Notes - Session 6 - Caching Levels

DB Selection - Questions

Questions asked live, and answers shared in recordings

Caching Levels

Caching happens primarily at 4 different levels

- Client Side Caching
- Content Delivery Networks (CDNs)
- Remote Cache (Redis)
- Database Caching

Client Side Caching:

- Stores frequently accessed data on client side
    - Example: Web browsers (browser cache), mobile phones (app cache)

- Caches static (almost static) data such as images, js files, user-information (non-sensitive)
- Data invalidation is performed through time-based expiration policy

Advantages:

- Performance Improvement: Significantly increases the performance as no external request is made to fetch cached resources
- Reduces load time (Enhances load speed): Accessing resources locally is significantly faster, improving user experience
- Reduces server load: Request for resources can be satisfied by the local cache without a network request to the server

Considerations:

- Stale Data: Cached information might be outdated (stale); mechanisms are needed to validate and update cached data
- Cache Size Management: Local storage resources are limited, efficient cache policies must be enforced (example: LRU cache eviction policy)

Content Delivery Network (CDNs):

- CDNs are a set of servers distributed across the world, with the intent of a CDN (server or datacenter) being closer to a significant user base
- CDNs are used to provide high availability and performance by geographically distributing servers closer to end users'

Advantages:

- Improved website load times: By caching / bringing content closer to the users
- Reduced bandwidth costs: Through cache optimization, less data is transferred
- Enhanced content availability and redundancy: Even if one server fails, the network can redirect traffic to other servers

Considerations:

- CDN costs: Higher traffic sites benefit more from a cost perspective
- Data Security: Ensure secure transmission of data, consistent with privacy and security regulations

CDNs use PULL-based strategy for updating data in CDNs

Browser Caching + CDNs are a very strong way to reduce website response time, and reduce server load

Remote Cache (Redis)

Remote caching involves a dedicated server or cluster where applications can read and write data

Advantages:

- Speed: Redis maintains data in RAM, facilitating quick read and write operations
- Scalability: Easily scales with growing data needs, supporting high-throughput
- Data Structure Support: Supports data structures such as strings, hashes, lists and sets

Considerations:

- Persistence: Requires additional configuration to persist data, as Redis is primarily in-memory
- Memory Leak: Every key stored should have an expiration, else data will stay forever in Redis

- Redis is way too smaller in size compared to a DB, thus we have to be particularly careful about what we cache

Database Caching

- Involves storing results sets or frequently used computations to prevent complex DB queries
- Instead of running the complex queries involving joins, we can store the pre-computed results in materialised views or DB tables for faster response

Examples:

- Pre Computing news feed for social media users, top 20 posts for each daily active users