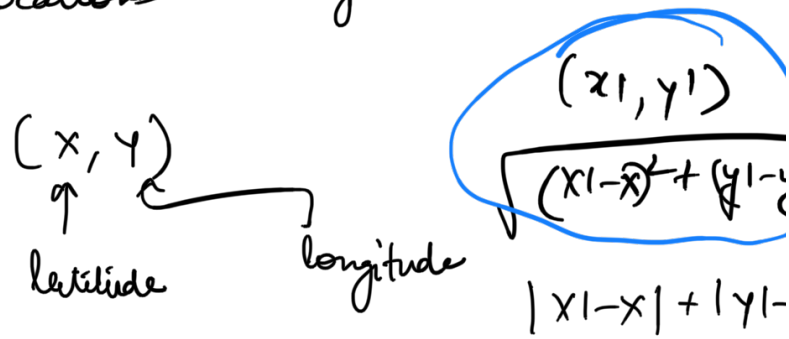
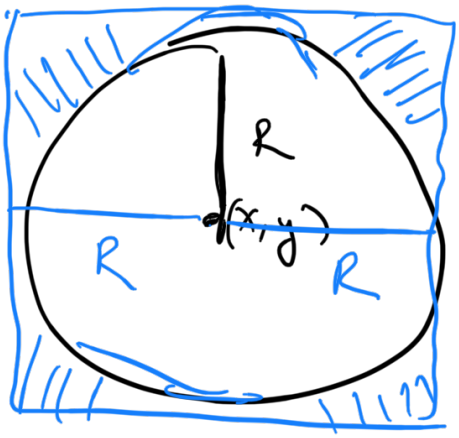
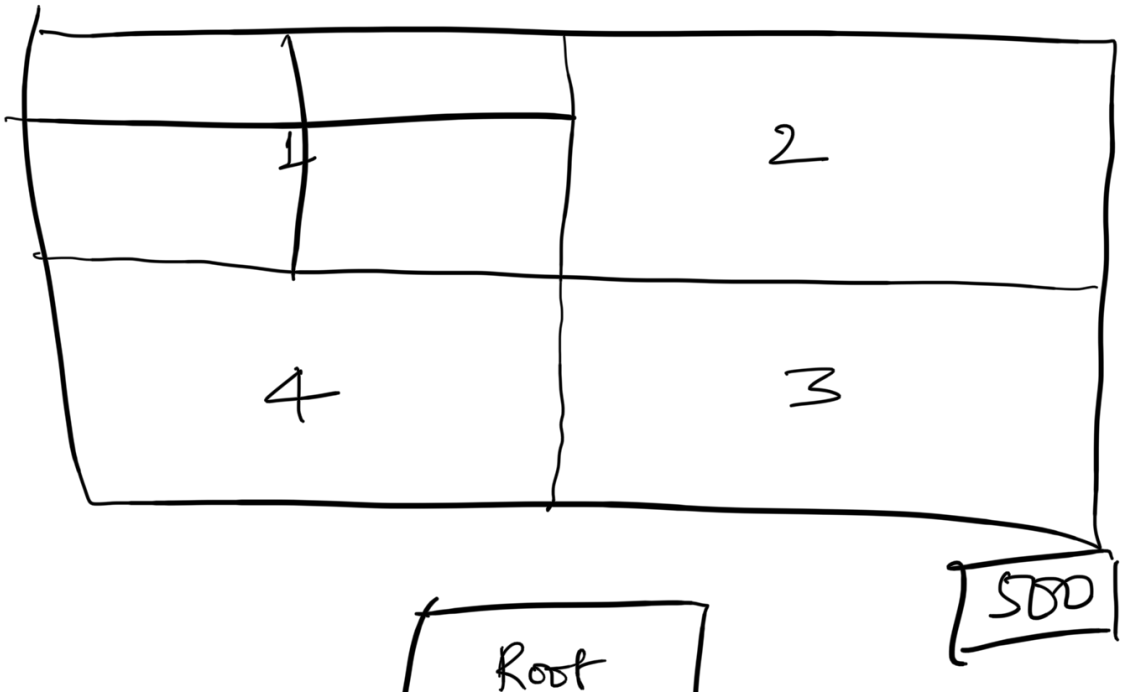
PLACE BASED SYSTEMS

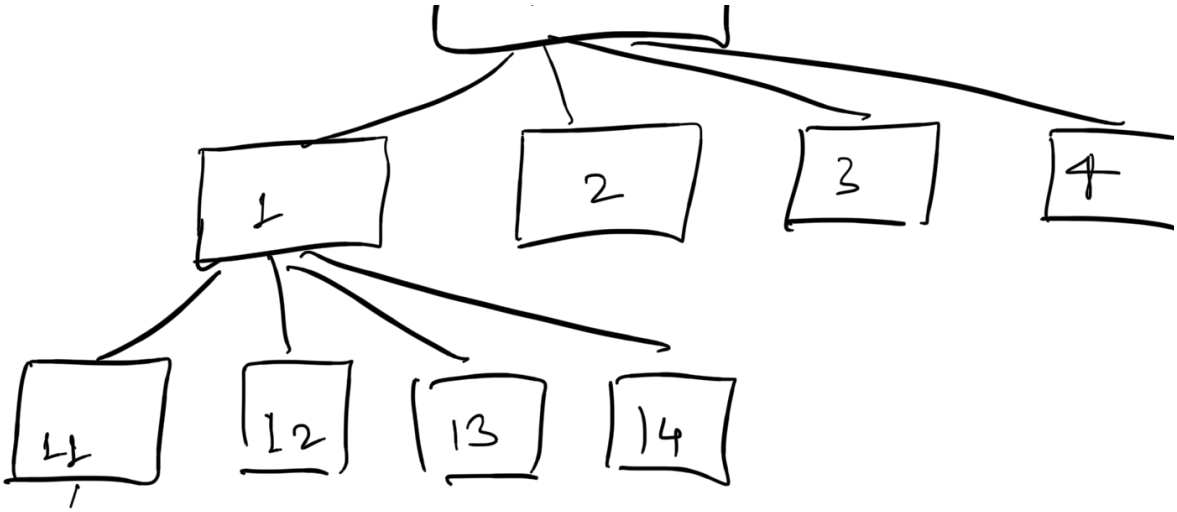
1. Nearest k neighbors given my location
2. Locations in given R radius





x → x-R, x+
y → y-R, y+





Node

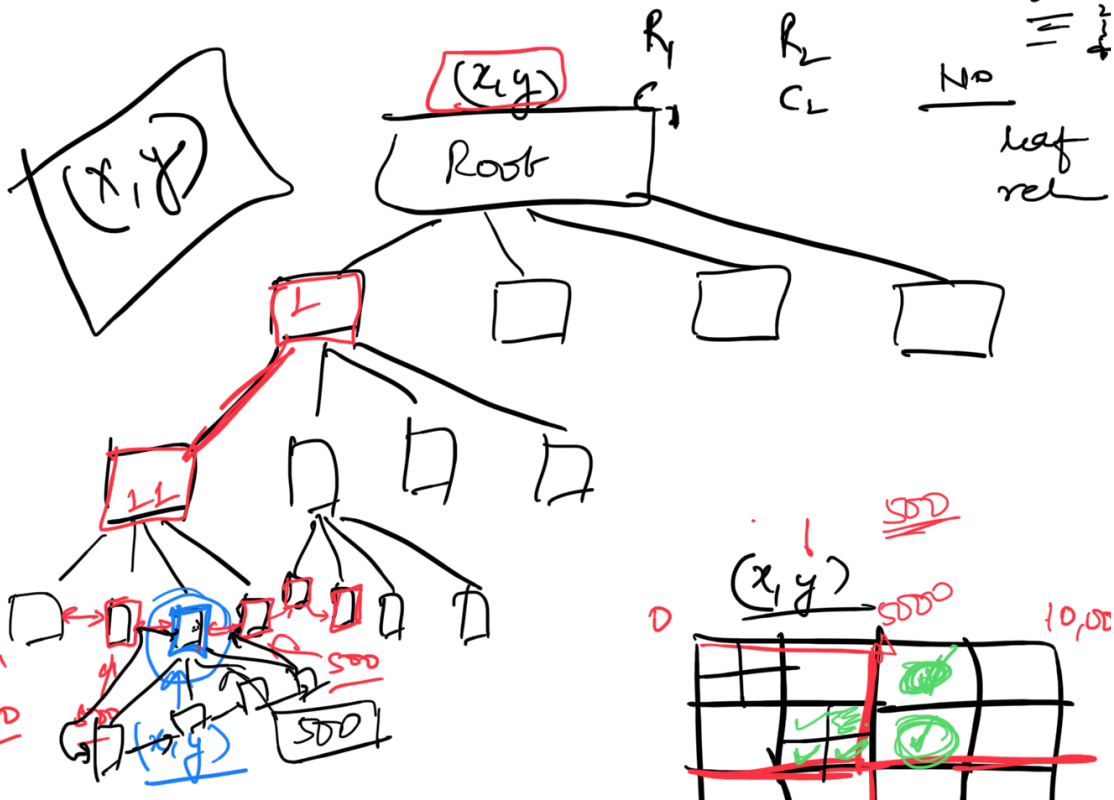
① Do I have more than 500 poi

YES

child 1 = 2 3 4

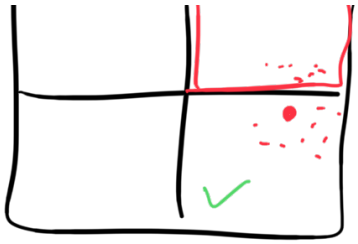
NO

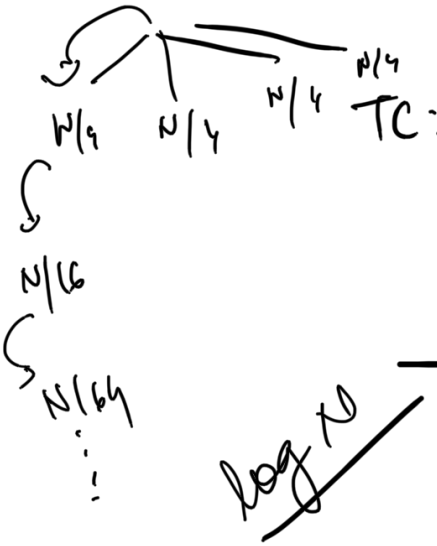
leaf rel



Database

Spatial





① ✓ Insert a new place → O(log N)

② ✓ Deletion of a place → O(log N)

③ Finding nearest (K) neighbors.

→ O(K + log N)

1 Billion Place.

10^{18}

$1\,GB \times 10^9$

TB

PB

EB

1 EB

DESIGN YELP

① Discuss MVP

① Search places near me

② Places
- Rating → 5B
- Description → 500B
- Category → 25B
- Media → diff storage

Latency

↳ Location → (x, y) → 8B → 16
↳ Timings → 12B

③ User account → Owner | Acco
↳ give ratings

④ Add / delete places.

SCALE

① City → 50000
↳ 15,000
↳ 5,000 others.
→ 25K

20k x 30k = 600 million globally

200

1TB ← Place → 1B
↳ 1KB

3TB ← User → 1B
↳ Account -
↳ Ratings → 4B
↳ Comments → 500B * 5 = 2KB (3KB)

4TB

Country → 100GB

QPS → 1 Billion Total

200M DAU → 200M search a day.

~~200~~ * 10^6
~~86400~~
~~432~~ 500

2000 QPS

8000 QPS
↳ lunch/dinner/breakfast

→ Eventual consistent

- C
- (A) ← ✓

Filters

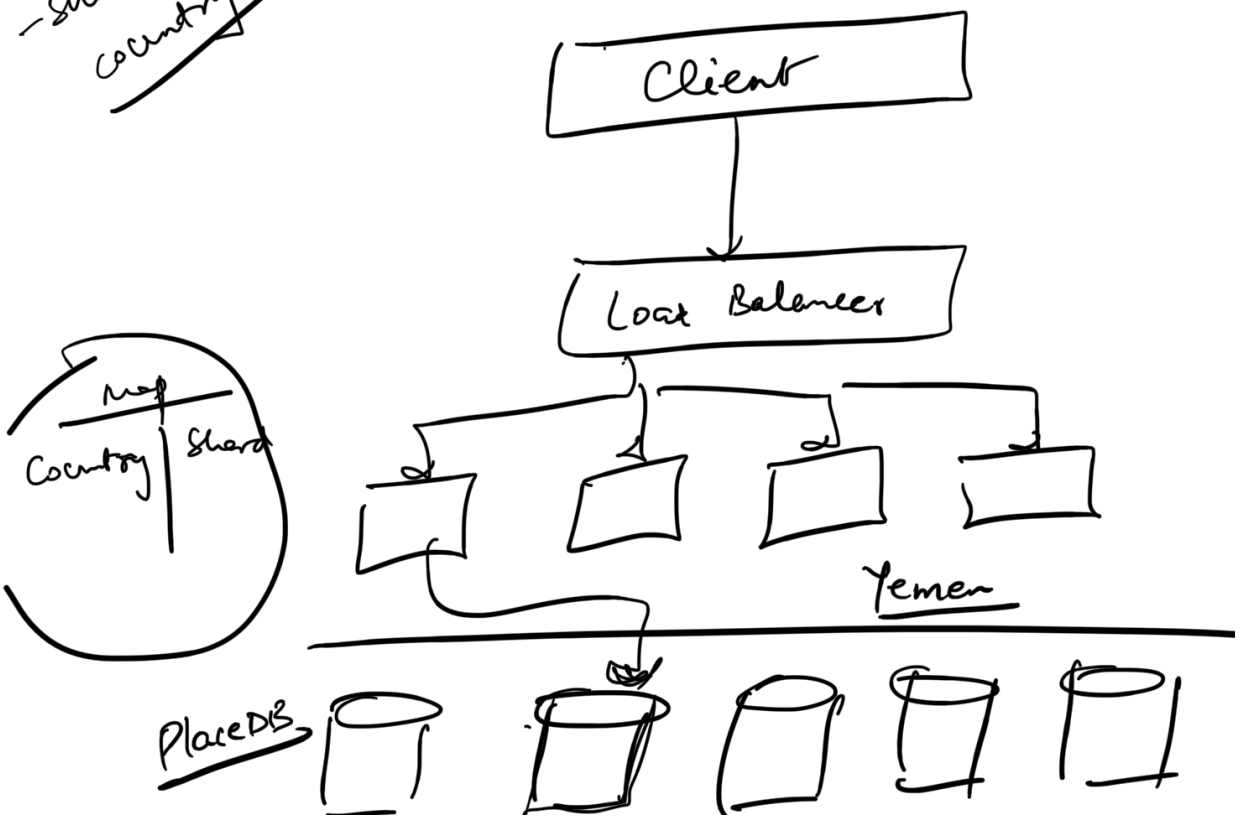
nearby

Sort by: popul

API design:

- search (userId, location, radius, filters, num-results, offset, sort,)
  filters ↑ JSON
- fetchplaceProfile (userId, place_id, minifie = false
- addPlace (------)
- deletePlace (userId, place_id)
- addReview ($user_id, place_id, rating, review timestamp)

- Shard by country → no further chording.



India

→ Place data

→ Sample of 5 handpicked ratings & cr