

Agenda.

⇒ Design Google Typahead

mic

microsoft

microservice

microwave

—

—

—

⇒ 4 step process.

1) **MVP** : Minimum Viable Product



Minimal set of requirements.

2) Estimation of Scale.

↳ Back of the envelope calculations.

→ Is sharding required?

→ Read Heavy System :  $x \gg y$

Write Heavy System :  $y \gg x$

Both Read & Write Heavy System :  $x \approx y$

# of read queries =  $x$

# of write queries =  $y$

### 3) Design Trade Offs | Design Goals.

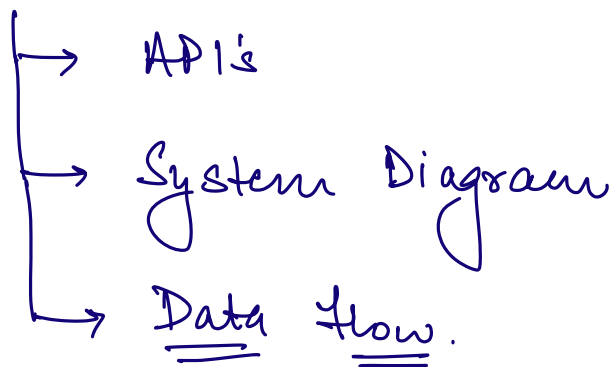
I) High Consistency (vs) High Availability.

II) Latency

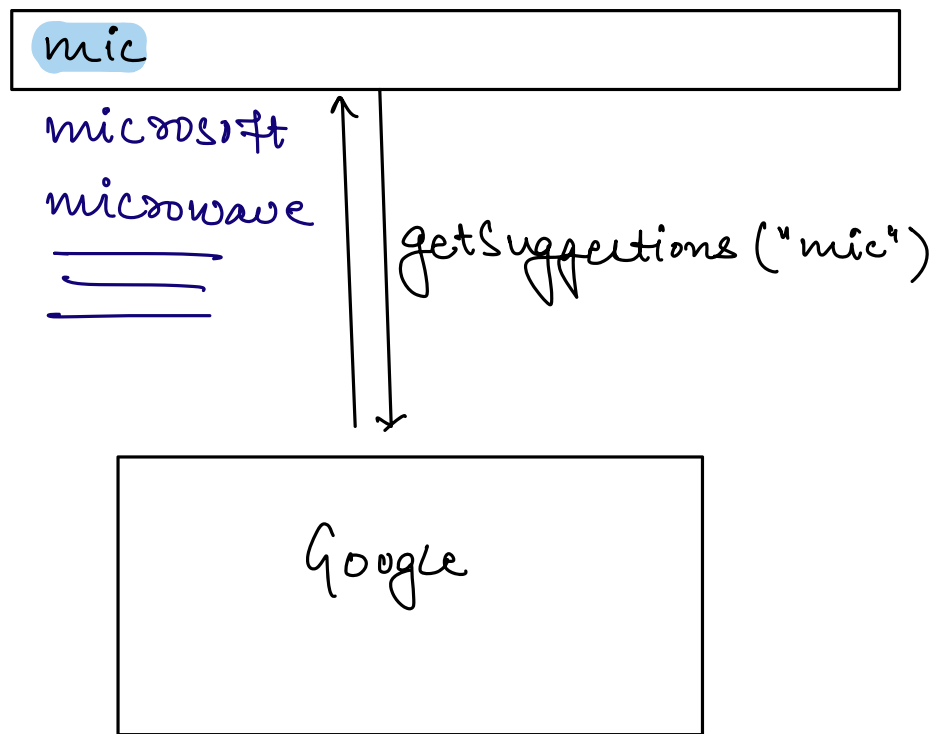
III) SQL (vs) NoSQL.

IV) Can we afford data loss?

### 4) Design Deep Dive



## # Design Google Typeahead



## # MVP : Minimum Viable Product

- get suggestions for the given prefix.
- We'll start giving the suggestions after min 3 characters.
- 10 suggestions.
- Relevant suggestions.

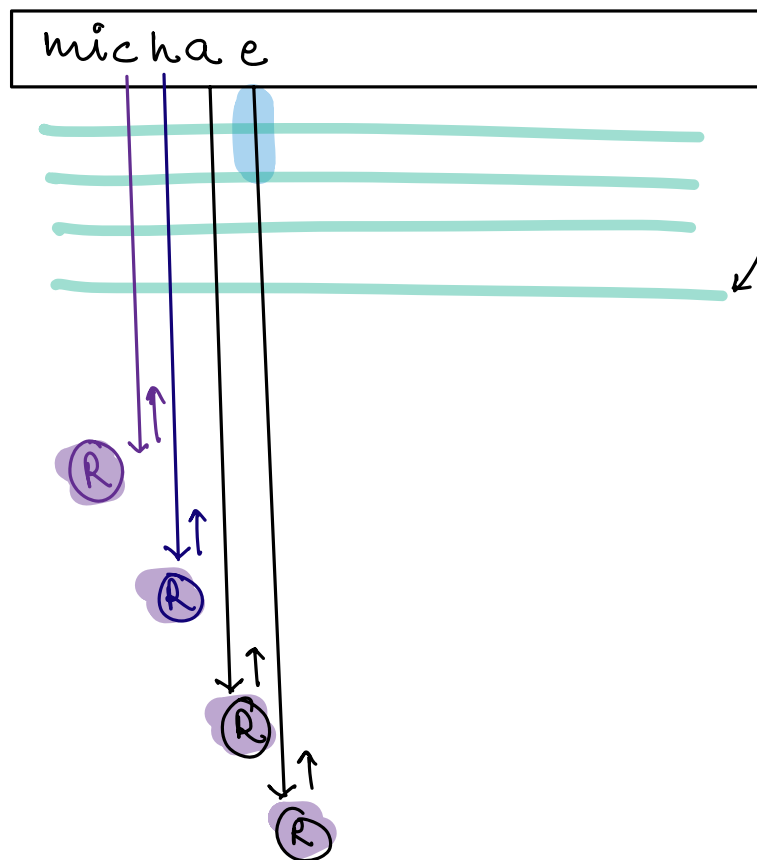
# # Estimation of Scale.

# of users on Google search : 4 Billion

$$DAU = \textcircled{1B}$$

$$\# \text{ of searches/user/day} = 10$$

$$\underbrace{\text{Total \# of searches/day}}_{\# \text{ of writes.}} = \textcircled{10B}$$



# of read queries = 5 \* # of write queries (search)  
Avg no. of calls  
to get the suggestions  
for every prefix before  
we make a search.

# of read queries = 50 B.

Both read & write heavy system

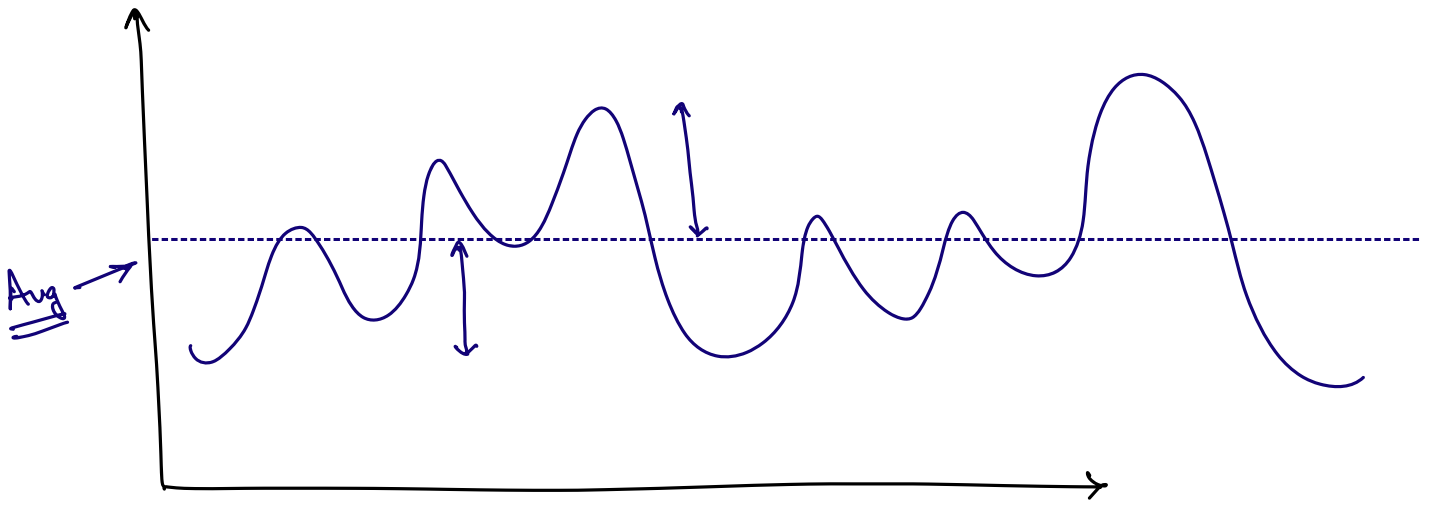
Qps.

↳ Queries per second.

$$\begin{aligned}\text{Write qps} &= \frac{10 \text{ B}}{24 \times 60 \times 60} = \frac{10 \times 10^{17}}{86400} \\ &= 10^5 \text{ qps.}\end{aligned}$$

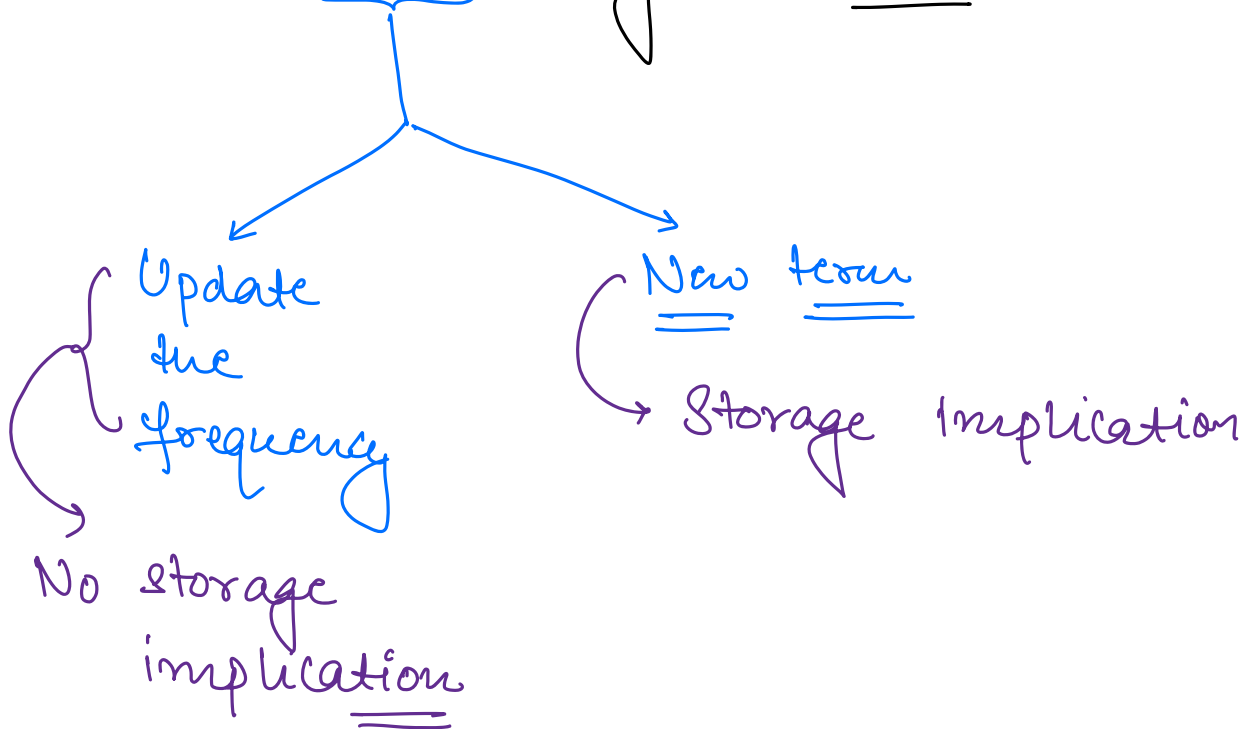
$$\text{Read qps} = 500,000 \text{ qps.}$$

$$\text{Peak qps} = 2 * \text{Avg qps.}$$



⇒ Write queries will contribute to the data in the Backend.

⇒ # of writes / Day = 10 B.



⇒ 10 B searches / Day

⇒ 15% new searches / Day

⇒ 1.5 new searches / Day

Storage per search.



50B

$$\begin{aligned}\text{Storage per Day} &= 1.5 \text{ B} \times 50 \text{ Bytes.} \\ &= \underline{\underline{75 \text{ GB.}}}\end{aligned}$$

$$\begin{aligned}\text{Storage for 10 yrs} &= 75 \times 365 \times 10 \text{ GB} \\ &= 75 \times 400 \times 10 \text{ GB} \\ &= 300 \times 10^3 \text{ GB.} \\ &= \underline{\underline{300 \text{ TB.}}}\end{aligned}$$

⇒ SHARDING is MUST.

# # Design Tradeoffs.

I) Available >>> Consistent

→ Highly Available but Eventually consistent.

II) latency  
→ Super low latency

## # API's

→ get Suggestions (query-prefix, limit = 5)

→ Search (search-query)  
→ Post