AlgoTutor

SYSTEM DESIGN

IMPORTANT CONCETPS

USERS

LOAD BALANCER
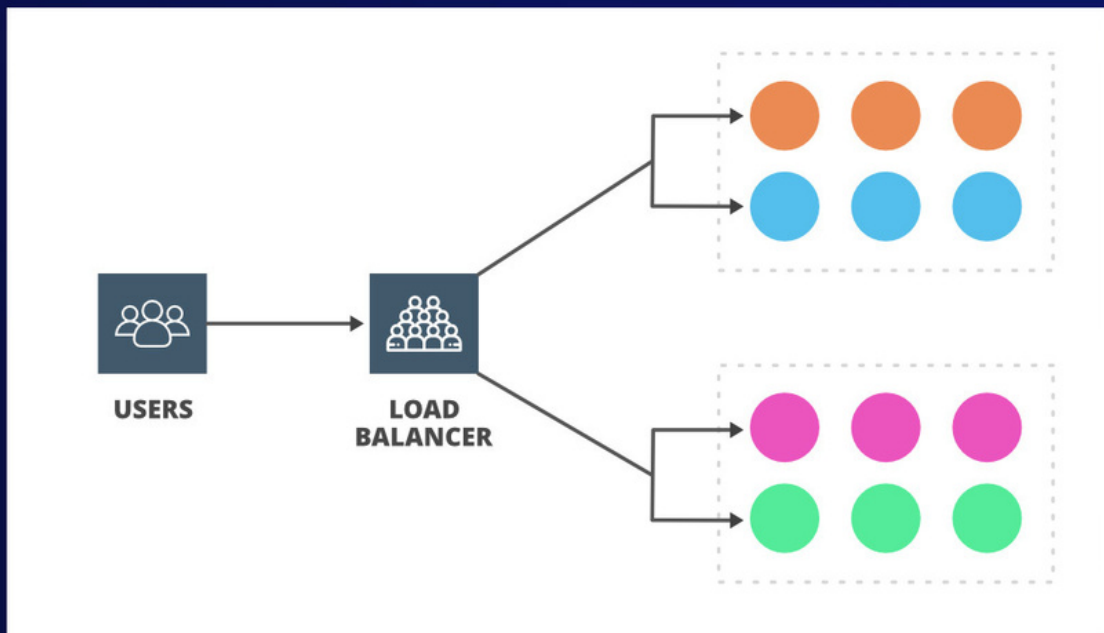
Introduction to System Design

System design refers to the process of creating a high-level plan or blueprint for the architecture, components, and interactions of a complex system.

It involves designing the structure, behavior, and functionality of the system to meet specific requirements and objectives.

💬Why System Design:

System design is essential because it enables scalability, efficiency, reliability, modularity, user experience, and effective collaboration.

It sets the foundation for successful system development and plays a crucial role in achieving the desired outcomes.

Examples:

- Google Material Design
- Apple Human Interface Guidelines
- Microsoft Fluent Design System
- Salesforce Lightning Design System
- Atlassian Design System
- Adobe Spectrum
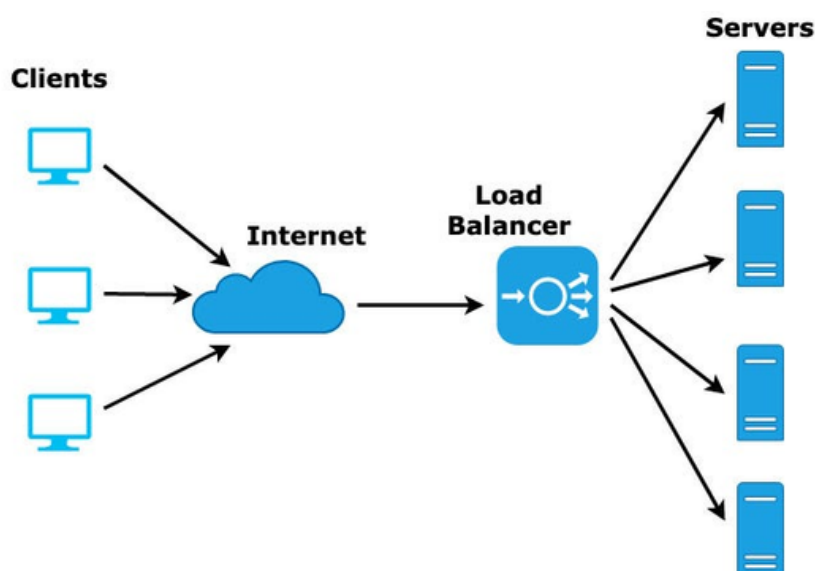
Load Balancer

❓ Problem Statement:

In system design, a common issue arises when a single server or resource becomes overwhelmed with incoming network traffic or workload, resulting in decreased performance, slow response times, and potential system failures.

This problem requires an effective solution to evenly distribute the workload across multiple servers or resources.

💡 Solution:

A load balancer acts as a traffic manager that evenly distributes incoming requests or workloads across multiple servers or resources in a system.

Its purpose is to optimize resource utilization, improve performance, and ensure high availability by preventing any single component from being overwhelmed.

Advantages of Load Balancer:

- Improved Performance
- Enhanced Scalability
- High Availability
- Efficient Resource Utilization
- Simplified Maintenance

want to Upskill Yourself ?

Explore our Popular Courses

Data Structure & Algorithms

Advance System Design (LLD + HLD)

Advanced Data Science & Machine Learning

MERN Full Stack Development

!! Visit Our Website !!

Caching

💬 Problem Statement:

A system is getting a lot of requests, and it is taking a long time to respond.
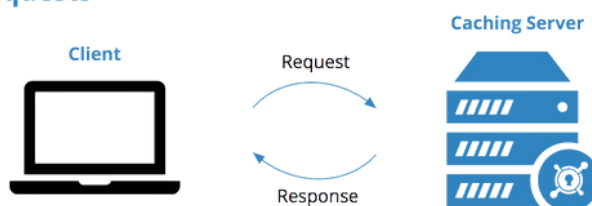
🧑‍🏫 Solution:

The system can use a caching system to store frequently accessed data in memory. This will reduce the amount of time it takes to respond to those requests, because the system will not have to go to the database to retrieve the data.

1st Request

Client → 1st Request → Caching Server → 1st Request → Origin Server

Client ← Response from origin ← Caching Server ← Asset is cached ← Origin Server

Subsequent Requests

Client → Request → Caching Server
Client ← Response ← Caching Server

💡 Caching is a technique that stores frequently accessed data in a temporary location, such as memory, to improve access speed.
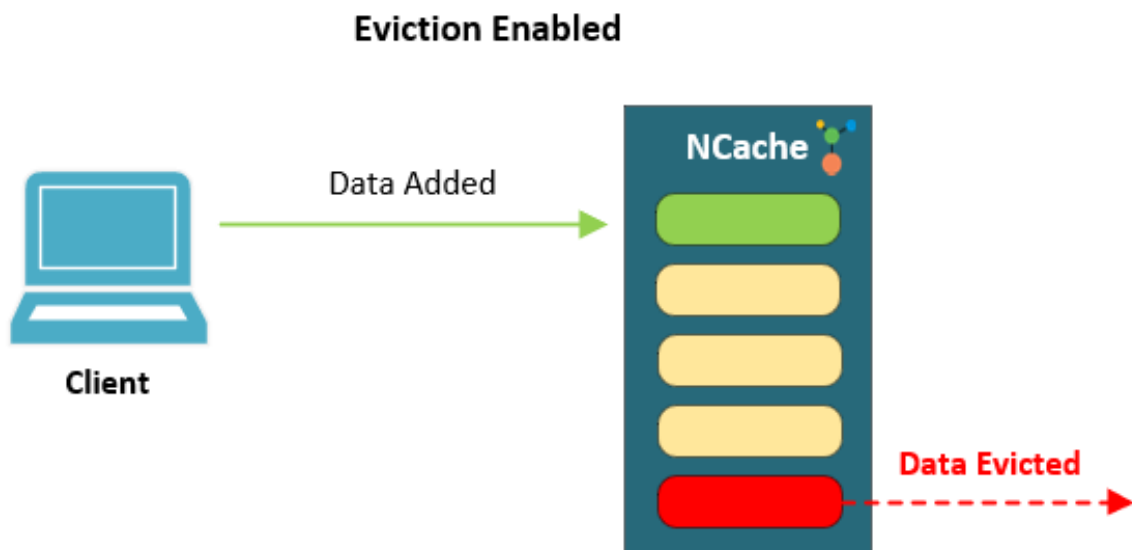
Cache Eviction Policies

💬 Problem Statement:

A system has a limited amount of cache space, and it needs to decide which data to evict when the cache is full.

💡 Solution:

The system can use a cache eviction policy to decide which data to evict. There are many different cache eviction policies, such as least recently used (LRU), least frequently used (LFU), and random.

Eviction Enabled

💡 A cache eviction policy is a strategy for deciding which data to remove from a cache when it is full

Explore Our Popular Courses

</> Data Structure & Algorithms

Advance System Design (LLD + HLD)

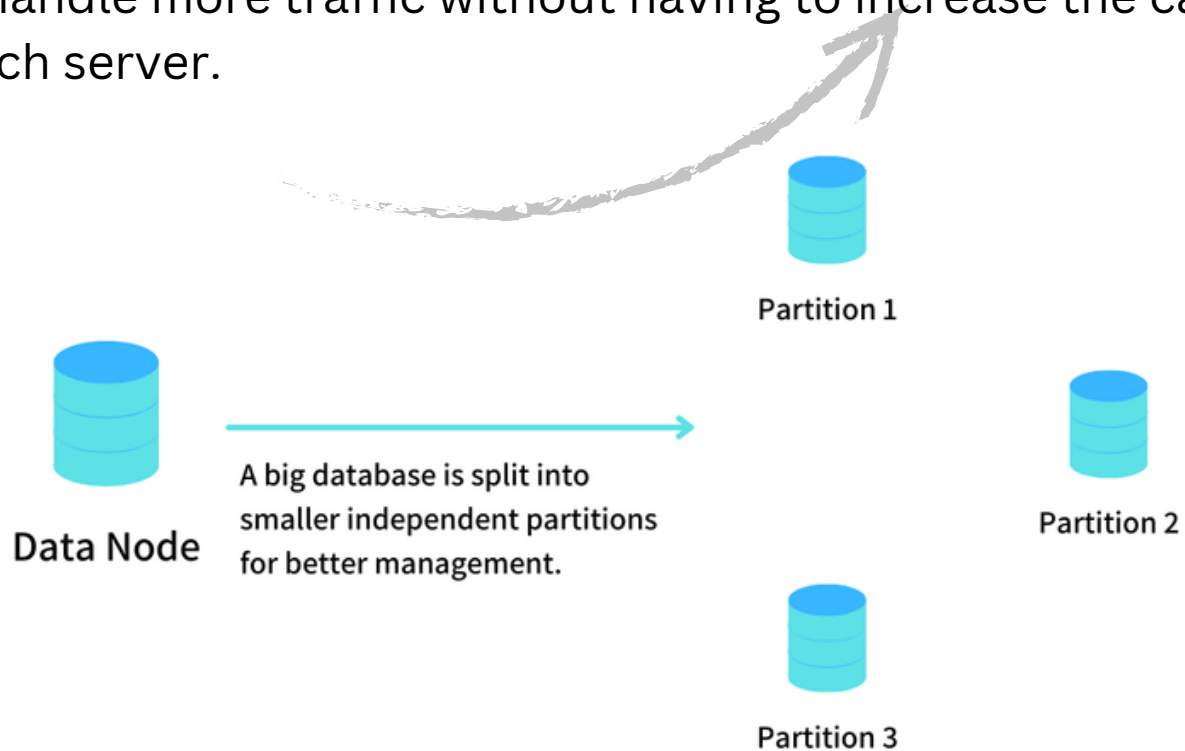Advanced Data Science & Machine Learning

MERN Full Stack Development

Data Partitioning

❓ Problem Statement:

A social media platform is experiencing high traffic and is struggling to keep up with the demand. The platform needs to find a way to improve performance without having to invest in more hardware.

💡 Solution:

The platform can use data partitioning to distribute the data across multiple servers. This will allow the platform to scale out and handle more traffic without having to increase the capacity of each server.

Partition 1

Partition 2

A big database is split into smaller independent partitions for better management.

Data Node

Partition 3

💡 Data partitioning is the process of dividing a large dataset into smaller, more manageable pieces. This can be done by dividing the data based on geographical location, time, or any other criteria that makes sense for the application

AlgoTutor

Why Choose AlgoTutor?

100 % PLACEMENT ASSISTANCE

1:1 PERSONAL MENTORSHIP FROM INDUSTRY EXPERTS

100 % SUCCESS RATE

23 LPA(AVG.)CTC

200+ SUCCESSFUL ALUMNI

147%(AVG.) SALARY HIKE

**LEARN FROM SCRATCH**

CAREER SERVICES

EXPLORE MORE

Data Redundancy

❓ Problem Statement:

A company is storing customer data in a single database. If the database fails, the company will lose all of its customer data.

💡 Solution:

The company can use data redundancy to store the customer data in multiple locations. This way, if one location fails, the data will still be available in another location.

Producers   Exchanges   Queues   Consumers
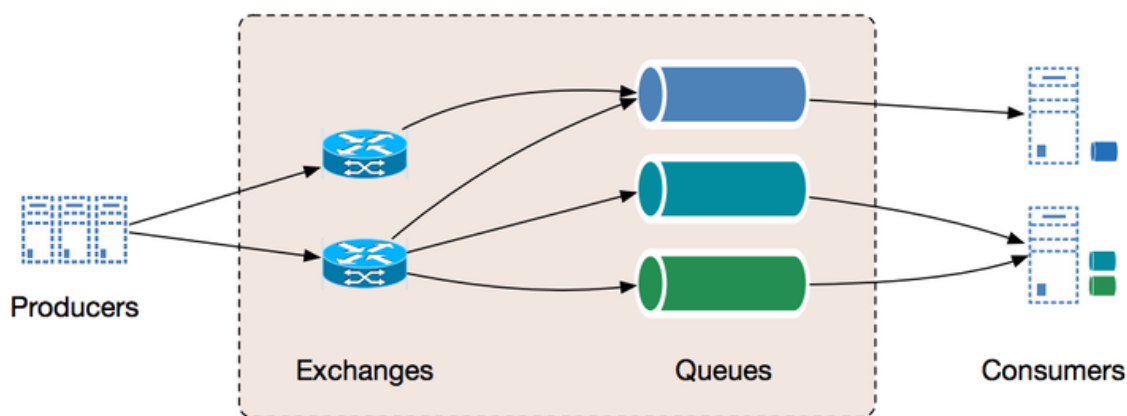
💡 Data redundancy is the occurrence of duplicate copies of similar data. It is done intentionally to keep the same piece of data at different places, or it occurs accidentally.

SQL & NO SQL

💬 Problem Statement:

A company is building a new social media platform. The platform will need to store a large amount of data, including user profiles, posts, and comments.

🖐️ Solution:

The company can use a combination of SQL and NoSQL databases to store the data. SQL databases are well-suited for storing structured data, such as user profiles. NoSQL databases are well-suited for storing unstructured data, such as posts and comments.

💡 SQL databases are based on a relational model, utilizing tables with predefined schemas to organize and store structured data.

💡 SQL databases, on the other hand, are designed to handle unstructured or semi-structured data. They use flexible schemas or no schemas at all, allowing for dynamic and scalable data models.
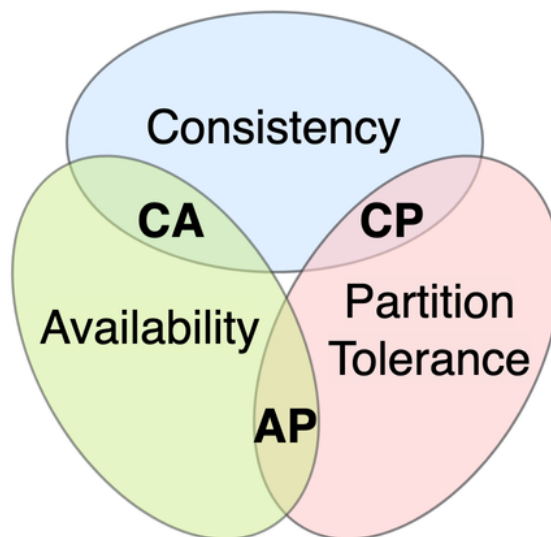
CAP Theorem

💬 **Problem Statement:**

The CAP theorem states that in a distributed system, it is impossible to simultaneously achieve consistency (all nodes see the same data at the same time), availability (system always responds to requests), and partition tolerance (system continues to operate despite network failures).

💡 **Solution:**

When designing a distributed system, the CAP theorem suggests that you must prioritize two out of three CAP properties based on your specific requirements.

💡The CAP theorem maintains that a distributed system can deliver only two of three desired characteristics, consistency, availability, and partition tolerance
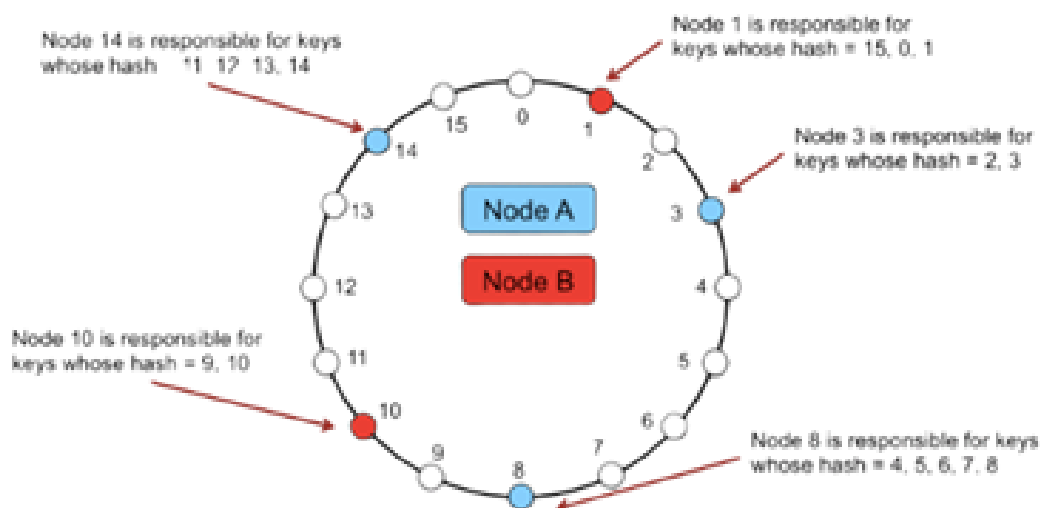
Consistent Hashing

💬 Problem Statement:

A large e-commerce website needs to distribute its traffic across a cluster of servers. The website wants to ensure that all requests are routed to the least loaded server.

🖐️ Solution:

The website can use consistent hashing to distribute its traffic. Consistent hashing is a technique that maps keys to servers in a way that ensures that the load is evenly distributed across all servers. When a request is received, the website can use the consistent hashing algorithm to determine which server to route the request to.

💡Consistent Hashing is a distributed hashing scheme that operates independently of the number of servers or objects in a distributed hash table by assigning them a position on an abstract circle, or hash ring.

Message Queues

💬 **Problem Statement:**

A large e-commerce website needs to process a large number of orders each day. The website wants to ensure that orders are processed in a timely manner and that they are not lost.

🧑‍🏫 **Solution:**

The website can use message queues to process orders. Message queues are a way of decoupling different parts of a system. In this case, the website can use message queues to decouple the order processing system from the web application.

💡A message queue is a form of asynchronous service-to-service communication used in serverless and microservices architectures.
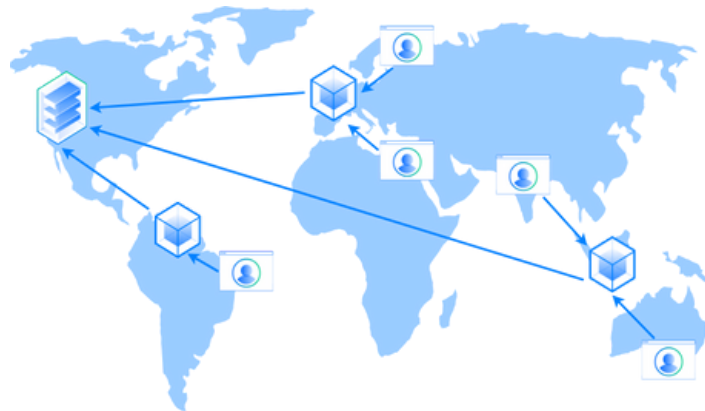
Content Delivery Network (CDN)

💬 **Problem Statement:**

A large e-commerce website needs to deliver its content to users all over the world. The website wants to ensure that users have a fast and reliable experience, regardless of their location.

🖐️ **Solution:**

The website can use a CDN to deliver its content. A CDN is a network of servers that are located all over the world. When a user requests a file from the website, the request is routed to the closest CDN server. This ensures that the user receives the file quickly and reliably.

CDNs are a popular solution for delivering large amounts of content to users all over the world. They can help to improve performance, availability, and reliability.

💡 A Content Delivery Network (CDN) is a geographically distributed network of servers that work together to deliver web content to users

ABOUT US

❖ AlgoTutor is an e-learning platform that provides students with the tools and resources they need to succeed in today's competitive World.

❖ Our courses are designed by experienced educators and industry experts.

❖ We offer a variety of learning formats, including video lectures, interactive exercises, and quizzes, so you can learn at your own pace and in your own way.

❖ AlgoTutor is also committed to providing students with the support they need to succeed. Our team of tutors is available 24/7 to answer questions and provide guidance.

USP of our Programs

100% Placement Assistance

1-1 personal mentorship from Indusrty experts

200+ Successful Alumni

147(Avg.)% Salary Hike

100% Success Rate

23 LPA (Avg.) CTC

Learn from scratch

Career Services

AlgoTutor

Start Your Professional Journey with Our Courses

DSA with System Design (HDL+LLD)

DSA & System Design with Full Stack

Advanced Data Science & Machine Learning

MERN Full Stack Web Development

Book a Free Demo Class & Get 10% Early Bird Discount

📞 +91-72600 58093

✉️ info@algotutor.io

🌐 www.algotutor.io