
System Design
Concepts
part -1

Scalability

- Scale to handling more number of requests
- Horizontal scaling
  - Adding nodes or instances
- Vertical scaling
  - More hardware resources in current machines

Data Partitioning

- Dividing large datasets into small manageable subsets
- To improve query performance, scalability and storage requirements.
- Sharding
  - Managing large databases into manageable subsets and hosting in different servers
- Partitioning
  - Dividing data into separate partition based on host , list and range.

Database Design

- Organize the data and create the relationship between them.
- Normalization - To reduce redundancy.
- Denormalization - To improve performance.
- CAP Theorem
    - To deliver any two characteristics from these three, consistency, availability, and partition tolerance.
- ACID - To ensure the sancity of the data.
- BASE - To handle large volumes of data and high concurrency

Caching

- Process of storing data in temporary storage and can be accessed in an easy way
- Consistency
  - Data should remain updated and consistent.
- Cache Eviction
  - Follow the policies or algorithm to manage data in cache.
  - LRU, LFU, MRU, FIFO
- Distributed cache
  - Store cache data across multiple servers

Consistency Models

- All nodes in a distributed system should have the same data at any time.
- Strong Consistency
  - All nodes have the same data at the same time.
- Eventual Consistency
  - To ensure the updates are reflected in all nodes.
- Causal Consistency
  - To ensure the sequence of events.