Agenda

1. What & Why of HLD — High Level Design
2. Case Study - del.icio.us
3. Scaling challenges
4. HLD curriculum overview

support@scaler.com

Pragy → SSE + Instructor @ Scaler.

Media.Net - 2nd largest Ad Tech. - Platform Engineer.
Predict what ads to show on what Pages.
1 Billion new row / day.
Past 30 days of data
≈ 30 B rows of data (HBase)
ML E / Bandit Algos
↳ Reinforcement Learning.
↳ Machine Learning

Gotocv/flow → web dev.
30K+ new users each year

UG → Batch - 3 - MIET
PG → Batch-1 - IIIT B

Programmers → write Code.

↓ transition

Solving Problems.

Data Structures & Algo efficient.

Making sure that our problems stay solved

Low Level Design

Senior positions → Scaling Challenges.

Staff Engineer @ Google → 1+ Cr /annum

question during the design round.

Q: "given a list of strings, sort them alphabetically"

[Sneha,
Ganesh,
Saran,
Himanshu,
⋮
]

→

[Ganesh,
Himanshu,
Saran,
Sneha]

Python : names = [...]
names.sort()

Java : List<String> names;
Collections.sort(names);

Q: "given a list of strings, sort them alphabetically"

Catch: There is 50 PB of data

1 bit ⇒ 0/1

1 byte ⇒ 8 bits

1 Kilo byte ⇒ 1000 bytes = 10^3 bytes

↳ SI prefix = 1000

1 KB

1 Kibbi Byte = 1024 bytes

1 KiB

Mibbi Byte / Gibbi Byte

1 Mega byte = 10^6 bytes

1 Giga byte = 10^9 bytes

1 Tera byte = 10^{12} bytes

1 Peta byte = 10^{15} bytes

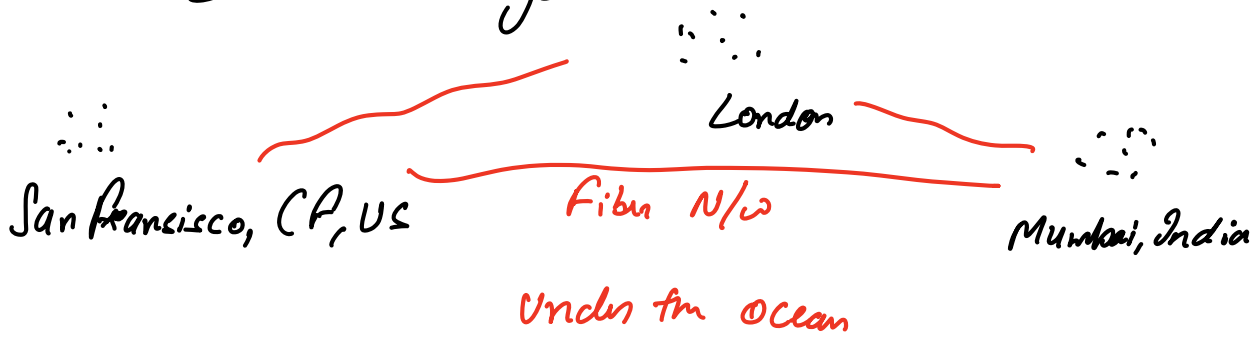
50 PB = 50,000,000 GB

50PB fits in 1 system? No!

avg laptop has 1 TB Harddisk.

you need 50,000 laptops to just store this data.

Massive Distributed System.



Despite the issues.

task needs to be completed

correct / efficient / durable

1. N/w can go down
2. Machines can fail
3. Data can get corrupted
4. Natural disasters
5. Network latency

↳ amount of time taken for something to happen.

'delay'

Map - Reduce

Simple problems ⟶ Scale

v. v. difficult

What challenges appear at scale, & how to tackle them.

Pre-requisites

1. Writing Code
2. Data Structure & Algo
3. Computer Fundamentals
   - Networks
   - Databases (SQL)
   - Operating Systems
4. Low Level Design

Mastered

90% understanding of the concepts discussed today → ok ✓

Del.icio.us Case Study

Joshua

↳ bookmarking Service in the cloud.

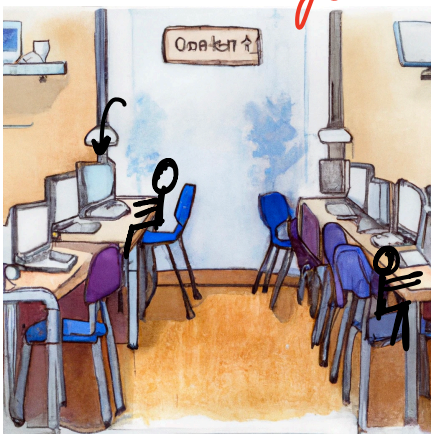
? Launched in

2002 — 2005 — 2006 — 2008

(2Kbps speed)
Dial-up
↓
Broad Band
↓
Fiber 1 GBps

Motivation

Internet Cafe



google.com / scaler.com

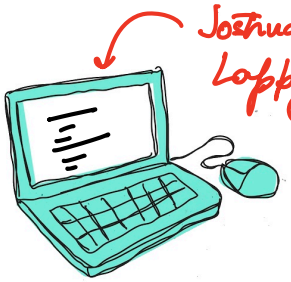
Bookmark them

→ no bookmarks in other PC

Features

Minimal Viable Product (MVP) – Proof of Concept

1. Auth – user registration & login
2. Add Bookmark — add Bookmark (user-id, URL)
3. Fetch Bookmarks — get My Bookmarks (user-id)



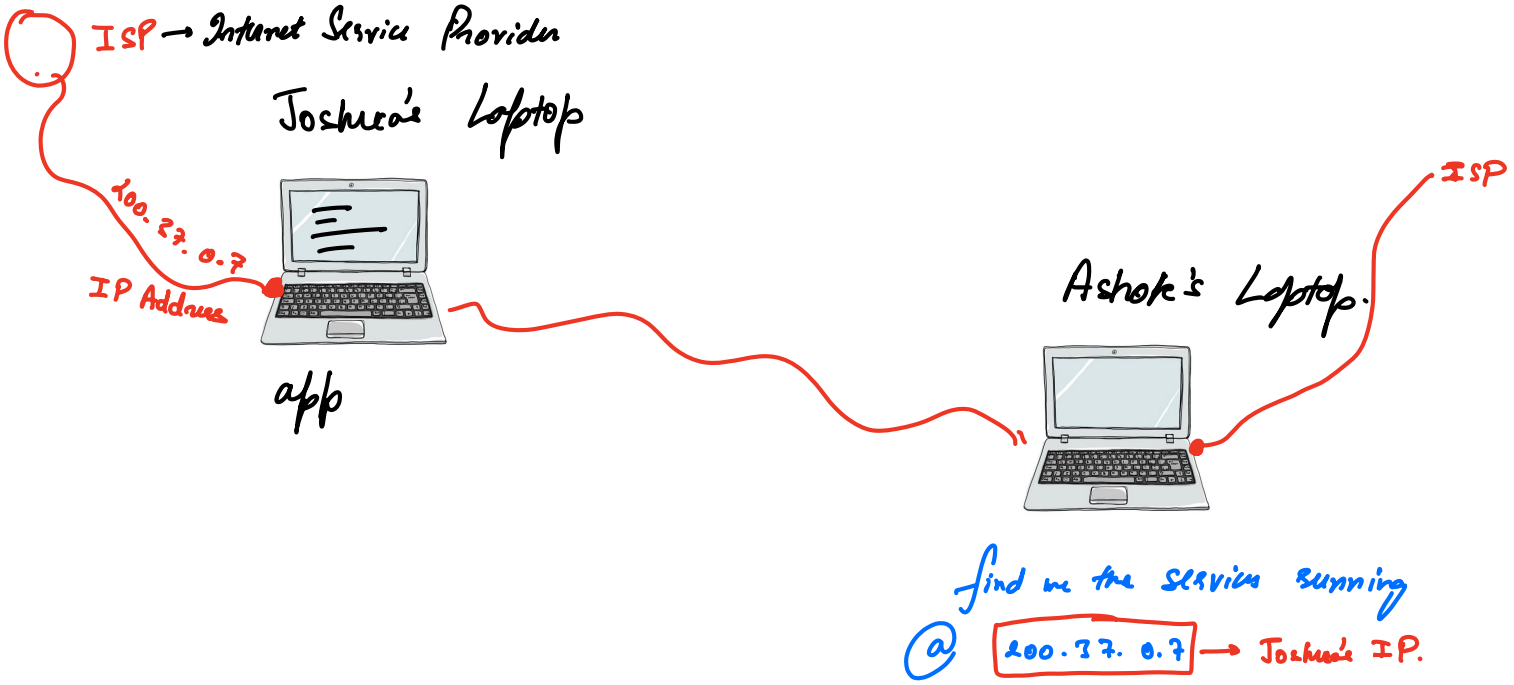
Joshua's Laptop.

Python + Django + MySQL
Web server App

127.0.0.1:8080/

to be continued..

Fantastic websites & where to find them



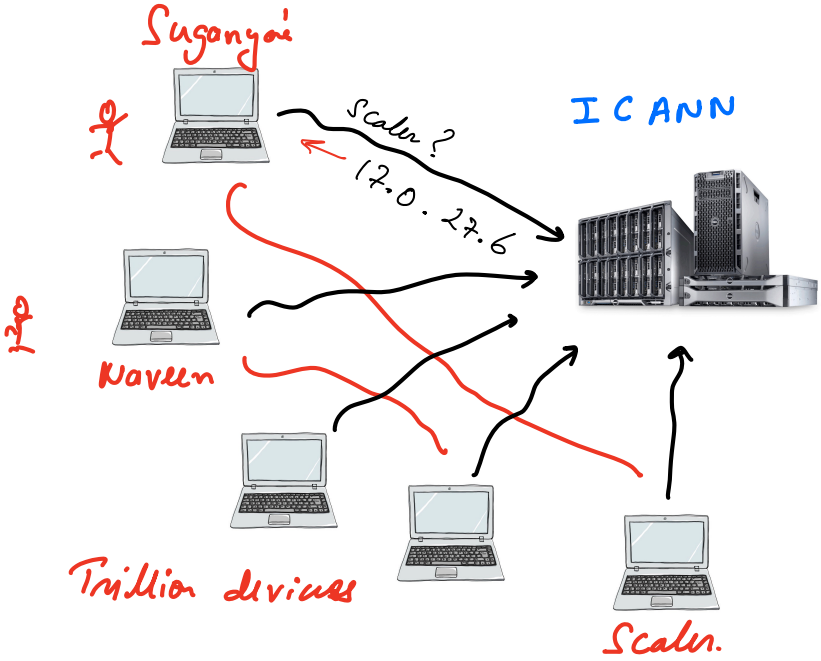
? Quiz - how does the browser know?

DNS Service

http://del.icio.us

Domain Name Mapping

Domain Name	IP Address
scaler.com	17.0.27.6
google.com	200.70.20.10
delicious	12.10.1.2
..	



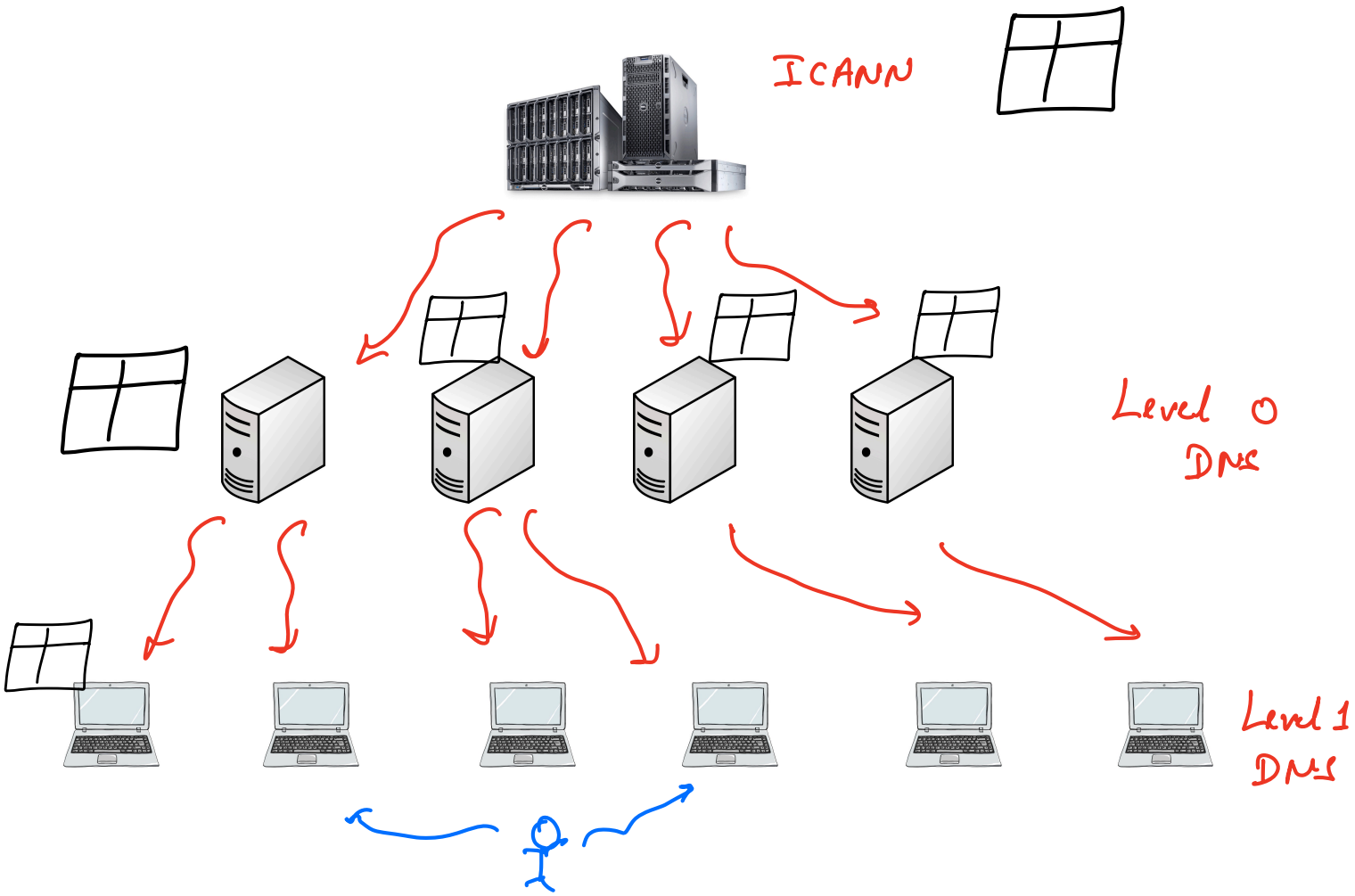
ICANN → I CAN'T !!

Single Point of failure SPOF

⚠ Bottleneck.

Domain Name Service

https://www.cloudflare.com/en-gb/learning/dns/what-is-dns/



Who maintains the DNS?

Internet Companies → Google / Microsoft / --

ISPs → Airtel / Jio / Comcast

Educational Institutes → Stanford / Harvard

Militaries

Anyone can create a DNS Server.

Demos

Del.icio.us continued ..

2006 → 25,000 Rs
Intel Pentium Core 2 duo → 2 cores @ 2.2 GHz
120 MB of RAM
40 GB of HDD

Joshua's System

System Configuration

RAM	120 Mb
Disk	40 GB
CPU	2 core

} 25,000 Rs

App — Django + Python
DB — MySQL
OS — fedora

Issues → if Joshua is running the website on his personal machine.

1. Down time — sleep / powercut / internet issue
2. Content for users b/w website & personal usage

Dedicated Machine to serve the website → Server.

↓

machine connected to internet with the sole purpose of providing some service

Quiz - space

user_bookmarks

user_id	URL
1	google.com
2	scaler.com
1	youtube.com
2	microsoft.com
8 bytes	1000 bytes

Size of each row is 1 KB

1 million new bookmarks each day.

1 bit ⇒ 0 OR 1 fundamental unit of information.

1 byte ⇒ 8 bits

1 KB ⇒ 10^3 bytes

1 MB ⇒ 10^6 bytes

1 GB → 10^9 bytes

Every bookmark ⟶ row in the database

Size of 1 row = 1 KB = 10^3 bytes / bookmark

New Bookmarks each day = 1 Million
= 10^6 bookmarks / day.

How much data are we getting / day?

= 10^6 bookmarks / day. * 10^3 bytes / bookmark

= 10^9 bytes / day. = 1 GB / day

Start with 40 GB space

Occupying 1 GB / day.

# of days for which we can survive = $\frac{40 \text{ GB}}{1 \text{ GB / day}}$ = 40 days.

? Quiz - time

10^2 Bytes/sec * 10^6 secs/day

1 GB/day.

$= 10^9$ bytes/day = 1 GB/day.

40 GB

#days = 40 GB / 1 GB/day = 40 days

10:43 → 10:55

Solutions

Buy Powerful Machines

1.

25,000 Rs
128 Mb RAM
40 GB HDD
2 core CPU

40 days

moved

Local vendor → give me the best system you have!
→ cash!

2 Lakh Rs
512 MB RAM
100 GB HDD
2 core CPU

→ 60 additional GB
60 more days

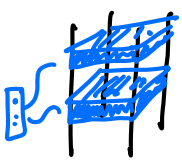
Vertical Scaling

IBM → Server grade hardware
20 lakh Rs
2 GB RAM
500 GB HDD
8 core CPU

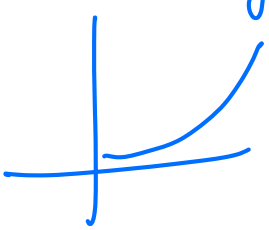
→ 400 additional days.

Server Rack
2 Crore Rs
10 GB RAM
2 TB HDD
22 core CPU

→ X 1 year
5 more years

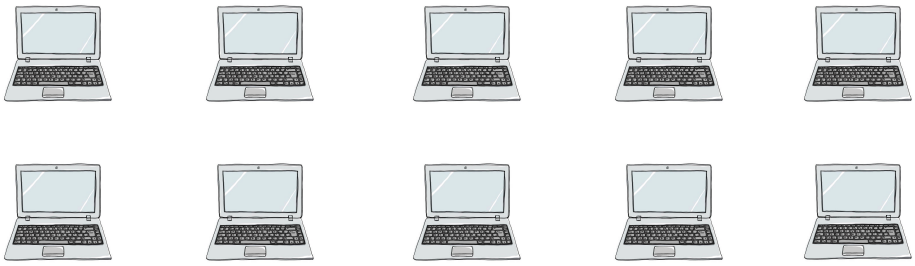


Rack



2. Buy lots of cheap machines

15,000 Rs
128 Mb RAM
40 GB HDD
2 core CPU



2 core Rs ⇒ 570

effective RAM: 72 GB
HDD: 22 TB
Cores: 1000+

80% discount!
Economy of scale

Horizontal Scaling

Vertical	Horizontal
(1) Super Easy → throw money	(1) v.v. difficult Entire HLD is about solving these difficulties.
(2) Hard Limit based on current day technology	(2) Do this infinitely!!
(3) New machine → stop using old machine	(3) Buy new machine & continue using old ones.

Eventually you will need Horizontal Scaling.

Potential of vertical Scaling nowadays.

Typical Server Config

RAM: 2 GB → 128 GB

HDD: 512 GB → 8 TB

CPU: 1 core → 8 cores

N/w: 10 Mbps → 1 Gbps

Max server config as of Sept 2023

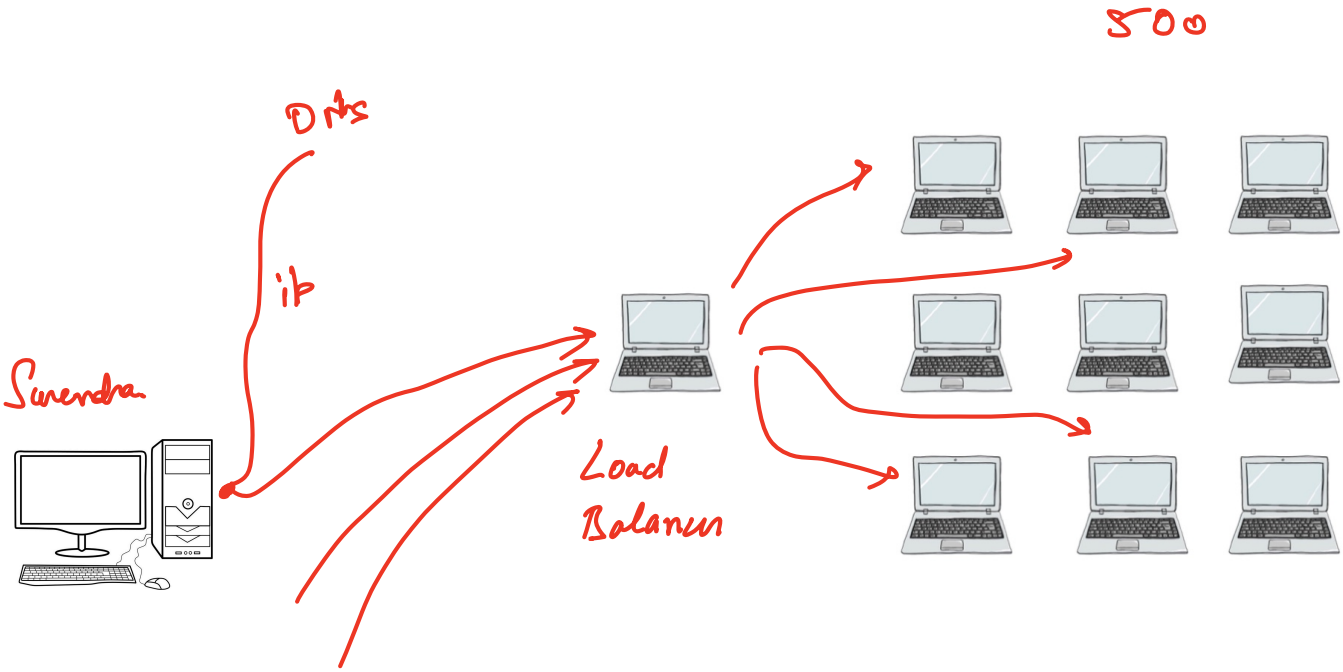
RAM: 12 TB

HDD: 2 PB

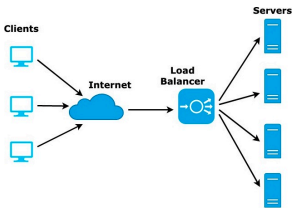
CPU: 570 cores. 570 cores

1 GB of L2 cache EPYC

Quiz - which ip?



Home work



Load Balancing

https://docs.google.com/document/d/1DxQzLpu1XPe_mRWsewNWtKL6E4uwKHQhBp7GX6Sg7qI/edit

How does the Load Balancer track which servers are up & running?

Further Challenges

? What if the Load Balancer goes down?

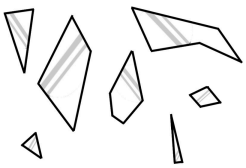
? Which machine should we send the request to?
Routing Algorithm

https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/

How to store the data?

? Can we store all data on 1 machine?

? Split randomly?



Sharding

Gangam Style → crosse billion views

crossed 2 billion views.

view count → (-ve)

32 bit signed integer → $[-2^{31} \dots 2^{31}-1]$ → 4 billion diff values

≈ [-2 Billion -- 2 Billion]

World Pop = 8 B

$2 * 2 * 2 = 8$ choices.

0/1 0/1 0/1 — 3 bits

0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

$8 = 2^3$

32 bits

2^{32} choices ≈ 4 billion

8 billion people

HLD curriculum

1.5 months

DSA
CS Funda
↳ OS
↳ DBMS
↳ CN
LLD

(1) Load Balancing & Consistent Hashing

(2) Caching → Local / global / distributed
→ Eviction
→ Invalidation → write thru / write-back / TTL

Case studies → Fb News feed cached
Code judge @ scale
LeaderBoard @ scale.

(3) CAP theorem / PACELC Theorem / Eventual Consist.

(4) Master – slave replication → Quorum
→ Tunable consist
→ Leader Election

(5) SQL vs NoSQL dbs.

ACID / BASE
Key-value Redis / DynamoDB
Doc MongoDB / _ .
Colum Cassandra / Hbase

How to choose

6. Sharding → how to shard

7. How NoSQL db work internally → LSM Trees. Bloom Filter

8. Case Study → Design Google Typeahead.

9. Case Study → FB Messenger.

10. Case Study → 2 Booking

11. Messaging Queue → Kafka

12. Case Study — Elastic Search

13. Case Study — S3 / file storage

14. Case Study — Uber / nearest neighbour → Quad Tree

15. Rate Limiter / ID gen

16. Hotstar

17. Microservices — Monolith / Event Driven / Pub Sub / SAGA / Distributed Tracing

18. Case study — IRCTC / ticket booking.

Stackoverflow → SQL db

100TB of in-mem (RAM) cache

user_id → 8 byte integer (64 bit)

auto inc ids in db.

1
2
3
4
!

predictable.

Point of attack.

1233 1234 1235

UUID / GUID

16 byte value

random.

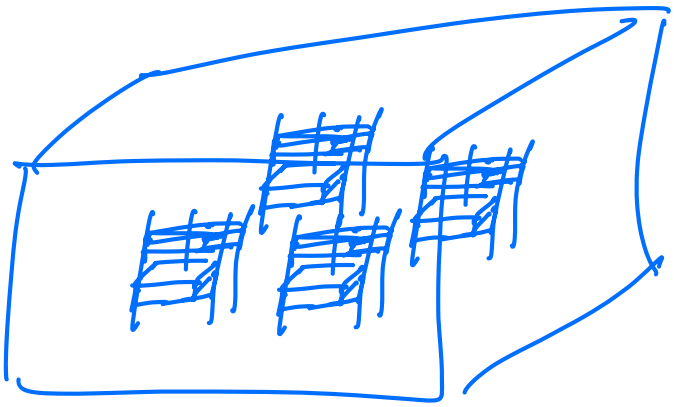
a7266 – 34a63 – aff 26255 – a – –

16 byte = 128 bit no.

$(23)_{10} = (10111)_2 = (17)_{16} = (27)_8$

Shoe Rack stor shoes

Spms rack stor spm.



Cloud — someone else's computer over the n/w.