**Performance Metrics of Components**

**Application**

Performance of an application can be judged by:

**API Response Time**: How optimized is the code to respond to a request. Amount of time taken to respond to a request.

**Throughput of APIs**: How many APIs in a given time, can an application support? It will depend on logic in code as well as the instance it is running on.

**Occurrence of errors**: If application throws a lot of errors and not able to handle edge cases, in that case, the application is not performant enough.

**Bugs/ Defects in code**: If an input causes an application to crash, the app is not resilient enough and not good in performance.

**Database**

How to measure performance of a database?

1. Time needed to output various database queries.
2. Number of queries executed per unit of time.

This performance depends on the code (from where read/write requests are coming) as well as the schema of database. How good is **schema**, how well it is **indexed**?

If your schema is designed in such a way that to serve a read, if multiple **joins** are required, then even if you have amazing infrastructure, your database will still take time to execute that query hence decreasing throughput.

**Cache**

Cache is in-memory data store having key-value pairs, how to measure its performance?

Latency of write operation: Time taken in doing writes.

Number of cache eviction and invalidations.

Memory of cache.

**Message Queues**

How to measure performance of message queues?

Rate of production and consumption of message.

Fraction of stale or unprocessed/ failed messages.

Number of consumers also affect throughput.

More **consumers**, more messages consumed per unit time, hence high **throughput**

But more consumers mean more resources consumed hence, **cost** also increased.

Hence we need to maintain that balance.

**Workers**

Just like application code, there are certain workers running within app or outside app, whose soul purpose is to perform a task.

There can be a worker whose purpose is just to pick up messages from queues and execute.

There can be worker to pull data from database and do some computation.

How to measure performance of workers?

Time taken to complete a job.

Amount of resources utilized in processing.

**Server Instances**

Every component we discussed till now, runs on some hardware.

Or any instance on cloud. Any instance requires 2 basic things: Memory and CPU.

An application needs some RAM/ Memory to store various data structures.

It also needs CPU to do computation.

If your application has memory leaks (or if you forget to free the memory allocation after use) then RAM usage can go high.

If your application does complex computation then CPU usage goes high.

**Performance Tools**

There are different Database performance management tools or application performance management tools.

Job of these tools is to show whole performance of the system.

Looking at different performance metrics, you can set alarms.

Ex: If memory usage for a database is going above 60%, raise an alarm.

Some tools are **New Relic, Vivid Cortex, and Data Dog**.

Some companies use their own tools.

AWS shows performs on Dashboard for different components used.