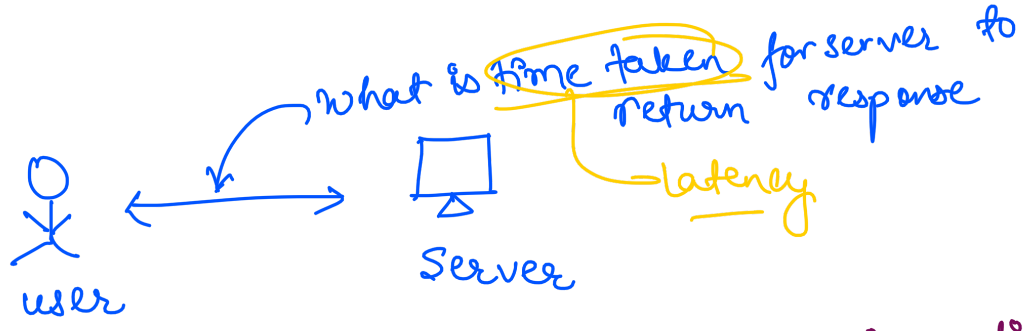
# Mystery of Time in Distributed Systems

## PART - 1

Subscribe for more such content

YouTube - MsDeep Singh

Motivation -



↳ There can be certain scenarios where we require time information in Distributed Systems.

another example → conflict resolution via Last write wins algorithm.

For time calculation, you need Clock

① Wall Clock

Java - System.CurrentTimeMillis()

↳ returns current time in milli seconds

↳ granularity depends on underlying OS.

→ In analogy to our wall clocks at home, we can reset it to some backward / forward time.

↑ Can system clock move forward / backward ?

dependent on system call - gettimeofday

It is affected by discontinuous jumps in System time.

example ⓐ Sysadmin modifies time

ⓑ NTP synchronization

Conclusion - Time can jump backward in wall clocks and might not be very suitable for calculation of time difference.

## ② Monotonic Clock

↑ ever increasing function → $f(x)_{T_2} > f(x)_{T_1}$ $(T_2 > T_1)$

Java - System.nanoTime()

↳ not related to any real time clock.

↳ used for timestamp comparison in same JVM instance.

↳ It is monotonic if support is available with OS.

Linux - clock-gettime

What about Distributed systems?

Example - Conflict resolution basis time on different machines

Last write wins algorithm

→ Monotonic clocks will not be useful here, we require actual time operation happened on different nodes.

What about time unsynchronization issues in machines?

→ Network Time Protocol (NTP) is in-general solution.

BTW NTP is not used for resolving conflicts, we'll be discussing more about NTP and other ways to calculate time in upcoming videos.

Thanks for watching

Happy Learning ☺