**Practice Problems**[Report Issue](https://github.com/LeetCode-Feedback/LeetCode-Feedback/issues)

Multiple Choice Question

Which of these are good use cases for implementing a cache?

Item information on an E-commerce app like Amazon

Prices in a tracker app for crypto currencies

Chats in a chat application like Whatsapp

User information like name, email, etcetera for users of the application

RedoSubmit

Correct.

The data for Item and User information does not frequently change for most users and is a good use-case for caching. For prices and chats, the data changes very frequently, and it would not be good to share stale data, so it’s not good to cache that information.

Multiple Choice Question

Which of the following should not be cached?

Results from a database call

Results from another service call

Aggregate statistics (like users of our product in last 7 days)

Simple mathematical calculations like (28+34)

RedoSubmit

Correct.

Computers are fast at doing calculations. If it’s faster to do a calculation than to make a network call to a cache and fetch the results, it’s best not to cache it. In this case, it’ll be much faster to do some mathematical calculation each time it’s requested than to store it in a cache and query the cache.

Multiple Choice Question

Suppose you are building a system like Gmail and want to decide on the storage for the image/video attachments. Which of the following is a good data storage to store such information?

CDN

Any Blob storage

MySQL Database

RedoSubmit

Correct.

The correct answer is any blob storage, which is meant for these kinds of files. You would not want to store this in a CDN because this isn't a use case where many users would want to see these attachments. Also, it’s generally a bad idea to store images and videos in a MySQL-like database.

Multiple Choice Question

Which is a good choice of database for building a payment system?

MySQL

Postgres

Redis

Elastic search

RedoSubmit

Correct.

Usually any Relational Database is considered good for payment like information, where we need the data to be consistent and allow atomic updates. Redis is more of a cache, and elastic search is more of a search engine, and probably not the best storage systems for Payment like information.

Multiple Choice Question

Say if you are building a group messaging application, and a user U1, wants to send a message M1 to a group G1 which has 50 users. You want to make sure that these messages are delivered to all the users whenever they are online. Which approach would you choose to send the messages to the remaining 49 users.

Synchronous approach

Asynchronous approach

RedoSubmit

Multiple Choice Question

What are some drawbacks of microservices based architecture?

Higher latency

Slow development cycle

High deployment time

High operational overhead

RedoSubmit

Correct.

Microservices have higher latency due to interservice API calls and high operation overhead due to the additional systems required to manage the microservices architecture effectively.

Multiple Choice Question

What are some properties of a good cache implementation?

High size

Low cache hit rate

Low size

High cache hit rate

RedoSubmit

Correct.

A good cache is always small in size and has a high cache hit rate, which makes it worth the investment.

Multiple Choice Question

Which protocol should you choose for communicating with clients when you are building an online shared document editor like Google Docs?

HTTP

WebSockets

SSH

RedoSubmit

Correct.

WebSockets is best here since the server might want to push the changes to the client that were done by some other user.

Multiple Choice Question

Suppose you are building a system like Twitter, and when a user tweets, their followers need to be notified about it. Should this use case be synchronous or asynchronous?

Synchronous

Asynchronous

RedoSubmit

Correct.

Asynchronous is the better approach because saving a tweet and sending notifications are very different things, and they don’t need to depend on each other.

Multiple Choice Question

Suppose you are building the login flow for your application. What entities would be good to rate limit on?

User id

IP Address

Operating system

App version

RedoSubmit

Correct.

We should rate limit on user id and IP address, to prevent brute-force attacks by a user or by a set of users from an IP. OS and App version would be the same for many users, and it might not add any value to rate limit on them.

Multiple Choice Question

Suppose you are building a Notification sending system that is meant for internal use by other services to send product notifications to users and is not exposed externally. Would you need to build a rate limiter for that?

Yes

No

RedoSubmit

Correct.

Yes, because we should have levers to control if one of the internal systems has bugs and starts sending a huge amount of notifications to users.

Multiple Choice Question

Should you talk about your class diagrams and function signatures in an HLD interview?

Yes

No

RedoSubmit

Correct.

No, because an HLD interview does not focus on your coding skills, but it’s more about your skills related to architecting large systems and their interactions. If the interviewer asks, then that’s different, and then you can get into those details.

Multiple Choice Question

Which protocol should you use for transmitting video content?

HTTPS

WebSockets

UDP

Telnet

RedoSubmit

Correct.

UDP, because it’s optimal for sending huge amounts of data and has very little overhead.

Multiple Choice Question

What additional systems you need to have in place for migrating to microservices from monoliths?

Centralized logging system

Load balancers

Deployment system

Service discovery

RedoSubmit

Correct.

Centralized logging and service discovery are things you additionally need when migrating to microservices. Load balancers and deployment system would be the same for microservices or monoliths.

Multiple Choice Question

If you are building a crypto trading app, which protocol should you choose for interactions with the user's app?

HTTP

WebSockets

HTTPS

RedoSubmit

Correct.

Websockets, because the server might want to send information to the client when the price changes.

Multiple Choice Question

What would be the impact of not choosing the right database when designing a system?

It does not have any impact.

The feature might work, but it would not be an optimal solution.

You would never be able to implement the feature with an incorrect database.

RedoSubmit

Correct.

With an incorrect database, you would still be able to build most of the features, but you’ll have to make this database work in ways it was not meant to work, which would be suboptimal.

Multiple Choice Question

What is wrong with MD5 hash function?

It is unstable and sometimes gives different output.

It can be reverse engineered.

It can have collisions.

It is too slow.

RedoSubmit

Correct.

It can be reverse-engineered, and that makes it bad for security-related scenarios. Also, it results in more frequent collisions than other common hash functions.

Multiple Choice Question

What’s a good data store if you want to provide text search capabilities?

Elasticsearch

Redis

Cassandra

MySQL

RedoSubmit

Correct.

Elasticsearch is built on top of Apache Lucene, which has great text searching capabilities.

Multiple Choice Question

If you are building a URL shortener, with 5 character short URLs consisting of numbers and uppercase alphabets [A-Z,0-9] , approximately how many unique URLs can your system support?

1.6 Million

60 Million

916 Million

1024 Million

RedoSubmit

Correct.

60 Million approximately, We would have 36 characters in our character set, A-Z(26) + 10 numbers. This would allow us to create 36^5 unique short URLs , which is equal to 60,466,176 unique URLs.

Multiple Choice Question

Suppose you need to report average temperature readings (over a few days) from IoT devices that send hourly temperature readings. Which database is optimal?

Cassandra

Influx DB

MySQL

Redis

RedoSubmit

Correct.

Influx DB is most suited for this kind of data access pattern. It is a time-series database optimized for aggregated reads and writes ordered by time and minimal updates.