Twitter Search

Tuesday, March 26, 2024 8:50 PM

FUNCTIONAL REQS

1. should be able to search (1 or more keywords) and it may support (and / or)

NON FUNCTIONAL REQS

1) LOW LATENCY (≈ 200ms)
2) availability over consistency

CAPACITY ESTIMATION

15 billion total users
500 M daily active users
400 M daily tweets

Traffic: 400 m / day
10:1 (read / write)

Storage:
400 m × 10 ⇒ 4 Billion read ~~tweet~~/day.

4 00 M × 300 bytes ⇒ 120000×10^6
⇒ 120×10^9 ≈ 120 GB per day.

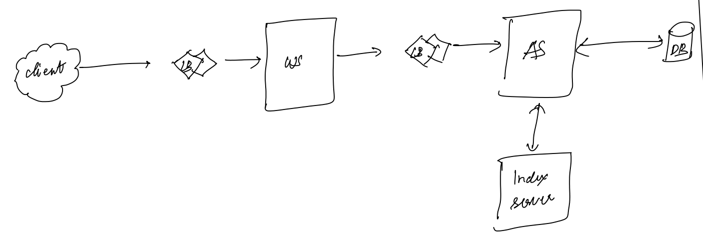
120 GB × 365 × 5
⇒ $600 \times 10^9 \times 365$ ⇒ 1490×10^4 ≈ 200 TB

Bandwidth: $\frac{120GB}{24 \times 60 \times 60}$

Cache ≈ $0.2 \times \frac{120GB}{24 \times 60 \times 60}$

INDEXING - It is a technique to help search faster

LOW LEVEL DESIGN



leetcode: 14474891, 14484792 ...
subscription: 15475961, 15478472 ....

[words: List of tweet Ids]

Assume only english language

300 k English words
200 K Nouns

Amazon is offering a discount of 20 %

15 unique words per tweet

1400 × 15 + 2.5 = 20000 = 20TB
↓
no of english words

Assuming a single server can accommodate - 200 GB

$\frac{20TB}{200GB}$ ≈ 100 (we need 100 servers)

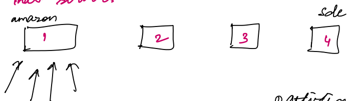
SHARDING

1. words
2. tweet Id

If some one searches for leetcode subscriptions

hash (leetcode) direct to index server.
hash (subscriptions)

Index servers

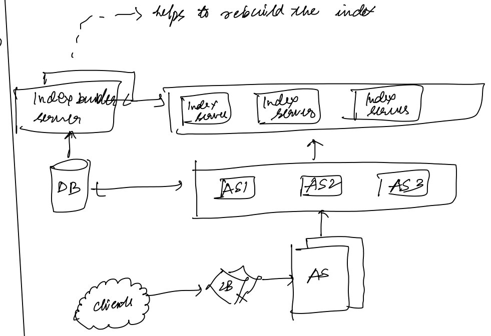


partition based on words

If amazon becomes a buzz word

(HOT WORD PROBLEM)

Partition based on tweetId - we can have hot tweet problem.
To avoid problem of hashing use consistent hashing -
How is the system going to react incase of a problem.



ranking all tweets
pick top 20

Inverted index
words: list of tweets

If partition is based on words.

If Index server goes down
rebuild the index server.
partition is based on words

Total 500K words × 5 (characters) = 2500K = 2.5 MB

Real time Index ! Index that has been built for certain time period.

400M (tweets/day) × 365 days × 2 years ⇒ 730 × 400M
⇒ 292 000 × 10^6
⇒ 280 billion tweets

as we are maintaining for 2 years.

280 billion × 5 (no of bytes for single tweet Id) ⇒ 1400 × 10^9 ⇒ 1.4 TB (space for tweet Id)

[Index server 1 : List of words]

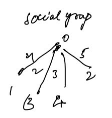
If partition is based on tweetId.

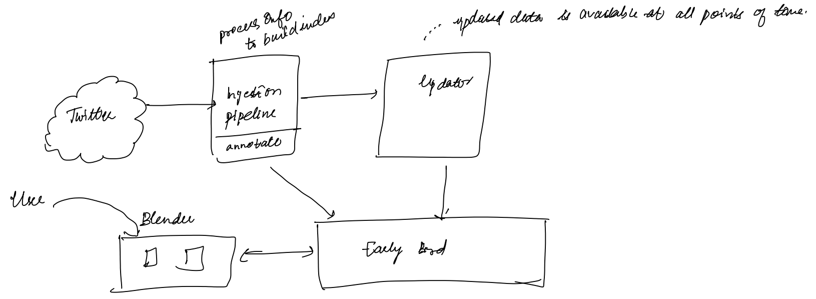
[Index server 1 : TweetIds]

DB access is relatively slower
It takes forever to rebuild.

TweetId is faster as DB has PK of tweet ids.

Ranking :
1. Time
2. Frequency (retweets, likes)
3. Social graphs




process info to build index
... updated data is available at all points of time.
Twitter
Ingestion pipeline
annotate
Updater
User
Blender
Early Bird

Blender
sort and give top 20 results
works similar to aggregator

Gives access to early informations

Lucene Index — commercially available index, helps in task of building and retrieving indexes

API

search (dev_key, search_string, page_index, sort, tweets/page