ONE PAGE SUMMARY

IT & Software › System Design › Caching Strategies

Introduction to Caching Strategies

Mastering Caching Strategies for Efficient System Design

DEFINITION

> Caching strategies are techniques used to store frequently accessed data temporarily for quick retrieval. They enhance system performance by reducing latency and load on the primary data source.

KEY CONCEPTS

Cache Types

Different types of caches include in-memory, disk, and distributed caches, each serving specific use cases.

Cache Hit vs. Miss

A cache hit occurs when requested data is found in the cache, while a miss means the data must be fetched from the original source.

Eviction Policies

Eviction policies like LRU (Least Recently Used) and FIFO (First In, First Out) determine which data to remove when the cache is full.

TTL (Time to Live)

TTL defines how long cached data remains valid before it is considered stale and removed.

Cache Coherency

Cache coherency ensures that all caches in a system reflect the most recent data state.

EXAMPLES

→ Web browsers caching images to speed up page loading.

→ Database query results stored in memory for faster access.

→ Content Delivery Networks (CDNs) caching website assets geographically.

MEMORY TIPS

★ Think 'Cache is King' for performance boosts.

★ Use 'Hit or Miss' to remember cache outcomes.

★ Remember 'LRU: Last Used, Remove' for eviction policy.

COMMON MISTAKES

✗ Neglecting to set appropriate TTL values.

✗ Overcomplicating cache designs with unnecessary layers.

✗ Failing to monitor cache performance and hit rates.

QUICK RECAP

Caching strategies improve system performance by temporarily storing frequently accessed data. Understanding cache types, hit/miss concepts, and eviction policies is crucial for effective implementation.