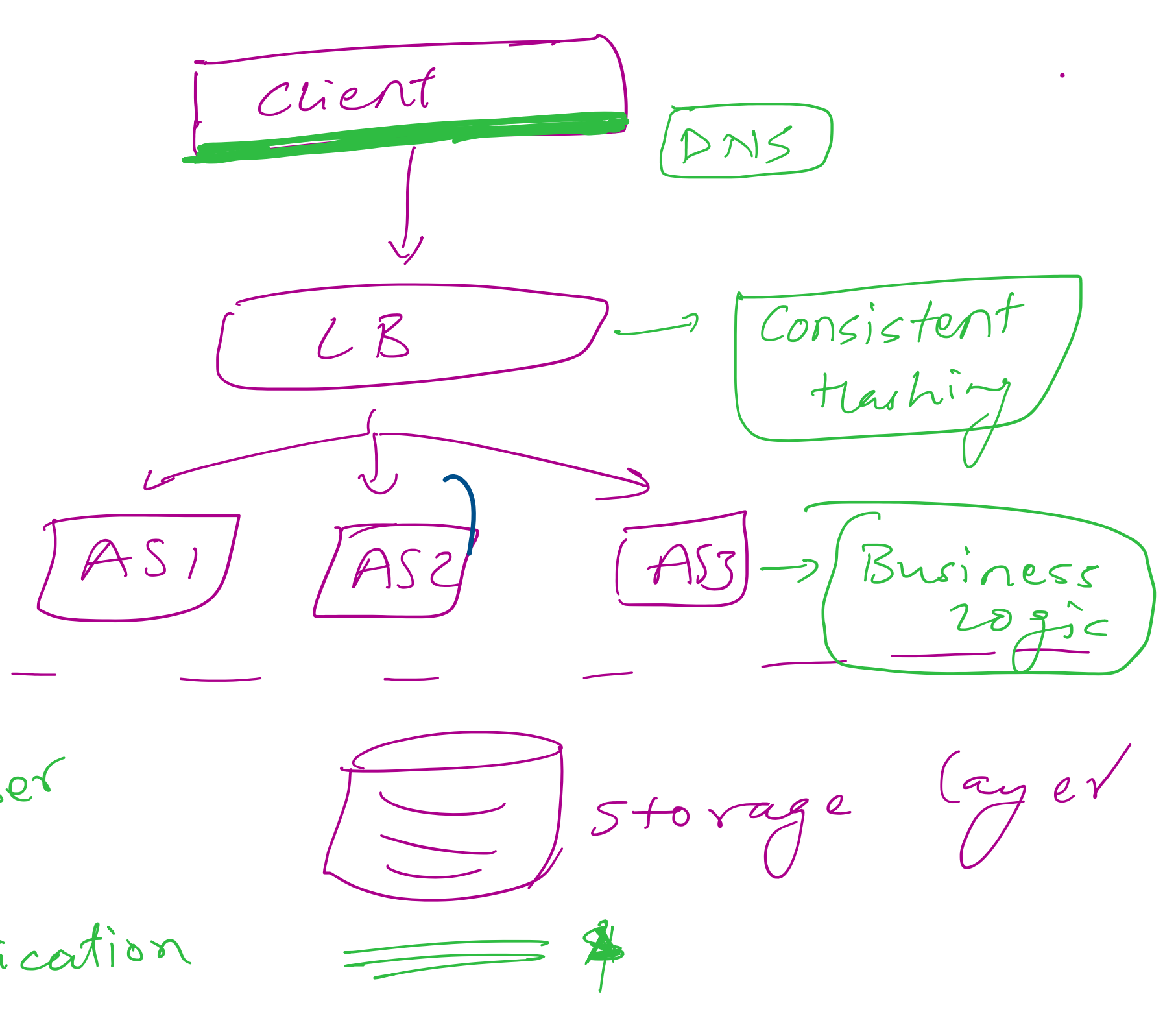
Sharding : Search Typeahead

Tuesday, 1 December 2020 5:12 PM

① SQL — Indexing
→ sharding → Devops pain

② shard unstructured data
search typeahead

③ Problem statement : Big Table

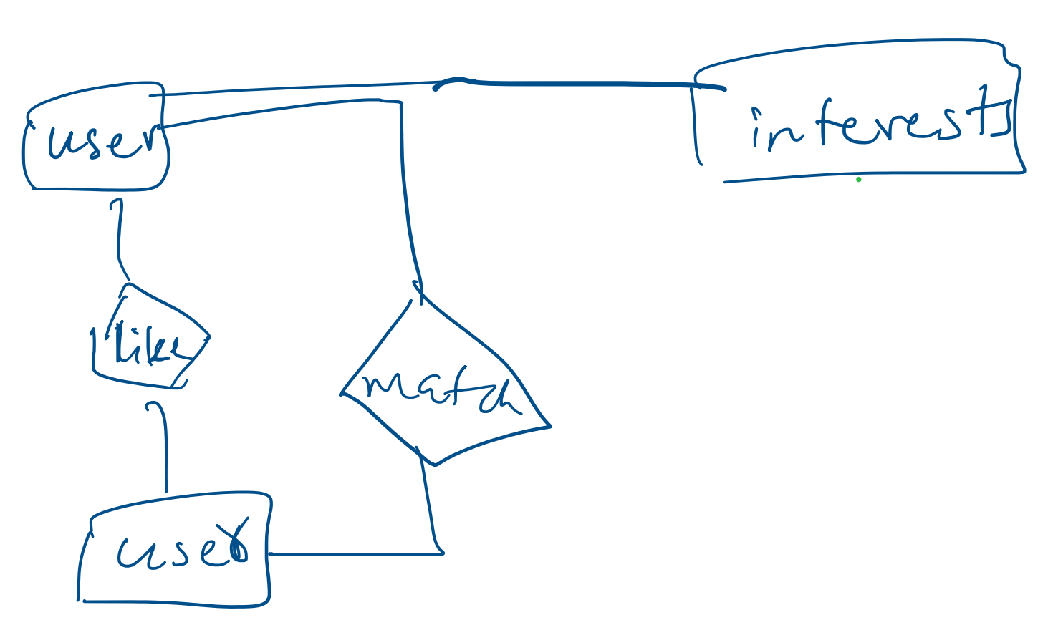


Caching
1) Browser
2) DNS
3) Application
4) Database

⊙ Traditionally, data was stored in SQL based dB (
- MySQL
- Oracle
- Postgress

⊙ Data is split into multiple tables

lets consider data of Tinder.



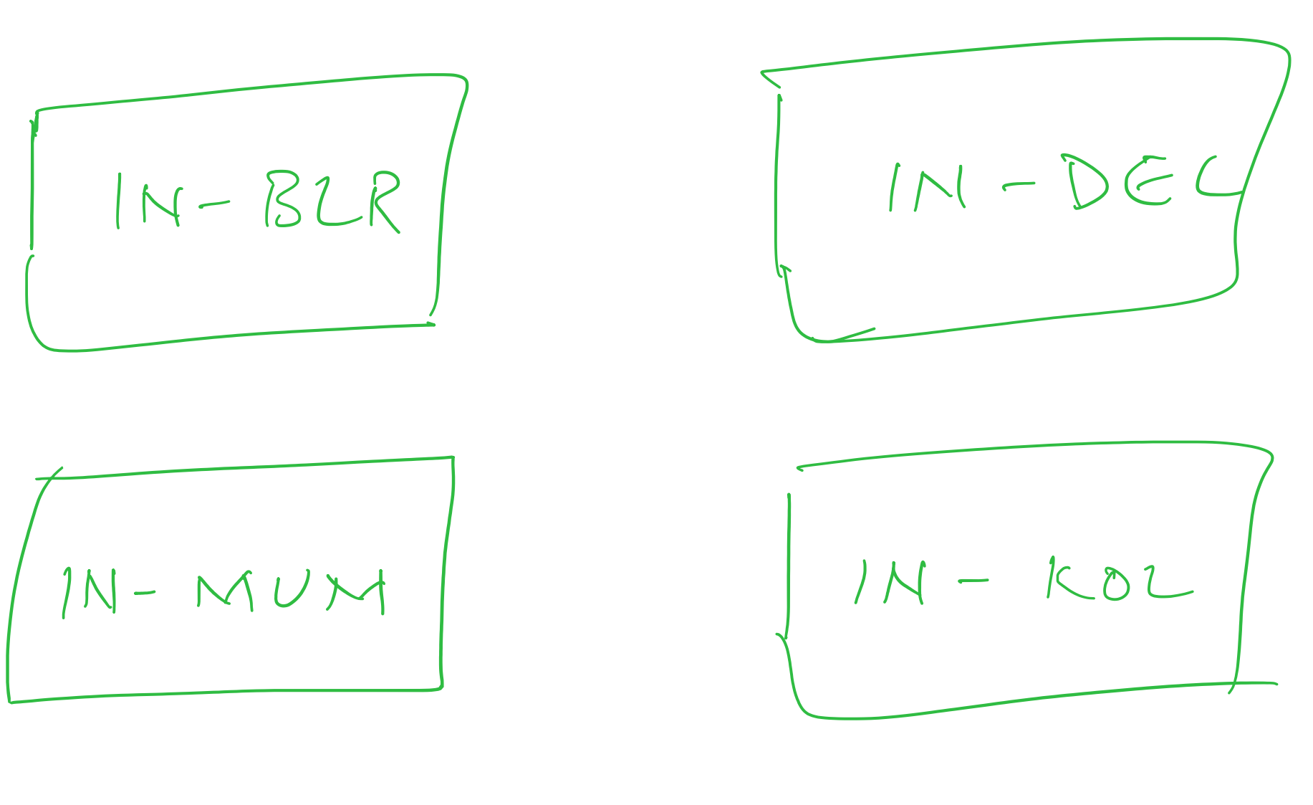
size of data depends on
a) # of users
b) interests

to meet growing needs we can
- use better machines (RAM, HDD, SDD)
- store data in multiple machines.

Tinder

⊙ shard the data on user_id

⊙ shard based on geo-hash



CONS

⊙ Manual way of adding & removing the shards.

⊙ Problem Statement

Can I create a system where I just throw in more resources and data is readjusted automatically.

Lets see another use case

Before google published a paper on Bigtable, every single company used to use multiple RDS servers and shards them manually.

⊙ Google used to scrape web URLs and had to store this dynamically increasing data.

⊙ Bigtable was the 1st DB which was truly No-SQL.

Google Search Typeahead

⊙ All typeahead suggestⁿ will have typed text as strict prefix

⊙ Suggestⁿ should be most popular search queries.

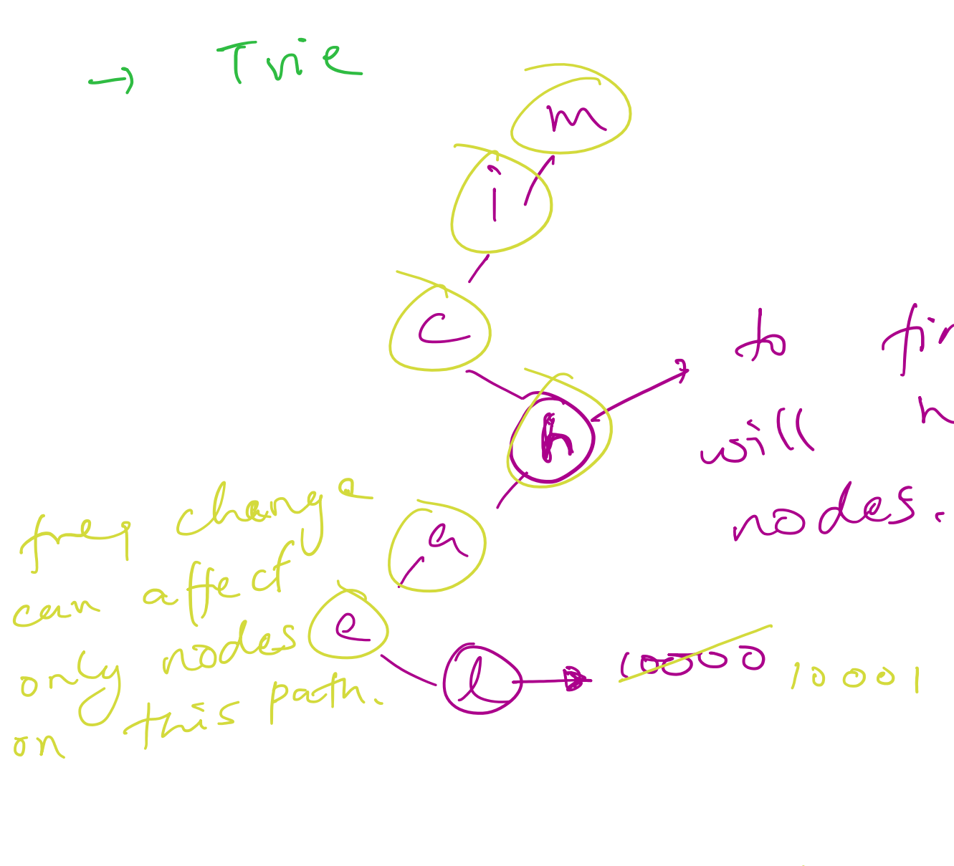
⊙ No personalisatⁿ.

→ Given 'prefix' : find top 5 popular queries with ⓧ as prefix

→ increase freq/popularity of ⓨ as search query.

On a single Machine

→ Trie



to find top 5 here we will have to go thru lot of nodes. O(n)

Node { char :
freq :
top 5 : }

→ write competes with read.

⊙ Replicate & update in background.
- miss out on nds. (update : 1 hrs)

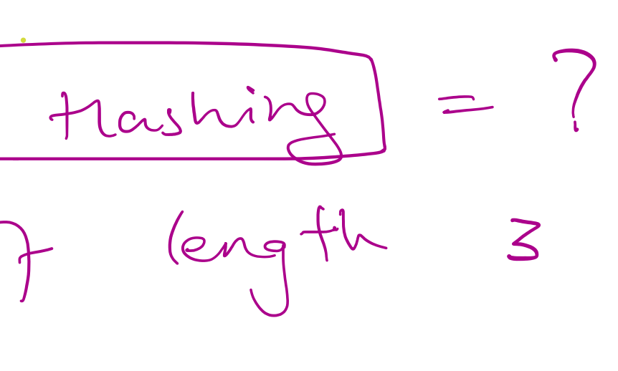
⊙ Barrier capacity as 10, threshold. (Hash Map)

⊙ Sampling

⊙ Time decay : ? f = f/d = f/1.01 (1%)
f/2 (%)

→ trie will not fit on single mc.

SHARDS = ?



⊙ Hot shards ? (A, S)

⊙ Not more than 26 shards

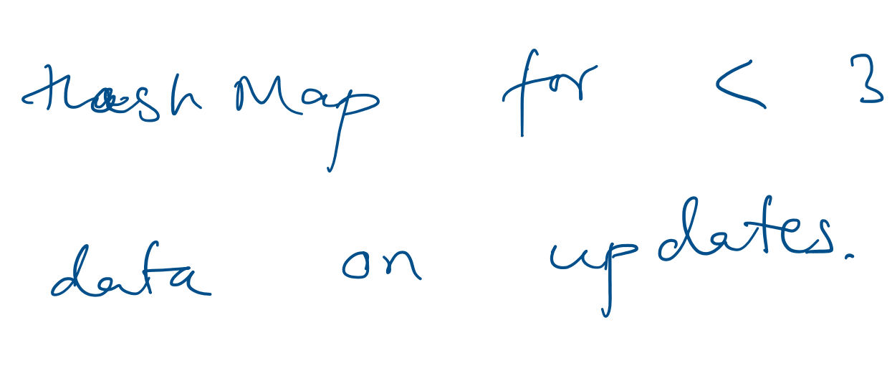
⊙ Uneven distributⁿ of data

Consistent Hashing = ?

⊙ prefixes of length 3

26 × 27 × 27

aaa aab - - - zzz



⊙ what if the search query is of length = 2.
⇒ Don't show suggestions

⊙ Global HashMap for < 3 chars

⊙ Persist data on updates. (HD)

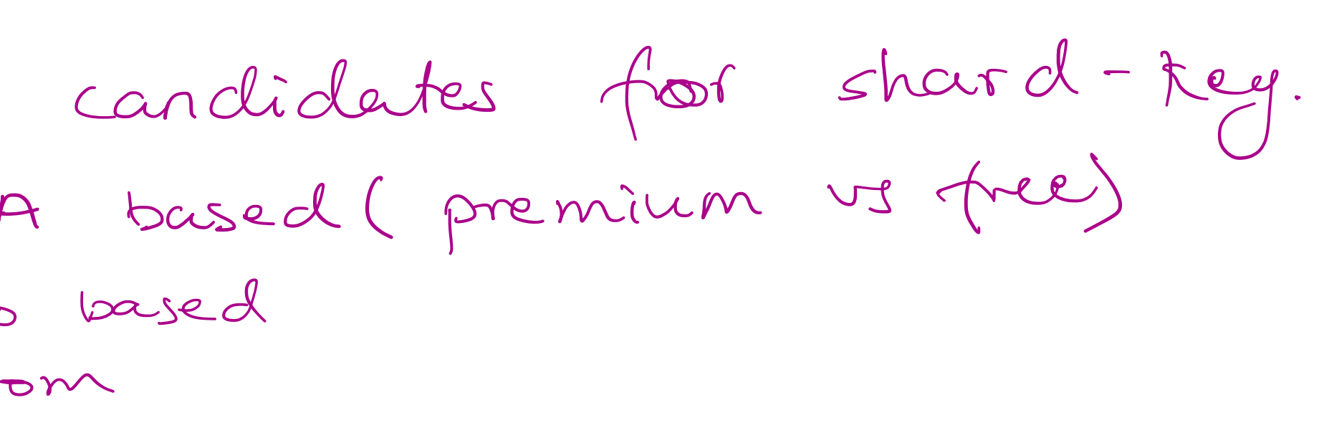
Google wanted to store such complex data structures.

x ———— x ———— x ———— x

SHARDING is spliting your data into multiple m/c so that it can be scaled horizontally.
- → Essentially distributing the data
- → concept true for any distributed system

⊙ Good candidates for shard-key.
- SLA based (premium vs free)
- Geo based
- Custom

CUBE SCALING



Scaler

⊙ Users : Learner, TA, Mentor, Instructor

⊙ Lectures : live, archived.

⊙ Assignments : DS-Algo, Design

⊙ Contest : Code Nation

⊙ Practise Problems :

How do I build No-SQL system?

① Denormalize data