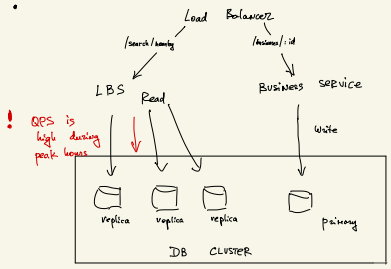
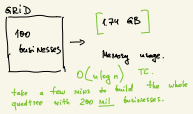
Proximity Service

- 20 km Radius [0.5, 1, 2, 5, 20]
  add / delete / update
  Do not refresh if speed is slow.
- return lat / long / radius
  add / delete / update not in realtime
  view detailed information about a business
- low latency
  privacy GDPR, CCPA
  high availability, scalability
- 100 mil of DU . Users 5 queries per day.
  QPS = $\frac{100 \text{ mil} \times 5}{10^5} = 5\,000$
- GET /v1/search/nearby ← req latitude: longitude: radius:
  + pagination → resp "total": "businesses": [bus. obj.]
  GET /v1/businesses/:id
  POST /v1/businesses
  PUT /v1/businesses/:id
  DELETE /v1/businesses/:id
- Data . Read volume is high. Write volume is low. (MySQL).

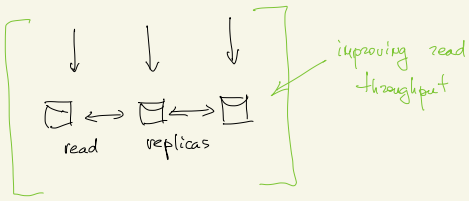


- Fetch nearby businesses:
  1. Two dimensional search + indexes ! O(N)
  2. Evenly divided grid uneven data distribution regarding grids
  3. Geohash easy to implement
     long / latitude => one dimensional string recursively dividing in Base32 presentation
     ex. 7hc 7qa
     has 12 precisions . Boundary issues
     • Bing map Redis Mongo Lyft
  4. Quadtree slightly harder to implement
     Leaf node = 832 bytes
     Internal node = 64 bytes
     • Elastic Search
     

  5. Google S2 geometry library
     google maps, tinder.
- Update . Deploy incrementally.
  Blue/Green deployment is acceptable.

- Deep Dive
  Scale DB:
  Sharding of Business Table ?

geohash	business_id
—	—
—	—
—	—



- No read cache solution .
  Can provide a good I/O . Would work almost as fast as an in-memory cache.