

High Level Design - 17 classes

curriculum overview - end of class

Pragy

Why High Level Design

→ ensure that future changes don't surprise us
Design & Scale

- ↳ Low Level
 - design & architecture of code
 - Scalable code
 - how large the codebase is
 - how many devs are involved
- ↳ High Level
 - design & architecture of infrastructure
 - Scaling the infra
 - how many users
 - how much data
 - how many requests/sec (load)

Staff Engineer @ Google (10^4 yrs)
 (1 Cr^+ base)

Given a file with 'strings' sort these strings alphabetically
Search queries

Catch - you have 50 PB of data = 50,000,000 GB of data.

8 bits - 1 byte

1000 bytes - 1 KB

1000 KB - 1 MB

1000 MB - 1 GB

1000 GB - 1 TB

1000 TB - 1 PB

Distributed Systems

multiple machines working together towards a singular goal
↓
Connected via some n/w

- ① Machines can restart
- ② n/w can be down
- ③ n/w can be congested / slow
- ④ Requests can time out
- ⑤ HDD failures
- ⑥ natural disasters
- ⑦ Computation errors / data corruption
- ⑧ Data Consistency

despite all
this we
wish to solve
our problems
reliably

HDD - scaling & infra challenges that arise
as we go from 100 users to billion users

Del.icio.us

- bookmarking website

Joshua - 2003

Youtube - 2004

AWS - 2006

Google Chrome - 2008

My first PC

in 2007

35,000 Rs/-

128 MB RAM

40 GB HDD

2 core CPU

8 Kbps dial up n/w

Idea - a website to store your bookmarks
so you can access them from any machine

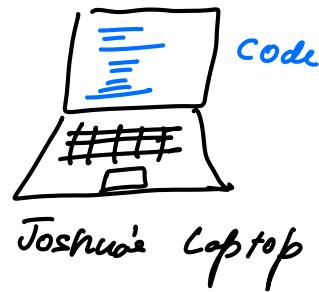
Minimal Viable Product (MVP) - proof of concept

↳ no fancy features
↳ any feature that you can implement later must be left for later

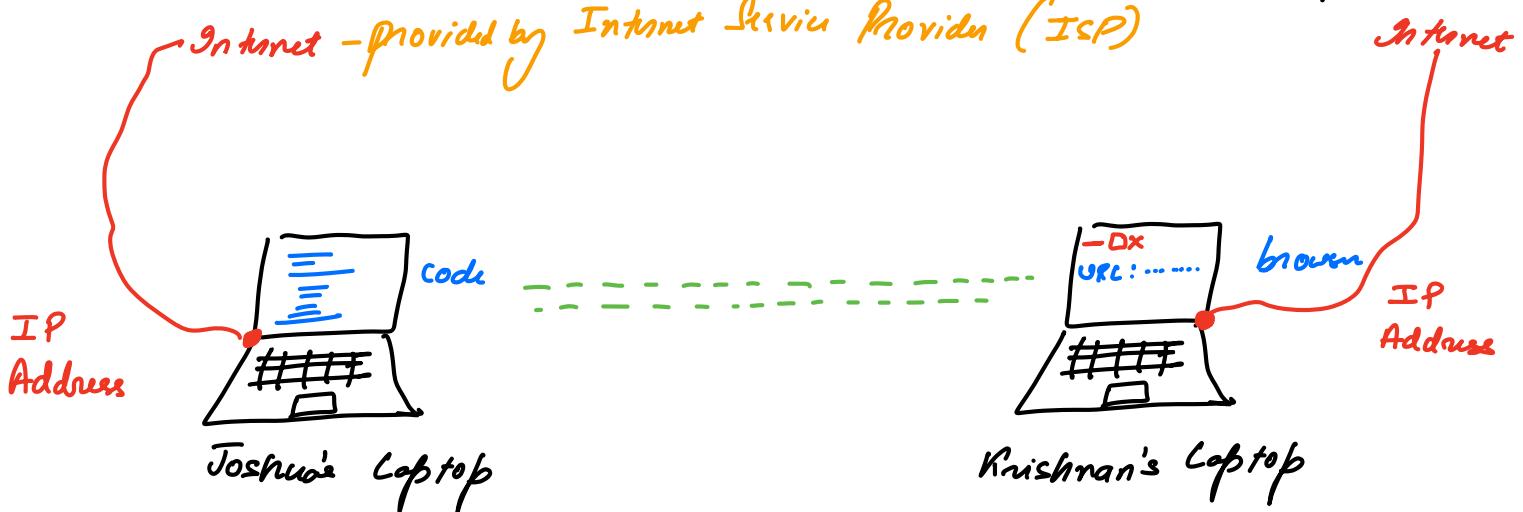
- ① Identify the user - register & login
- ② add Bookmark (user-id, URL)
- ③ get My Bookmarks (user-id) → List of URLs

folders / categories / tags
delete / update / rename
logout } later

Python + Django } with code
 Vanilla Javascript



Internet - provided by Internet Service Provider (ISP)



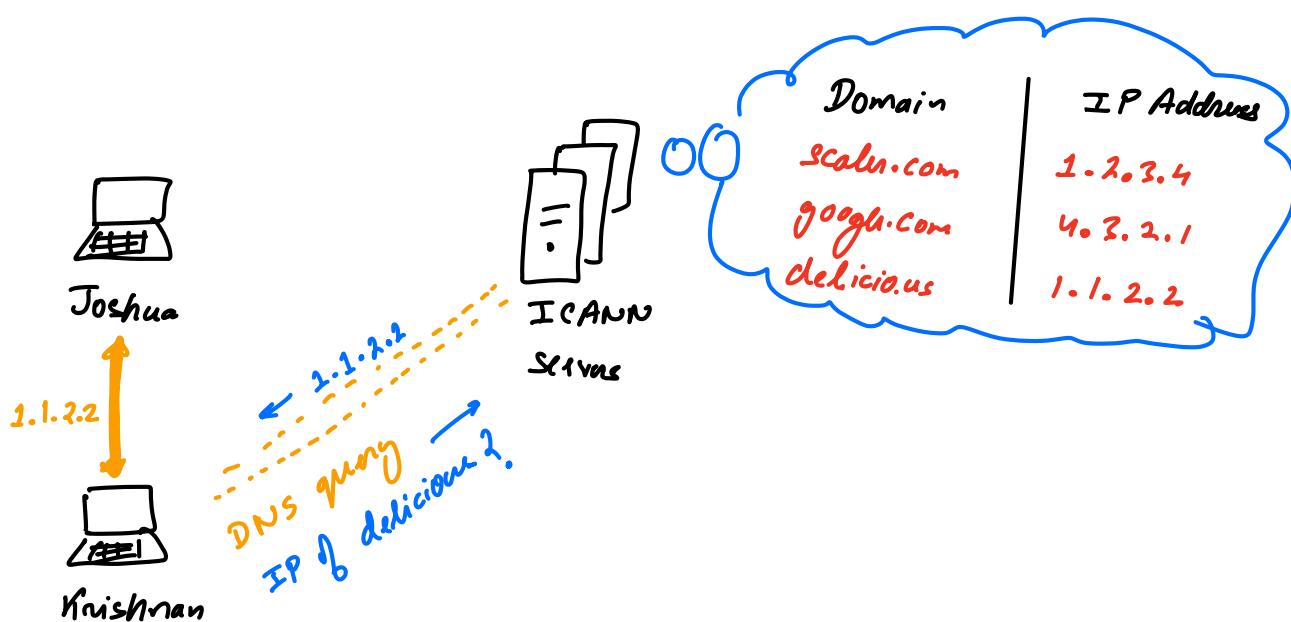
ISP examples - ACT / Airtel / Jio / Comcast / Mayai / AT&T /
 Starlink

- ① Krishnan's Laptop must know the IP Address of Joshua's Laptop
- ② Krishnan - open browser
 - enter website name <https://del.icio.us/>

Domain Name Service (DNS)

ICANN - authority for all domains & websites on the internet

Domain registrars - godaddy / namecheap / Route 53) ...
Purchase a domain \$



- ① Since there are billion of users & trillion of devices on internet ($\approx 5B$)



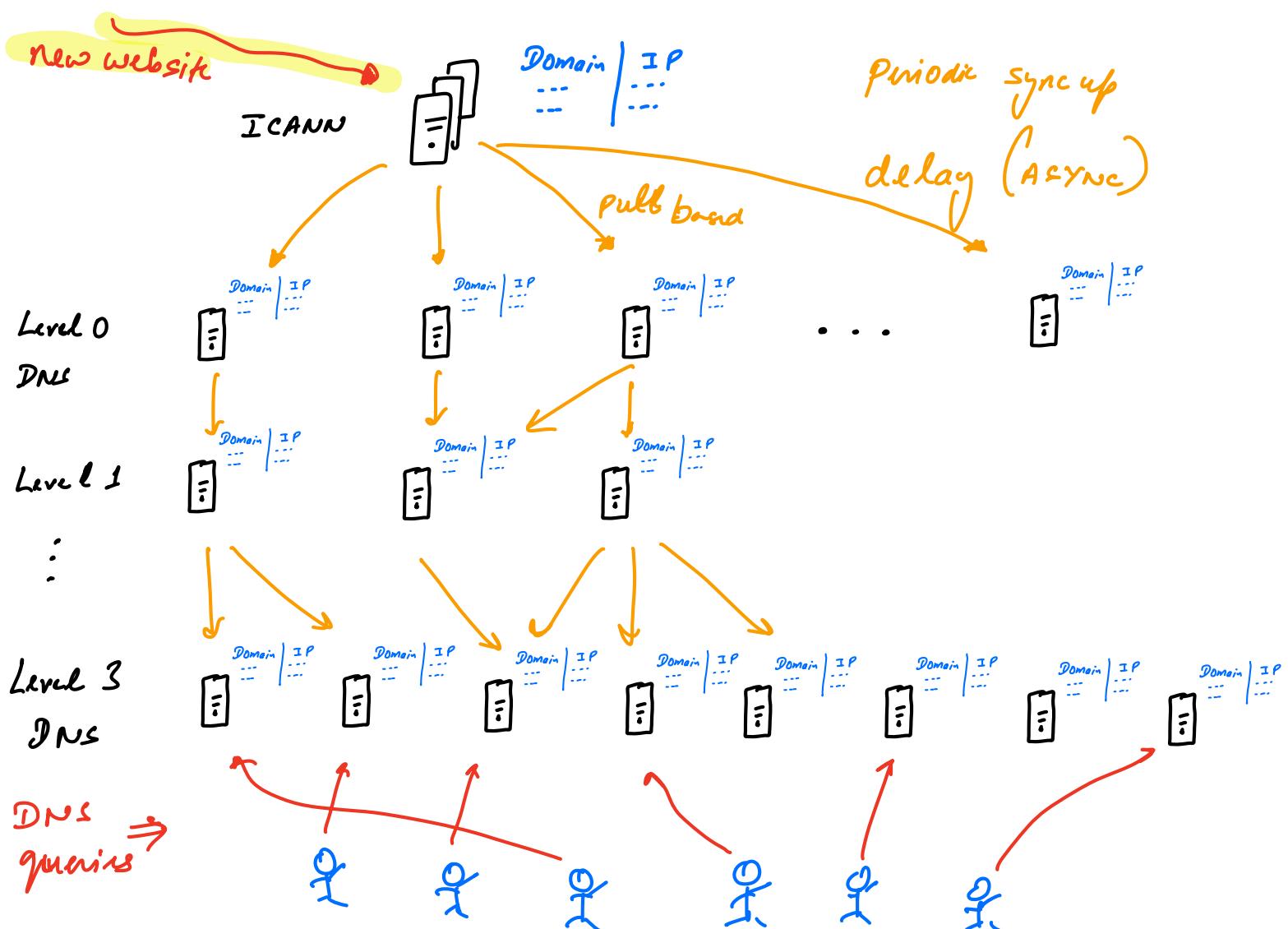
- Bottleneck** - Servers are not able to handle the overwhelming load

- ② if ICANN servers crash - entire internet goes down
Single Point of Failure (SPOF)

Lots of companies rested in the existence & smooth functioning of internet

ISP / Google-Amazon-Apple / Militaries / Research Orgs / NASA
Government / Universities

∴ these companies set up extra DNS servers



- ① Any question in HLD class - asked publicly
- ② Any question - posted only in Questions tab

Q How does your laptop know which DNS to connect to?

- Level 3 DNS only
- by default ISP sends a config packet + your routes to configure the default DNS

Tata / Jio - website blocked

ISP's own DNS

most likely blocked via DNS

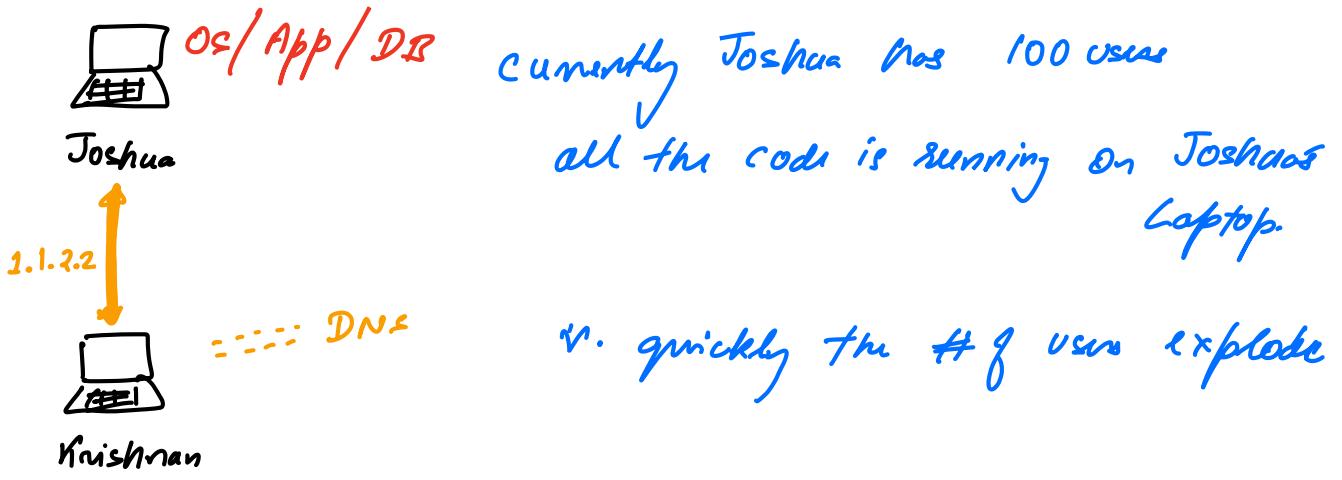
- You can override
 - at routes level
 - at OS level
 - at browser level

} You Google DNS
8.8.8.8
8.8.4.4

All HLD classes → 9 PM - 11.30 PM (± 10 mins)

First 15 mins - ^{IST} recap/assignments

doubt → 11.30 PM - 12.30 AM - no new curriculum
content will be discussed here
break - 10.30 PM



Database - MySQL to store bookmarks

User-bookmarks

Scalability

User-id	URL
1	google.com $\approx 1KB$
1	Scala.com
2	youtube.com

bigint varchar(1000)

8 bytes 1000 bytes

Q1 Space for 1 M new bookmarks each day?

$$\begin{aligned}
 & 1M \frac{\text{bookmarks}}{\text{day}} * \frac{1KB}{\text{bookmark}} = 1M + 1KB/\text{day} \\
 & = 10^6 * 10^2 \text{ bytes/day} \\
 & = 10^9 \text{ bytes/day} \\
 & = 1 GB/\text{day}
 \end{aligned}$$

Q2 How many days to run out of space?

$$\frac{40 \text{ GB}}{1 \text{ GB/day}} = 40 \text{ days}$$

(Best case)
in reality

- ① Some space is occupied by OS / code / ...
- ② # of user is growing everyday

How to handle this scale

① Vertical Scaling

\$\$
easy but unsustainable

Replace current system with a more powerful one



35000 Rs
40 GB HDD
128 MB RAM
2 cores CPU

40 days

migrated

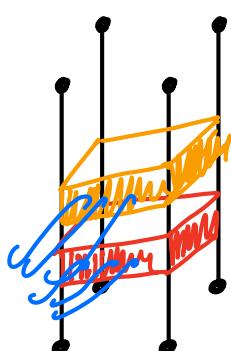
gaming



4 lakh Rs
100 GB HDD
512 MB RAM
4 cores CPU

+ 60 days

IBM
Siemens
grad
hardware



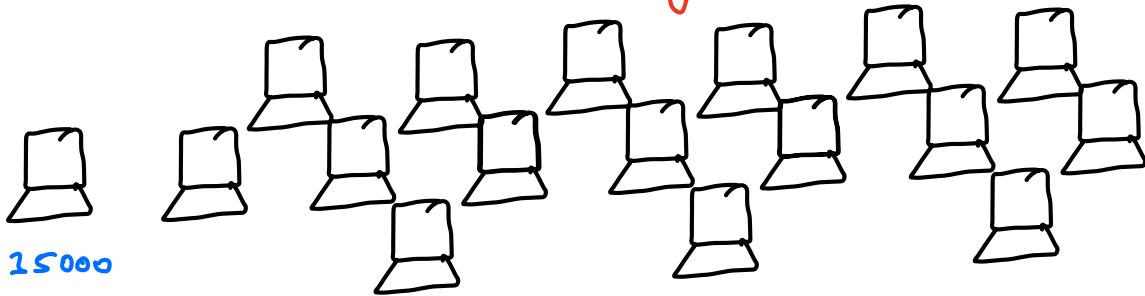
5 crore Rs
1 TB HDD
5 GB RAM
32 cores

+ 1 years

② Horizontal Scaling

v. cost effective

buy lots of cheap hardware
write logic to make them work together



15000

128 MB RAM

40 GB HDD

2 core CPU

1500 laptops

economy of scale

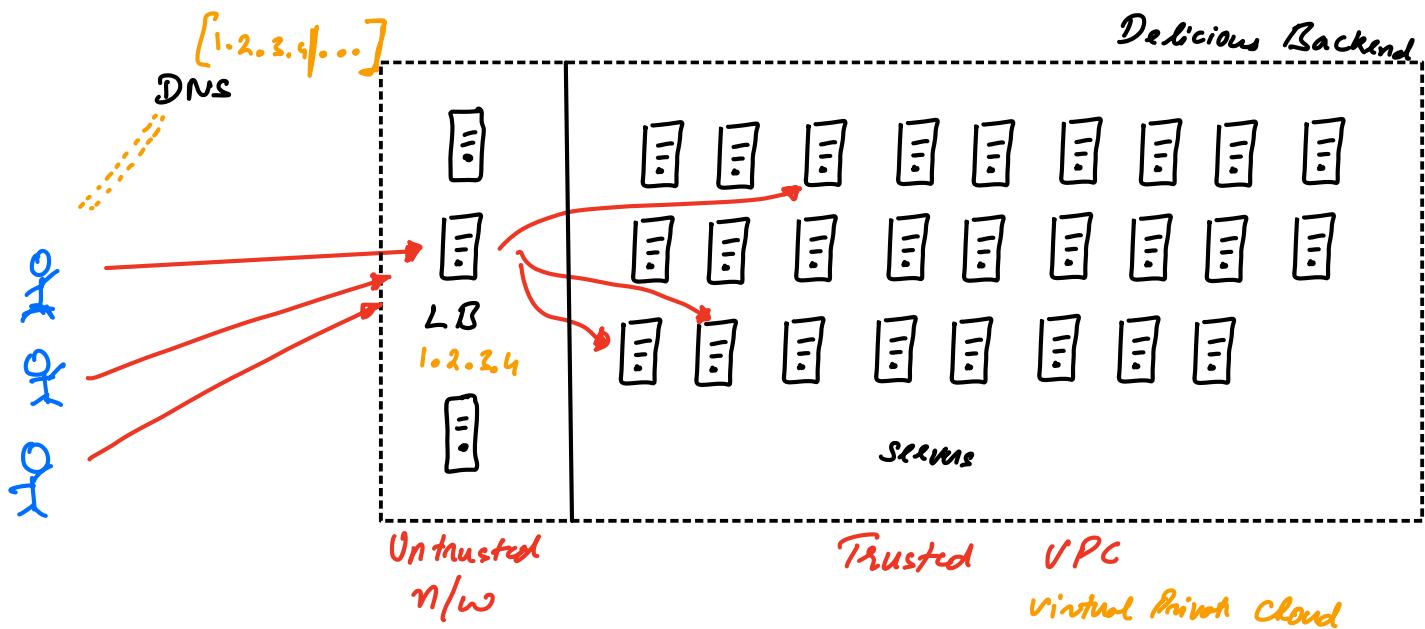
discounts :: large purchase

Horizontal scaling - dragon - pain in the ass!

HLD - how to deal with challenges that come with horizontal scaling.

10:33 → 10:45

Load Balancer



Q: which IP to register in DNS?

A: designate one machine as LB & give it a special task

Q what does LB do?

A the only job is to forward the requests

- ① load remains (almost) equally distributed
- ② user gets a unified view

Q isn't the LB now the bottleneck?

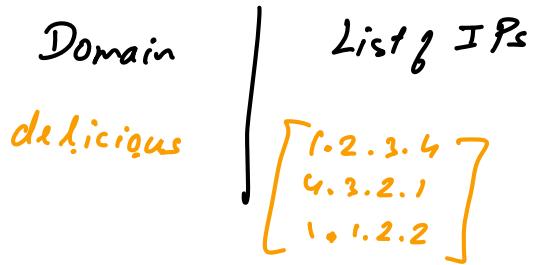
App Server - deserialize - db queries - ORM - processing
(joins)
~1000 req/sec - auth - sorting & pagination - logs - db inserts
- generate response - serialize

Load Balancer - request - headers
- decide which server to forward it to
100,000+ req/sec
easily
- forwards it

Routing Algo

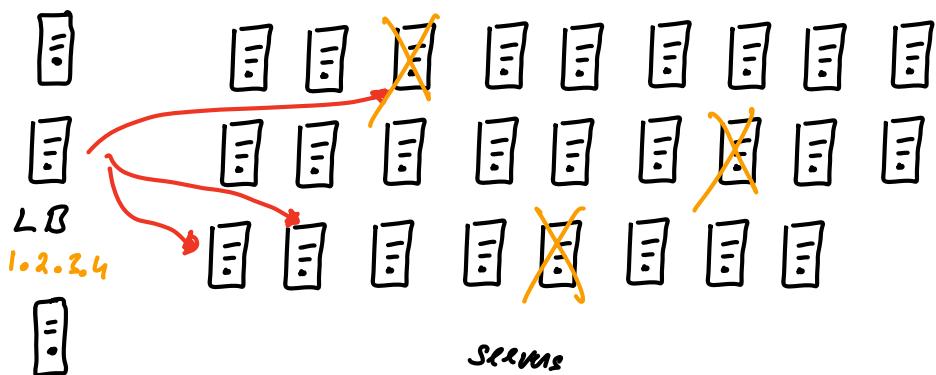
- Q what if the load is much higher? 10M+ queries/sec
- Q isn't LB a Single point of failure (SPoF)

- Have multiple LBs
- DNS stores the IPs of all these LBs



- On DNS query the DNS returns
 - GeoDNS - the nearest LB
 - normal - all the IPs - client chooses at random

Q How does LB know which servers are alive/available?

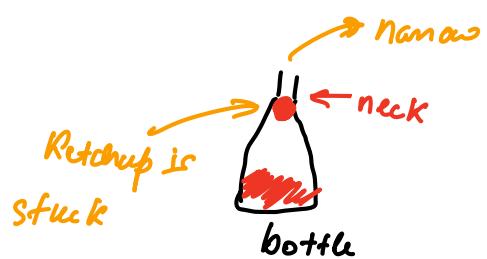


if a server is down LB should not forward any traffic

① Heartbeat - push from servers to LB

if heartbeat is missed consecutively then assume dead

② Health check - pull from LB



Q What is the routing algo?

How to decide which request to send to which server?

~~Round Robin?~~

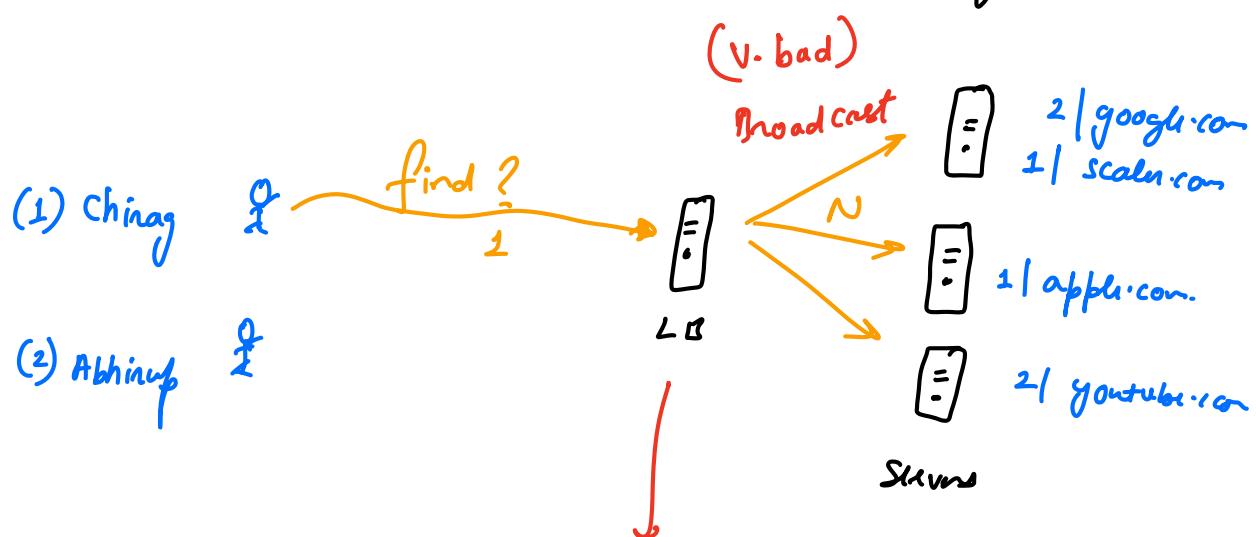
Routing Algo → how the data is being stored?
depends on

Sharding → Horizontally partitioning data across multiple servers

Can we store all bookmarks in 1 server? No

∴ we must distribute the bookmarks - how?

Q Can we distribute bookmark randomly?



"no idea about
what server has what data"

We should distribute
data in logical way

So that we are able to
retrieve it later

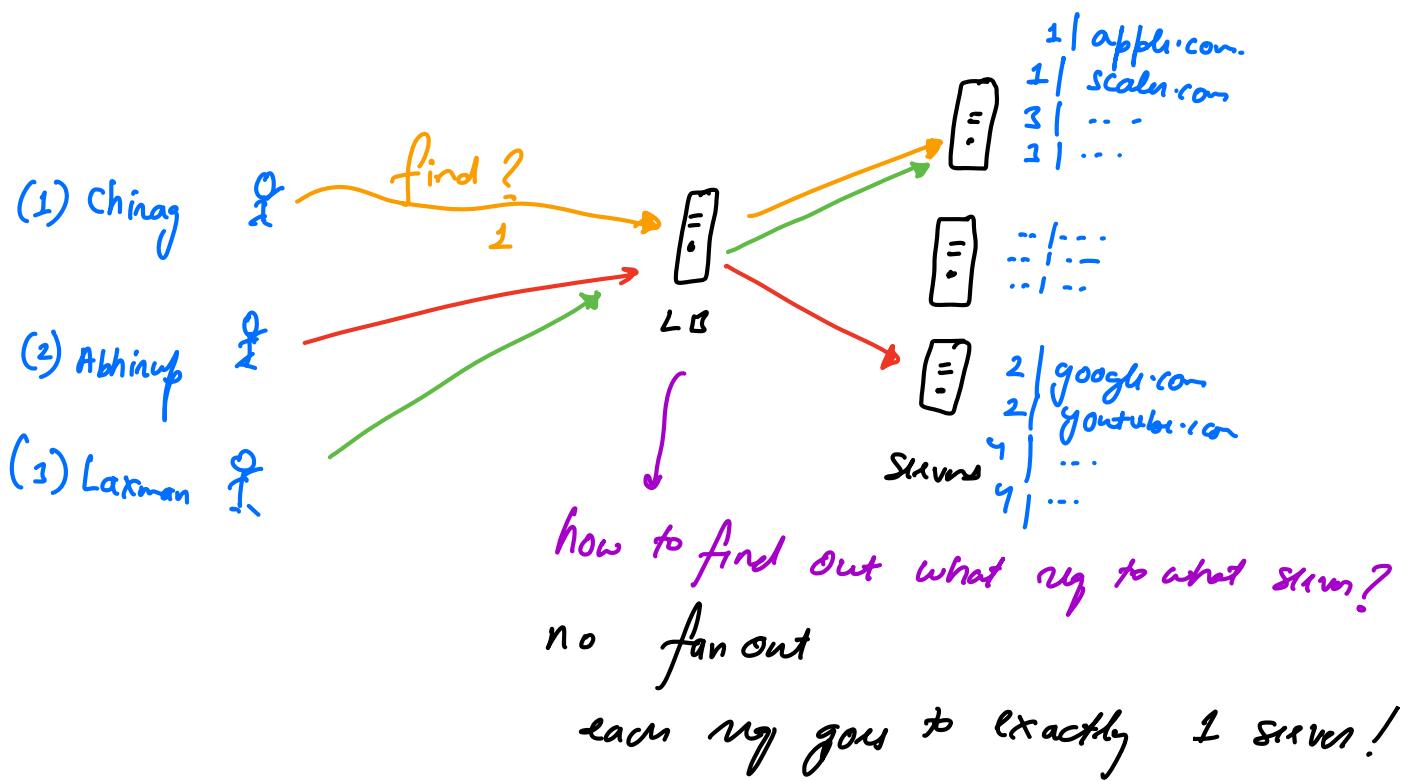
Fan out read / write



Japanese fan

data is partitioned / chorded by **Sharding Key**

for delicious - sharding key — **User-id**



Routing Algo / Mapping of user-id to serv-id

which user's data is in which serv.

next class

< 2 years of experience
(SDE 1)

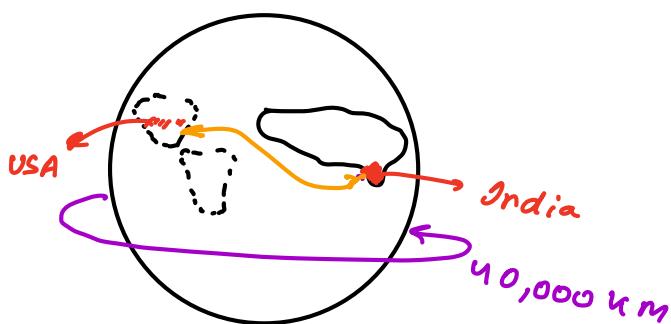
DSA / SBU /
Concurrency / LCD / Projects

> 2 years experience
(SDE 2+)

↓
+ LCD

Google DNS (USA)
d.d.-&-s

○
(India)



Round Trip from India to US
↳ 40,000 km

Round Trip Time (RTT)
from US to India?

$$C = 299792458 \text{ m/s}$$

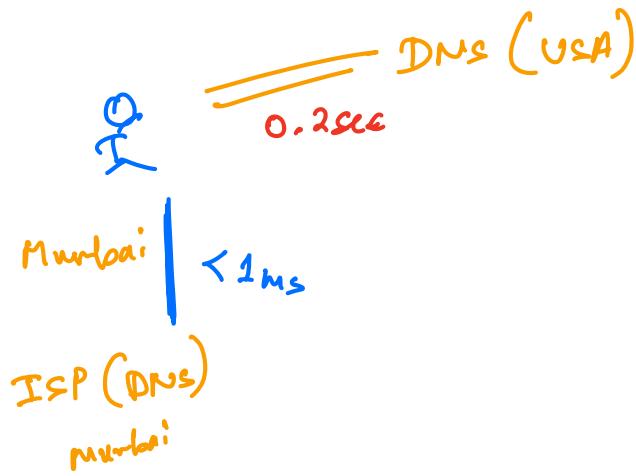
vacuum $\approx 3 \times 10^8 \text{ m/s}$

speed of light in fiber optic cables
 $\approx 2 \times 10^8 \text{ m/s}$

$$\text{time} = \frac{\text{distance}}{\text{velocity}}$$

$$= \frac{90000 \text{ km}}{2 \times 10^8 \text{ m/s}} = \frac{90000 \times 10^3 \text{ m}}{2 \times 10^8 \text{ m/s}}$$

website



As of March 2024

Typical server config

RAM	2GB - 22GB
HDD/SSD	500GB - 8TB
CPU	1 core - 16 cores
n/w	100Mbps - 10Gb/s

Max config that
money can buy

RAM	12 TB
SSD	5 PB
CPU	300 cores (600 threads)
n/w	10 Tb/s

AI/M2 ChatGPT (14ms)

Stock exchange	NYSE/NSE
Video processing	Render Avatar
Weather sim	Meteoris
Particle Coll.	LHC
Hard Tech	Fusion (Helion)