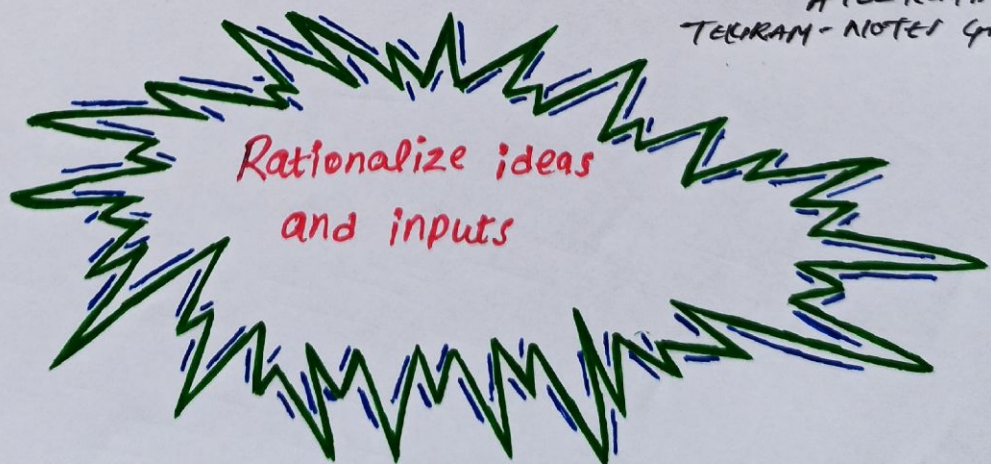
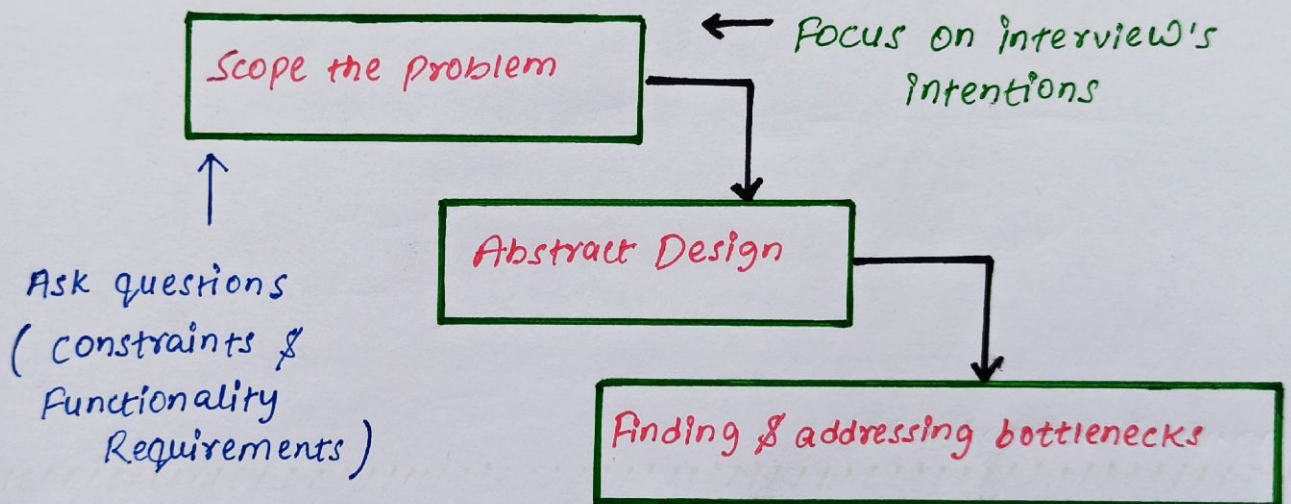


SYSTEM DESIGN NOTES

System Design Basics

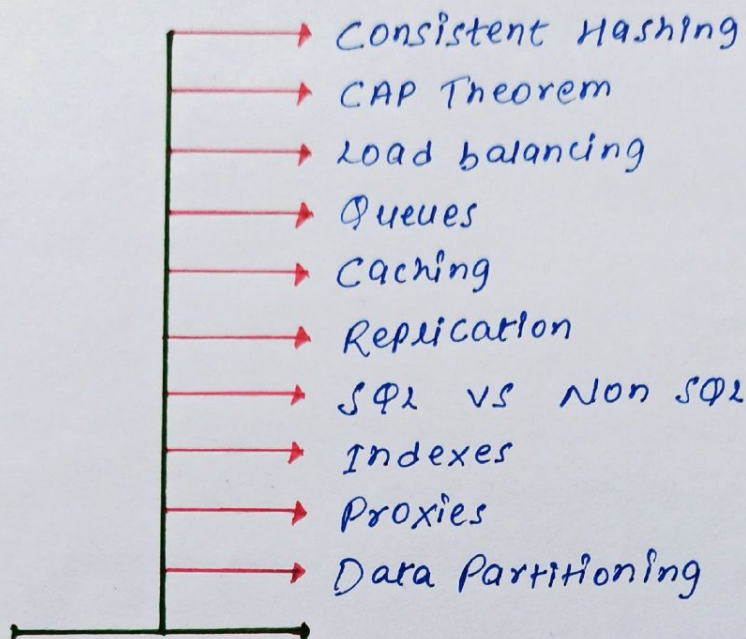
- 1). Try to break the problem into simpler modules
(Top down approach).
- 2). Talk about the trade-offs
(No solution is perfect)

Calculate the impact on system based on all the constraints and the end test cases.



ATUL KUMAR (LINKEDIN)
TECHRAM - NOTES GALLERY.

- 1). Architectural pieces / resources available .
- 2). How these resources work together .
- 3). Utilization & Tradeoffs

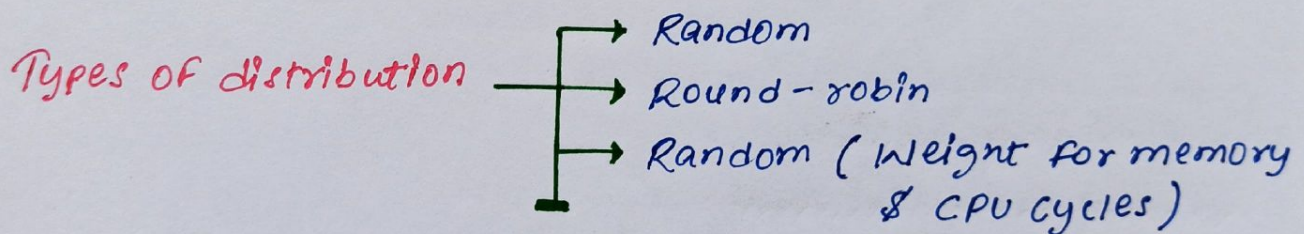


ATUL KUMAR (LINKEDIN).
TELEGRAM - NOTES GALLERY.

Load Balancing

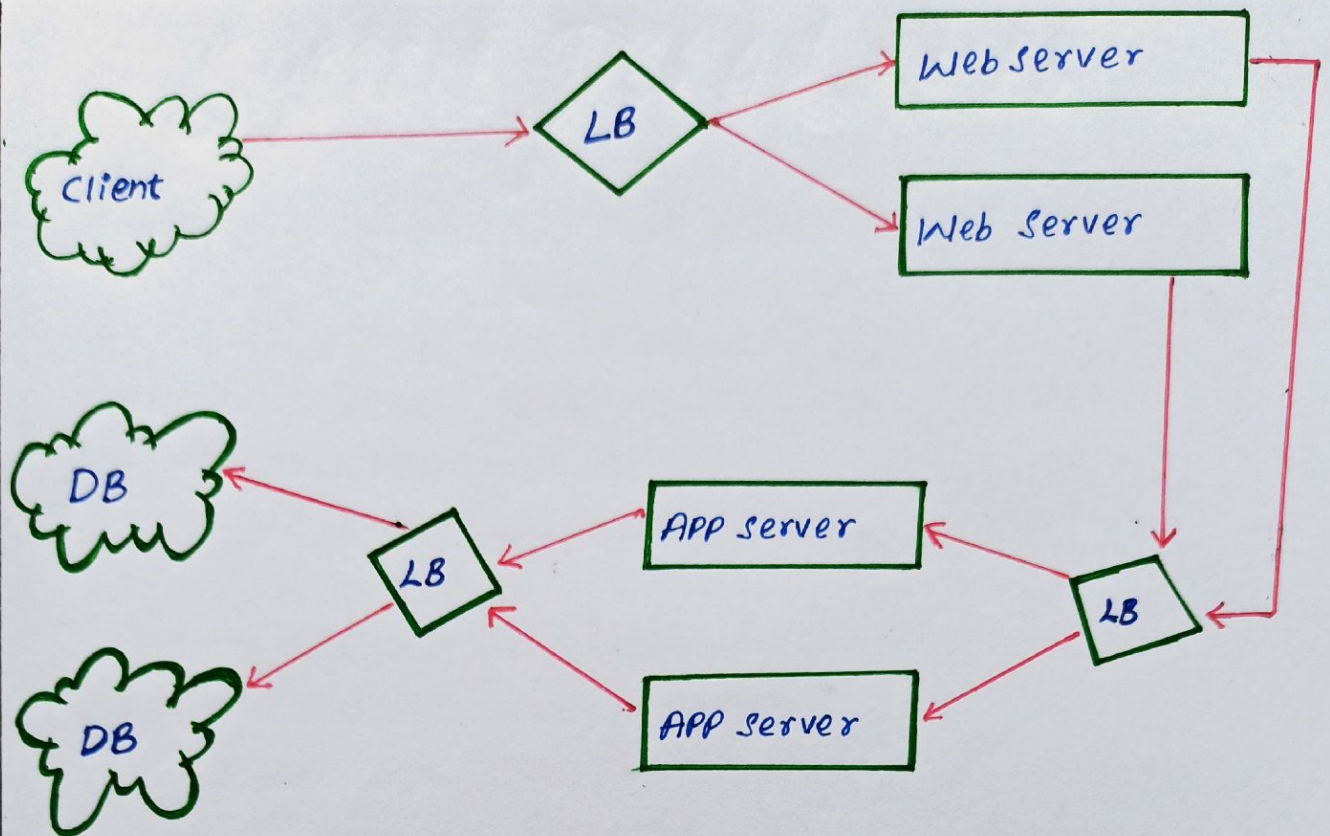
(Distributed System)

Types of distribution



To utilize full scalability & redundancy , add 3 LB

- 1). User $\xleftrightarrow{\text{LB1}}$ Web Server
- 2). Web Server $\xleftrightarrow{\text{LB2}}$ App server / Cache server
(Internal platform).
- 3). Internal Platform $\xleftrightarrow{\text{LB3}}$ DB.



Smart Clients

Takes a pool of services hosts & balances load.

- detects hosts that are not responsive
- recovered hosts
- addition of new hosts

Load balancing functionality to DB (cache, services)

★ Attractive solution for developers.

(small scale systems)

As system grows → LBs (standalone servers)

Hardware Load Balancers :

Expensive but high performance

eg. Citrix Netscaler

Not trivial to configure.