Subject ParallelComputing

Name Shakir Maute

1. Let me respond to each of your question.ForeignKey

Inside a refugee camp at metro station scenario:

A) Concepts of Computing and Optimization:

This can be formulated as a resource allocation and scheduling problem. This includes the following key computing concepts:

Since the people who wait for trains are crowding, it enables you to distribute them.

Load Balancing: Number of passenger distribute Trains and Rail.

Train scheduling algorithms Solution for train arrivals and departures.

Capacity planning optimization of the limited resource to operate trains on railways

For Throughput - To get the maximum number of passengers transported per unit time.

Parts to optimize:

Train capacity utilization

Railway usage efficiency

Passenger waiting time

Overall system throughput

B) Worst and best scenarios:

Worst scenario:

Many people burden the organization by just standing on roads of new city, unmanaged traffic (queue) leads to chaos and huge crowd.

Trains are not full and have long wait times

Some trains are packed others run empty due to lack of load balancing.

Not much throughput of passengers transported.

Best scenario:

Queue optimization Passenger Balancing

Schedule the trains in such a way that waiting time is minimized, and train utilization is maximized.

Balanced workload with railway and train-wise load pressure.

The maximum possible throughput is achieved in the constraints provided.

2.Critical components influencing performance and optimization:

a) Data storage:

Type of storage flash,

Storage capacity

Read/write speeds

File system efficiency

b) Data transfer:

Cost and performance: Processor speed as well an architecture

RAM capacity and speed

Speed(Bandwidth) of buses(memory bus, system buses etc.)

USB, Lightning port as I/O interfaces

c) Connectivity:

Modem/baseband processor

Antenna design and placement

Wi-Fi chip

Bluetooth module

NFC chip (if applicable)

So, optimization means driving the chips to peak performance and balancing these components for power/area constraints.

3. Throughput:

Server:

Powerful multi-core CPUs and parallel processing High Throughput

Workload specific hardware (network processors, storage controllers)

Storage and High-Bandwidth Memory

Designed for cyclical high volume applications

Metrics:

Possible Throughput: Hundreds of Thousands--or Millions for TPS

Aggregate data throughput = Gigabytes/second

Phone:

Bottlenecked by mobile SoC architecture and power limits

Designed for peak, not sustained highend throughput

Cost-effective memory and storage propositions but with the band-width reduced.

Metrics:

Chargebacks: Hundreds to Thousands of Chargeback Requests & TPS : Milliseconds or seconds The cost can range from hundreds into the five figure category for a single attack

Pass-through Rate: Low GB/s to Mid Hundreds of MB/sec

Scalability:

Server:

Vertical scaling: Upgrading to more power hardware (CPU, RAM and storage)

Auto-Scaling: Distributing load between multiple servers.

Load balancers for balancing traffic across instances

Scalability of data access through database sharding

Metrics:

Linear or near-linear scalability

Capacity to absorb increased online volumes at short notice

Phone:

Physical limitations in hardware scalability

Optimizations at the software level (eg faster algorithms, caching)

Utilizing cloud services to process when we are able

Parallel and delayed transaction processing

Metrics:

Software updates and optimizations for better performance

Proper load shedding mechanisms to allow for good experience even when under heavy load

Reliability:

Server:

Common hardware components (power supplies, storage, network interfaces)

RAID configurations for data backups

Error correcting code (ECC) memory

High Availability (failover systems, clustering)

Monitoring & Automatic error recovery

Metrics:

Why this matters Example Uptime percentage (ex 99.999% or "five nines")

Mean Time Between Failure (MTBF)

Mean Time To Recovery (MTTR)

Phone:

Constrained by available space and cost, hardware redundancy is limited

Handling (Or Not Handling) Errors In Software

Integrity checks and automatic retries for data

Offline Functionality and Synchronizing Data Offline.

Metrics:

App crash rate

of successful transactions completed

Maintaining Data Integrity for killed sessions.

Key differences:

Throughput: Servers can deliver orders of magnitude more throughput by virtue of their large hardware footprint and lack-of-power constraints. Desktop is more focus on battery and long time computation. Phones are just optimize for power efficiency and burst performance so primarly build that into mind while designing i guess in QA stage they kind of realize, Oops performance deployed!

Scalability: Servers can scale either through hardware upgrade, scale-out using distributed computing. The solution is software optimizations and offload processing to cloud for phones.

Availability: Servers feature redundant hardware and software in their design for high reliability. At the end of day, it's more a reliability and graceful command under load at software-level for phones.

For processing 1 billion transactions:

* A server or server cluster could potentially handle this in real-time or near real-time, with high throughput and reliability.
* A phone would likely need to process these transactions over an extended period, possibly offloading some processing to cloud services, and implementing robust error recovery and data integrity mechanisms to ensure reliability over the long processing time.