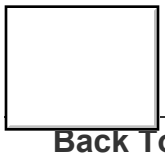


Back To Course Home

Grokking Modern System Design Interview for Engineers & Managers

0% completed

Search Course

System Design Interviews

Introduction

Abstractions

Non-functional System Characteristics

- Availability
- Reliability
- Scalability
- Maintainability

Fault Tolerance

Back-of-the-envelope Calculations

- [ ]

Building Blocks

- [ ]

Domain Name System

- [ ]

Load Balancers

- [ ]

Databases

- [ ]

Key-value Store

- [ ]

Content Delivery Network (CDN)

- [ ]

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

Mark Course as Completed

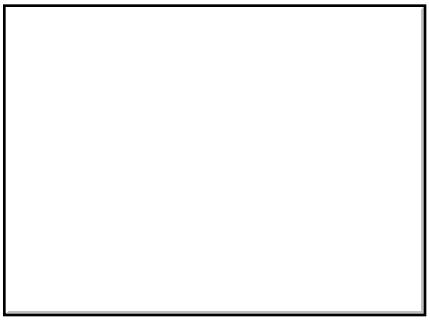
Scalability

Learn about scalability and its importance in system design.

We'll cover the following

What is scalability?#

Scalability is the ability of a system to handle an increasing amount of workload without compromising performance. A search engine, for example, must accommodate increasing numbers of users, as well as the amount of data it indexes.



The workload can be of different types, including the following:

- **Request workload**: This is the number of requests served by the system.

- **Data/storage workload**: This is the amount of data stored by the system.

Dimensions#

Here are the different dimensions of scalability:

- **Size scalability**: A system is scalable in size if we can simply add additional users and resources to it.
- **Administrative scalability**: This is the capacity for a growing number of organizations or users to share a single distributed system with ease.
- **Geographical scalability**: This relates to how easily the program can cater to other regions while maintaining acceptable performance constraints. In other words, the system can readily service a broad geographical region, as well as a smaller one.

Different approaches of scalability#

Here are the different ways to implement scalability.

Vertical scalability—scaling up#

Vertical scaling, also known as "**scaling up**," refers to scaling by providing additional capabilities (for example, additional CPUs or RAM) to an existing device. Vertical scaling allows us to expand our present hardware or software capacity, but we can only grow it to the limitations of our server. The dollar cost of vertical scaling is usually high because we might need exotic components to scale up.

Horizontal scalability—scaling out#

Horizontal scaling, also known as "**scaling out**," refers to increasing the number of machines in the network. We use commodity nodes for this purpose because of their attractive dollar-cost benefits. The catch here is that we need to build a system such that

many nodes could collectively work as if we had a single, huge server.

Vertical scaling versus horizontal scaling

Reliability

Next

Maintainability

Mark as Completed

Report an Issue