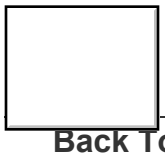


Join

Back To Course Home

Grokking Modern System Design Interview for Engineers & Managers

0% completed

Search Course

System Design Interviews

Introduction

Abstractions

Non-functional System Characteristics

Back-of-the-envelope Calculations

Building Blocks

Domain Name System

Load Balancers

Databases

Key-value Store

Content Delivery Network (CDN)

System Design: The Content Delivery Network (CDN)

Introduction to a CDN

Design of a CDN

In-depth Investigation of CDN: Part 1

In-depth Investigation of CDN: Part 2

Evaluation of CDN's Design

Quiz on CDN's Design

Sequencer

Distributed Monitoring

Monitor Server-side Errors

- [ ]

Monitor Client-side Errors

- [ ]

Distributed Cache

- [ ]

Distributed Messaging Queue

- [ ]

Pub-sub

- [ ]

Rate Limiter

- [ ]

Blob Store

- [ ]

Distributed Search

- [ ]

Distributed Logging

- [ ]

Distributed Task Scheduler

- [ ]

Sharded Counters

- [ ]

Concluding the Building Blocks Discussion

- [ ]

Design YouTube

- [ ]

Design Quora

- [ ]

Design Google Maps

- [ ]

Design a Proximity Service / Yelp

- [ ]

Design Uber

- [ ]

Design Twitter

- [ ]

Design Newsfeed System

- [ ]

Design Instagram

- [ ]

Design a URL Shortening Service / TinyURL

- [ ]

Design a Web Crawler

- [ ]

Design WhatsApp

- [ ]

Design Typeahead Suggestion

- [ ]

Design a Collaborative Document Editing Service / Google Docs

- [ ]

Spectacular Failures

Concluding Remarks

Course Certificate

Mark Course as Completed

System Design: The Content Delivery Network (CDN)

Understand what problems a CDN solves.

We'll cover the following

Problem statement#

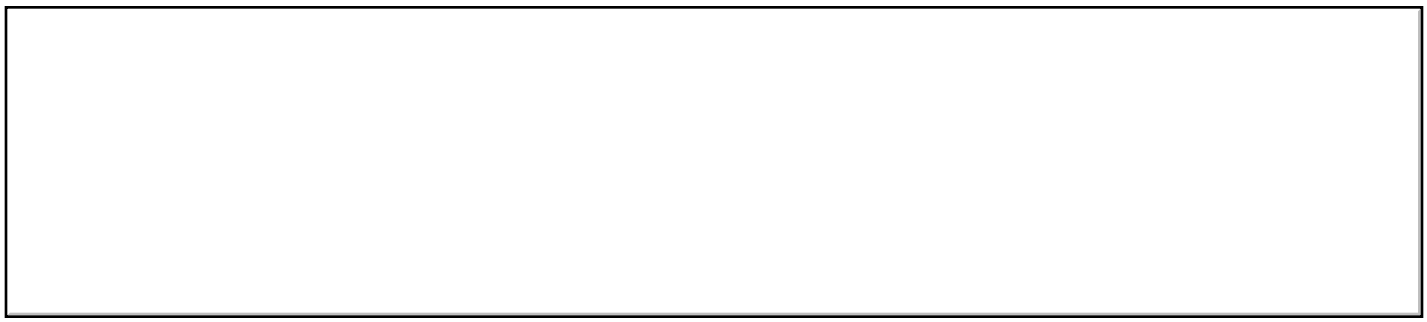
Let’s start with a question: If millions of users worldwide use our data-intensive applications, and our service is deployed in a single data center to serve the users’ requests, what possible problems can arise?

The following problems can arise:

- **High latency**: The user-perceived latency will be high due to the physical distance from the serving data center. User-perceived latency has many components, such as transmission delays (a function of available bandwidth), propagation delays (a function of distance), queuing delays (a function of network congestion), and nodal processing delays. Therefore, data transmission over a large distance results in

higher latency. Real-time applications require a latency below 200 milliseconds (ms) in general. For the Voice over Internet Protocol (VoIP), latency should not be more than 150 ms, whereas video streaming applications cannot tolerate a latency above a few seconds.

> **Note:** According to one of the readings taken on December 21, 2021, the average latency from US East (N. Virginia) to US West (N. California) was 62.9 ms. Across continents—for example, from the US East (N. Virginia) to Africa (Cape Town)—was 225.63 ms. This is two-way latency, known as round-trip latency.

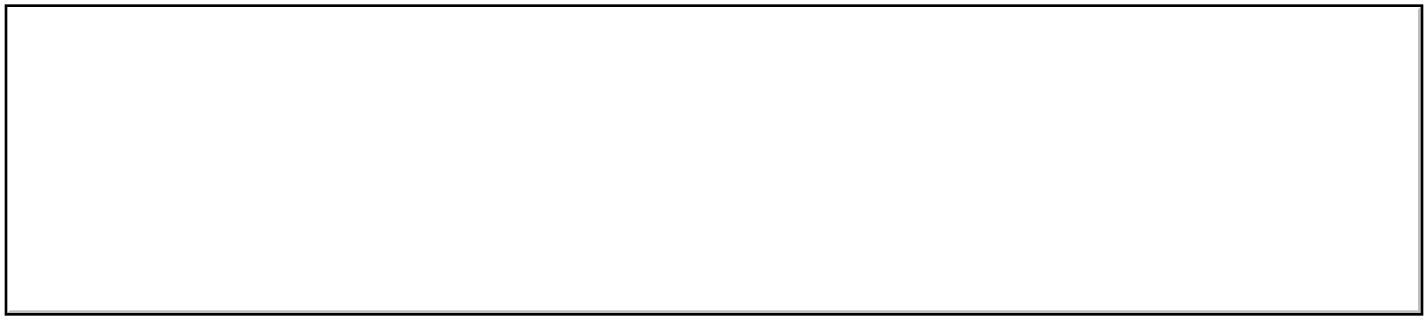


Origin data center entertaining users' requests across the globe

- **Data-intensive applications**: Data-intensive applications require transferring large traffic. Over a longer distance, this could be a problem due to the network path stretching through different kinds of ISPs. Because of some smaller Path message transmission unit (MTU) links, the throughput of applications on the network might be reduced. Similarly, different portions of the network path might have different congestion characteristics. The problem multiplies as the number of users grows because the origin servers will have to provide the data individually to each user. That is, the primary data center will need to send out a lot of redundant data when multiple clients ask for it. However, applications that use streaming services are both data-intensive and dynamic in nature.

> **Note:** According to a survey, 78% of the United States consumers use streaming services, which is an increase of 25% in five years.

- **Scarcity of data center resources**: Important data center resources like computational capacity and bandwidth become a limitation when the number of users of a service increases significantly. Services engaging millions of users simultaneously need scaling. Even if scaling is achieved in a single data center, it can still suffer from becoming a single point of failure when the data center goes offline due to natural calamity or connectivity issues with the Internet.



User growth over the years for Facebook and YouTube applications

> **Note:** According to one study, YouTube, Netflix, and Amazon Prime collectively generated 80% of Internet traffic in 2020. Circa 2016, the CDN provider Akamai served 15% to 30% of web traffic (about 30 terabits per second). For 90% of Internet users, Akamai was just one hop away. Therefore, we have strong reasons to optimize the delivery and consumption of this data without making the Internet core a bottleneck.

How will we design a CDN?#

We've divided the design of CDN into six lessons:

1. **Introduction to a CDN**: We'll provide a thorough introduction to CDNs and identify the functional and non-functional requirements.
2. **Design of a CDN**: We'll explain how to design the CDN. We'll also briefly describe the API design.
3. **In-depth Investigation of CDN: Part 1**: This lesson explains caching strategies

and CDN architecture. Also, we'll discuss various approaches to finding the nearest proxy server.

4. **In-depth Investigation of CDN: Part 2**: We'll discuss how to make content consistent in a CDN and the deployment of proxy servers. We'll also cover the custom and specialized CDN in detail.
5. **Evaluation of CDN**: This lesson will provide an evaluation of our proposed design.
6. **Quiz on CDN System Design**: We'll reinforce major concepts of CDN design with a quiz.

Let's think about the solution to the discussed issues in the next lesson.

Enable Fault Tolerance and Failure D…

Next

Introduction to a CDN

Mark as Completed

Report an Issue