy servers usually serve content from the RAM.
- CDN proxy servers are placed near the users to provide faster access to content.
- A CDN can also be the provider of proxy servers located in the ISP or **Internet exchange points (IXPs)** to handle high traffic.

- The request routing system ensures that users are directed to the nearest proxy servers, as discussed earlier.
- The proxy servers have long-tail content stored in nonvolatile storage systems like SSD or HDD. Serving from these resources results in a more negligible latency than we'd see from serving content from origin servers.



Long-tail content

- As was discussed previously, proxy servers can be implemented in layers where if one layer doesn't have the content, the request can be entertained by the next layer of proxy servers. For example, the edge proxy servers can request the parent proxy servers. Placing proxy servers at specific ISPs could be the best option when most traffic comes from those ISP regions.

Availability

A CDN can deal with massive traffic due to its distributed nature. A CDN ensures availability through its cached content that serves as a backup whenever the origin servers fail. Moreover, if one or more proxy servers in the CDN stop working, other operational proxy servers step in and continue to drive the web traffic. In addition, edge proxy servers can be made available through redundancy by replicating data to as many proxy servers as needed to avoid a single point of failure and to meet the request load. Finally, we can use a load balancer to distribute the users' requests to nearby active proxy servers.

Scalability

The design of CDN facilitates scalability in the following ways:

- It brings content closer to the user and removes the requirement of high bandwidth, thereby ensuring scalability.
- Horizontal scalability is possible by adding the number of reading replicas in the form of edge proxy servers.
- The limitations with horizontal scalability and storage capacity of an individual proxy server can be dealt with using the layered architecture of the proxy servers we described above.

Reliability and security

A CDN ensures no single failure point by carefully implementing maintenance cycles and integrating additional hardware and software when required. Apart from failures, the CDN handles massive traffic loads by equally distributing the load to the edge proxy servers. We can use scrubber servers to prevent DDoS attacks and securely host content. Moreover, we can use the **heartbeat protocol** to monitor the health of servers and omit faulty servers. Real-time applications also build their own specified CDNs to prevent content leakage problems and securely serve content to their end users.

Conclusion

Since its inception in the 1990s, the CDN has played a vital role in providing high availability and low-latency content delivery. Nowadays, CDNs are considered a key player in improving the overall performance of giant services.