

- Data Structures & Algorithms
- Low Level Design - how to design code structures so that they scale w.r.t.
 - time
 - team size
- High Level Design - scalability of infrastructure
 $1000 \text{ users} \rightarrow 5 \text{ Billion users}$

Staff Engineer @ Google

files containing strings (query strings)

how do I drink water without spilling?

why is the sky blue?

how do I ask my crush out?

!

task: sort them alphabetically in Asc

$50PB = 50,000,000 GB$ of data.

⇒ 50 Petabytes of data

1 bit	- 10^0
4 bits - nibble	- 10^4
8 bits - byte	- 10^8
Kb - 10^3	
Mb - 10^6	
Gb - 10^9	
Tb - 10^{12}	
Pb - 10^{15}	

~~Read~~
~~Add~~
datacenter?

multiple machines

- ↳ faulty
- ↳ lost power
- ↳ lost n/w
- ↳ heterogeneity

↳ located apart geographically
 ↳ vast distances
 ↳ high latency $C = 299,792,458 \text{ m/s}$
 $\text{Signal} \approx 2 \times 10^8 \text{ m/s}$ $\leftarrow 43 \times 10^8 \text{ m/s}$



$$\begin{aligned}
 t &= \frac{d}{s} = \frac{40000 \text{ km}}{2 \times 10^8 \text{ m/s}} \\
 &= \frac{4 \times 10^4 \times 10^3 \text{ m}}{2 \times 10^8 \text{ m/s}} \\
 &= 0.2 \text{ sec} \\
 &= 200 \text{ ms}
 \end{aligned}$$

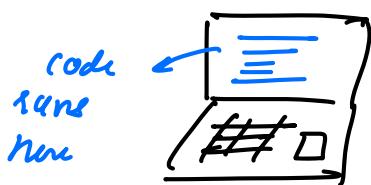
Simple problem $\xrightarrow{\text{Scale}}$ extremely difficult

Case Study - Delicious - bookmarking website 2003
 Joshua ~ bookmarks on stand on cloud.

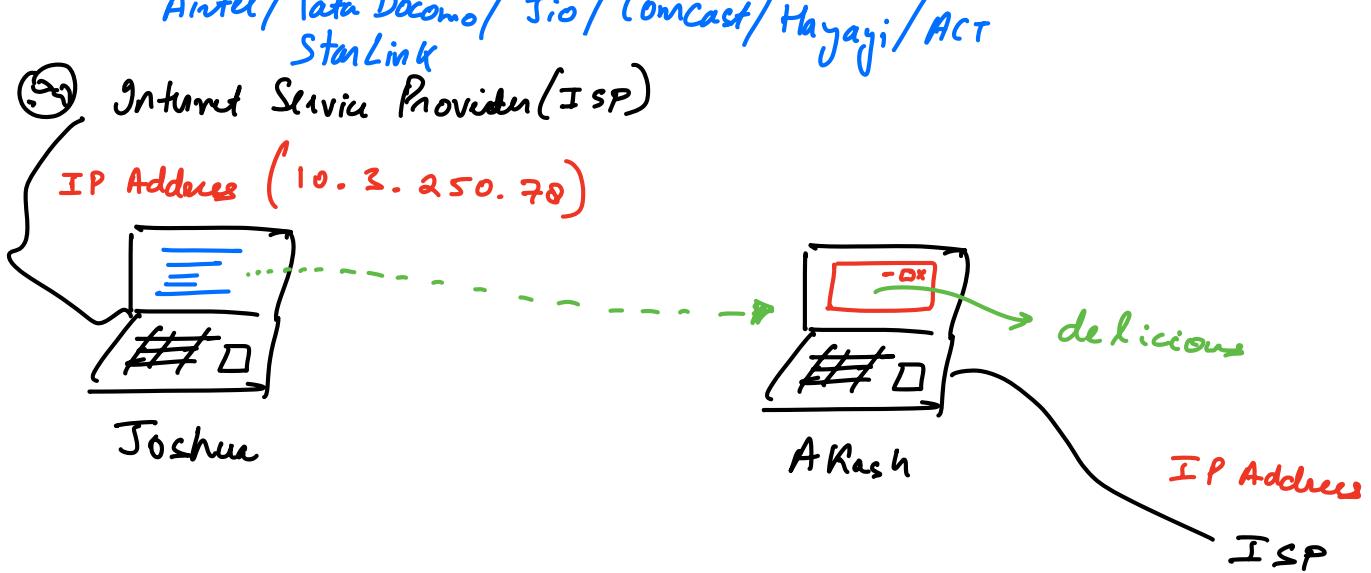
- MVP - Minimal Viable Product
 no fancy features it demonstrates something
 - Auth flows → Authentication → Authorization can
 register / login extend
 • add Bookmarks (user-id, URL)

- Functional Requirements
 explicitly write code / APIs
 authentication
 give SRK, so please give me the nucleus lands code
 authorization

- show AllBookmarks (user-id) → list of URLs



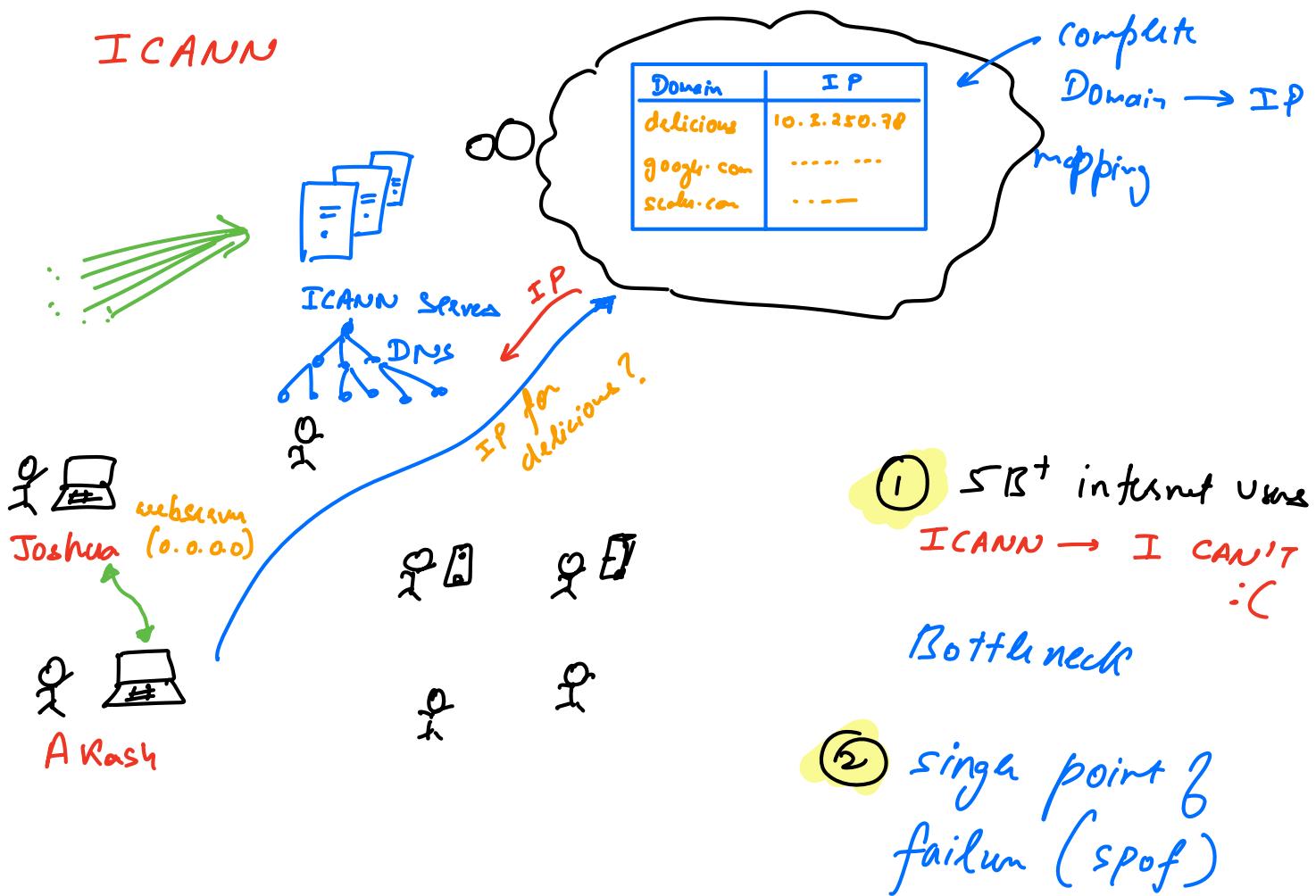
spring + Java / Python + Django / C++ Dragon } etc
MySQL



- ① Akash can remember the IP address of delicious & type it in Browser.
- ② Akash wants to simply type in the 'domain name'
<http://del.icio.us>

1st person to land on moon
— Neil Armstrong
2nd person — Buzz Aldrin
you can 'recognize' but you can't recall

Domain Name Service (DNS)



localhost = 127.0.0.1 → not accessible over n/w

0.0.0.0 → accessible to other systems in n/w

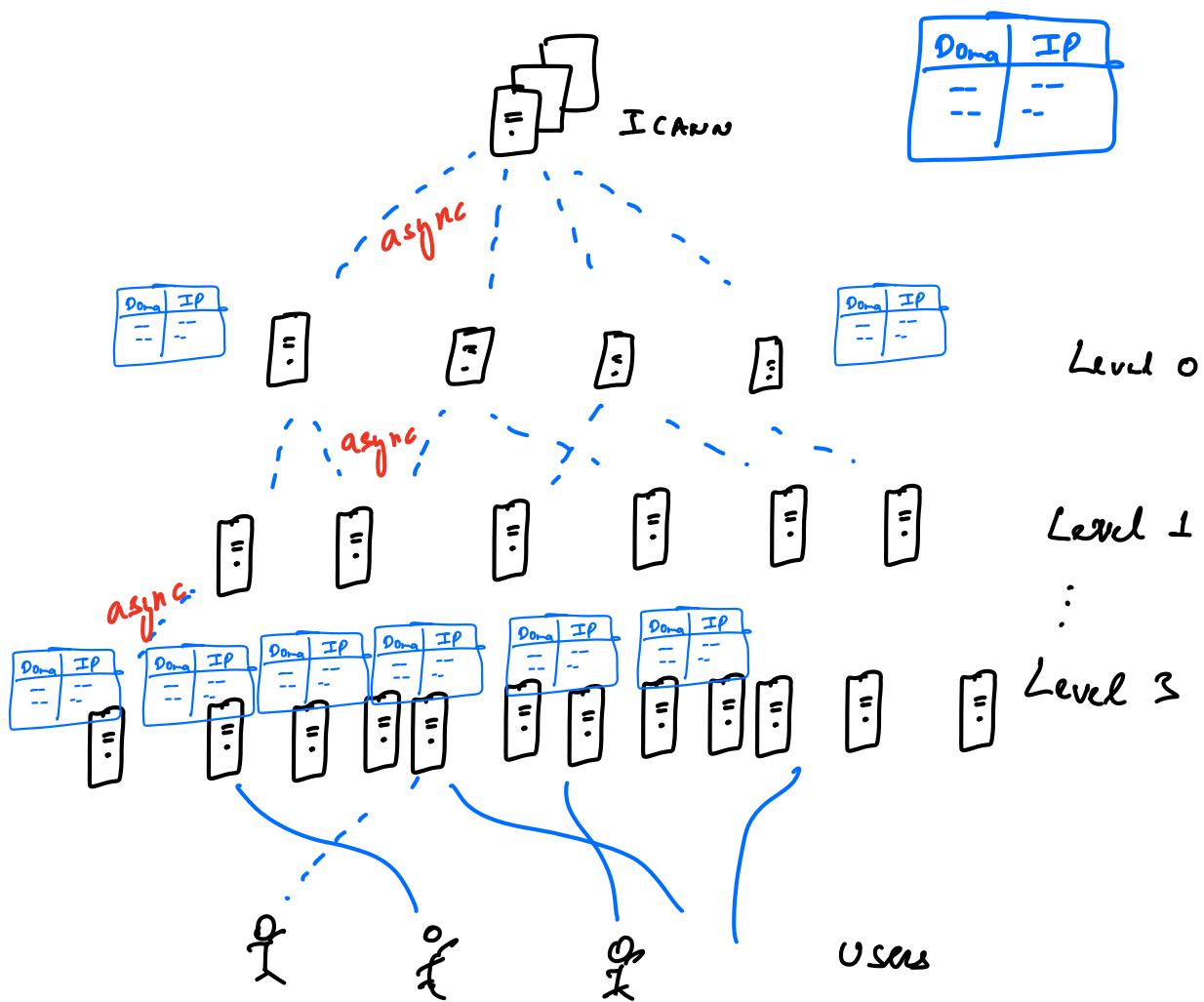
DNS architecture

ICANN = top authority

Domain Registrars - godaddy / namecheap / Route 53 ...
Purchase

DNS servers - independent servers which resolve DNS queries

ISPs / Internet giants - Search engines / Militaries / Govts / Edu /
Social media / Research Labs



How does Laptop know DNS server's IP?

- ISP → signal to router - configures DNS
- Can override
 - Router
 - OS

browsing.

Joshua — friends — colleague — city ... viral

1 10 1000 10,000 1M⁺ users

2007



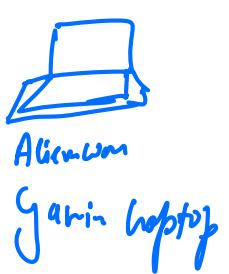
40 days
Joshua's
web server / DB
OS / ... / movies

35000 Rs
128 MB RAM
40 GB HDD
2 com CPU
8 Kbps n/w

<u>User-bookmarks</u>	
user-id	URL
100000	100000 varchar(1000)

1Kb ← bijint

1M new bookmarks
each day



5 Lakh Rs
512 MB RAM
80 GB HDD
2 com CPU
16 Kbps

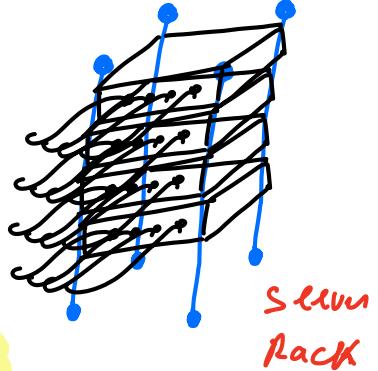
1M ~~bookmarks~~ * $\frac{1\text{Kb}}{\text{bookmark}}$

$$= 10^6 * 10^3 \frac{\text{bytes}}{\text{day}}$$

$$= 1GB/\text{day}$$



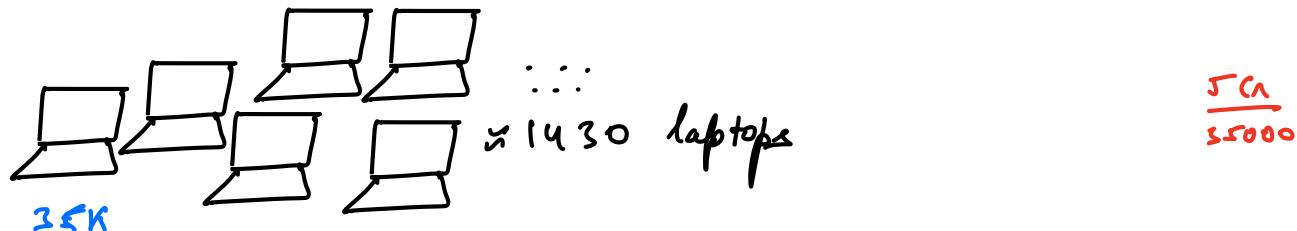
5 Crore Rs
2 GB RAM
1 TB HDD
1 com CPU
1Mbps



Vertical
Scaling

Replacing old system with a more powerful one

Horizontal scaling — buy lots of cheap machines & hook 'em up!



effective spec

RAM	180 GB
HDD	55 TB
Cores	2800 cores

economy of scale
← even cheaper than this!

Vertical scaling

- v. simple
- limit \leftarrow budget technology of current day

• Compute intensive

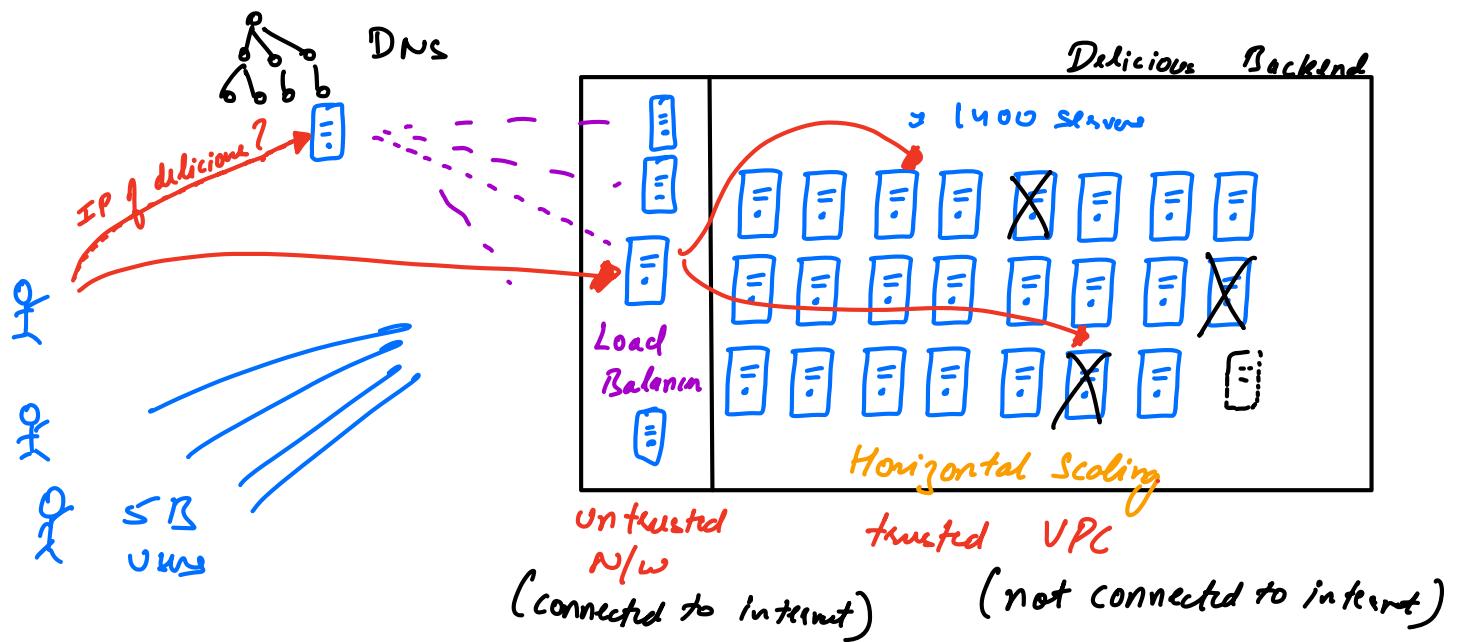
- Audio/video processing
- Stock exchange
- AI/ML

Chat GPT ≈ 1.7 T
Params (32 bit)

this model is ≈ 5.6 TB
in size

Horizontal scaling

- v. v. difficult ? entire field
- no scaling



① which server's IP should DNS provide?

- designates one server as Load Balancer (LB)
- provides its IP

② what is Load Balancer?

a server that forwards incoming requests to other servers to ensure an even load distribution

↳ # of requests/sec (RPS)
↳ amount of data

③ why do we need LB?

∴ we are doing horizontal scaling.

④ How does LB decide which request to send where?

Routing algorithm?

⑤ How does LB keep track of which machines are running?

Heartbeat / Healthcheck.

Push from Server to LB Pull from LB to Server

⑥ Doesn't LB become a bottleneck?

App server (1000 req/sec)

- ① req
 - ② decode (JSON - deserialize)
 - ③ Auth
 - ④ DB queries
 - ⑤ ORM
 - ⑥ Slice & dice
 - ⑦ agg
 - ⑧ encode
 - ⑨ send
- } expensive & slow

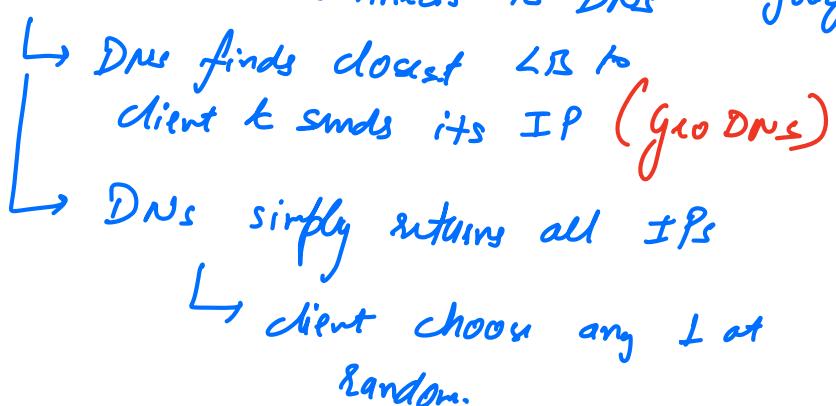
LB ($200,000 + \text{req/sec}$)

- ① req
- ② headers
- ③ where to send it?
- ④ sends it

Still a single LB isn't enough to handle Google load!

⑦ LB is a SPOF?

- Yes it is
- therefore we will use multiple LBs
- which IP is registered?
all LB's
- now when client connects to DNS

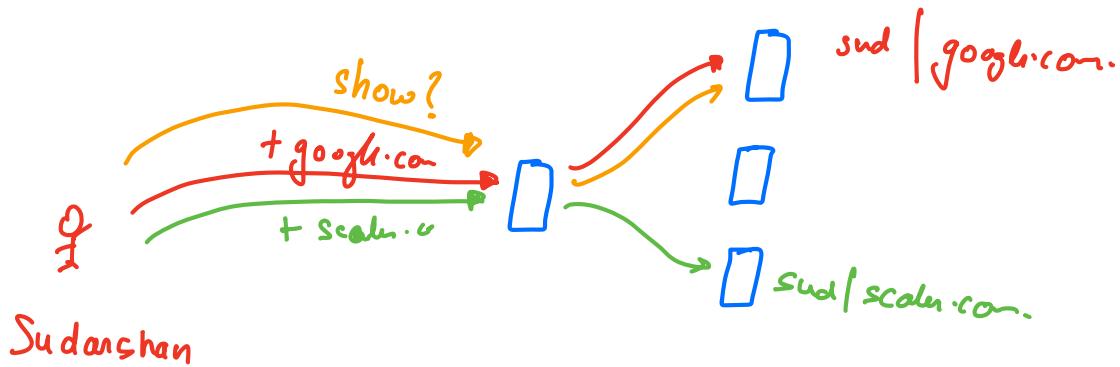


IP address(es)
[≡]

Routing Algorithm?

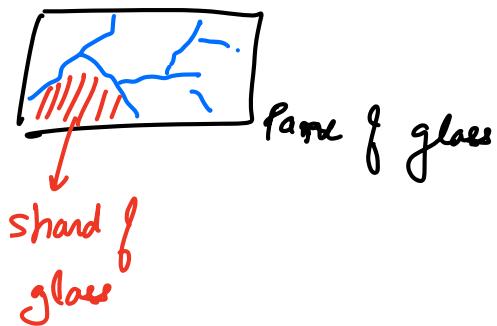
Consistent Hashing

which ring to which server.
this depends on where & how the
data is stored?



Sharding

splitting data across machines
horizontal partitioning across servers



Replica → always for data