




Explore
(/explore)




Tracks
(/tracks)



My Courses
(/mycourses)



Edpresso
(/edpresso)



Refer a
Friend
(/refer-a-
friend)



Create
courses/grokking-
system-design-
/view/YQWGjIZz

Grokking the System Design Interview

(/collection/5668639101419520/56490502)

87% completed



Q Search Course

Glossary of System Design Basics



- System Design Basics
(/courses/grokking-the-system-design-interview/B892KY261z2)
- Key Characteristics of Distributed Systems
(/courses/grokking-the-system-design-interview/YQWGjIZZVz9)
- Load Balancing
(/courses/grokking-the-system-design-interview/3jEwI04BL7Q)
- Caching
(/courses/grokking-the-system-design-interview/3j6NnJrpp5p)
- Data Partitioning
(/courses/grokking-the-system-design-interview/mEN8IJXV1LA)
- Indexes
(/courses/grokking-the-system-design-interview/gxkVE8NEvXj)
- Proxies
(/courses/grokking-the-system-design-interview/N8G9MvM4OR2)
- Redundancy and Replication
(/courses/grokking-the-system-design-interview/xV1qvj6PKk)
- SQL vs. NoSQL
(/courses/grokking-the-system-design-interview/YQIK1mDPgpK)
- CAP Theorem
(/courses/grokking-the-system-design-interview/RMkqx1Egxqz)
- Consistent Hashing
(/courses/grokking-the-system-design-interview/B81vnyp0GpY)
- Long-Polling vs WebSockets vs Server-Sent Events
(/courses/grokking-the-system-design-interview/gx7wZzWn5Vj)

Appendix



- Contact Us
(/courses/grokking-the-system-design-interview/7nnxwPxAl7i)

Key Characteristics of Distributed Systems

We'll cover the following



- Scalability
- Reliability
- Availability
- Efficiency
- Serviceability or Manageability

Key characteristics of a distributed system include Scalability, Reliability, Availability, Efficiency, and Manageability. Let's briefly review them:

Scalability

#

Scalability is the capability of a system, process, or a network to grow and manage increased demand. Any distributed system that can continuously evolve in order to support the growing amount of work is considered to be scalable.

A system may have to scale because of many reasons like increased data volume or increased amount of work, e.g., number of transactions. A scalable system would like to achieve this scaling without performance loss.

Generally, the performance of a system, although designed (or claimed) to be scalable, declines with the system size due to the management or environment cost. For instance, network speed may become slower because machines tend to be far apart from one another. More generally, some tasks may not be distributed, either because of their inherent atomic nature or because of some flaw in the system design. At some point, such tasks would limit the speed-up obtained by distribution. A scalable architecture avoids this situation and attempts to balance the load on all the participating nodes evenly.

Horizontal vs. Vertical Scaling: Horizontal scaling means that you scale by adding more servers into your pool of resources whereas Vertical scaling means that you scale by adding more power (CPU, RAM, Storage, etc.) to an existing server.

With horizontal-scaling it is often easier to scale dynamically by adding more machines into the existing pool. Vertical scaling is

PK

educative(/learn)

Explore(/explore)

Tracks(/tracks)

My Courses(/mycourses)

Edpresso(/edpresso)

Refer a Friend(/refer-a-friend)

Create Courses(/courses/grokking-the-system-design-interview/YQWGjIZz)

Grokking the System Design Interview

(/collection/5668639101419520/56490502)

87% completed

Search Course

Glossary of System Design Basics

System Design Basics(/courses/grokking-the-system-design-interview/B892KY261z2)

Key Characteristics of Distributed Systems(/courses/grokking-the-system-design-interview/YQWGjIZZVz9)

Load Balancing(/courses/grokking-the-system-design-interview/3jEwl04BL7Q)

Caching(/courses/grokking-the-system-design-interview/3j6NnJrpp5p)

Data Partitioning(/courses/grokking-the-system-design-interview/mEN8IJXV1LA)

Indexes(/courses/grokking-the-system-design-interview/gxkVE8NEvXj)

Proxies(/courses/grokking-the-system-design-interview/N8G9MvM4OR2)

Redundancy and Replication(/courses/grokking-the-system-design-interview/xV1qvj6PKk)

SQL vs. NoSQL(/courses/grokking-the-system-design-interview/YQIK1mDPgpK)

CAP Theorem(/courses/grokking-the-system-design-interview/RMkqx1Egxqz)

Consistent Hashing(/courses/grokking-the-system-design-interview/B81vnyp0GpY)

Long-Polling vs WebSockets vs Server-Sent Events(/courses/grokking-the-system-design-interview/gx7wZzWn5Vj)

Appendix

Contact Us(/courses/grokking-the-system-design-interview/7nnxwPxAl7i)

regardless of the message size.

Size of messages representing the volume of data exchanges.

The complexity of operations supported by distributed data structures (e.g., searching for a specific key in a distributed index) can be characterized as a function of one of these cost units. Generally speaking, the analysis of a distributed structure in terms of ‘number of messages’ is over-simplistic. It ignores the impact of many aspects, including the network topology, the network load, and its variation, the possible heterogeneity of the software and hardware components involved in data processing and routing, etc. However, it is quite difficult to develop a precise cost model that would accurately take into account all these performance factors; therefore, we have to live with rough but robust estimates of the system behavior.

Serviceability or Manageability

#

Another important consideration while designing a distributed system is how easy it is to operate and maintain. Serviceability or manageability is the simplicity and speed with which a system can be repaired or maintained; if the time to fix a failed system increases, then availability will decrease. Things to consider for manageability are the ease of diagnosing and understanding problems when they occur, ease of making updates or modifications, and how simple the system is to operate (i.e., does it routinely operate without failure or exceptions?).

Early detection of faults can decrease or avoid system downtime. For example, some enterprise systems can automatically call a service center (without human intervention) when the system experiences a system fault.

← Back(/courses/grokking-the-system-design-interview/B892KY261z2)

Next →COMPLETED(/courses/grokking-the-system-design-interview/3jEwl04BL7Q)

Report an Issue

Ask a Question(https://discuss.educative.io/c/grokking-the-system-design-interview-design-gurus/glossary-of-system-design-basics-key-characteristics-of-distributed-systems)