#ASLI ENGINEERING

Realtime Data Aggregation with DynamoDB

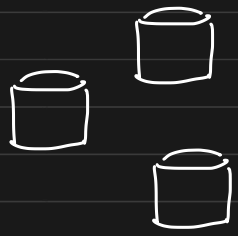
SWIPE

BY

ARPIT BHAYANI

Realtime Data Aggregation with DynamoDB

What is DynamoDB?



DynamoDB is low-latency KV store offered by AWS.

Operations it supports are GET, SET, PARTIAL UPDATE, DEL, TTL

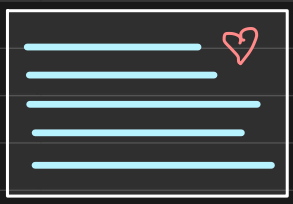
Deliveroo Data Model

Deliveroo does food delivery and it is quite popular in Europe.

1. Restaurant owners list their restaurants
2. Users can browse through them, pick their favourite, and order

Users can also mark some restaurants as 'FAVOURITE'

favourites table



restaurant_id _ user_id : Partition key

↑ This restaurant 'favourite' of this user ←

created_at
user_id
} other attributes

1. Getting if a user 'u' has restaurant 'r' in favourite

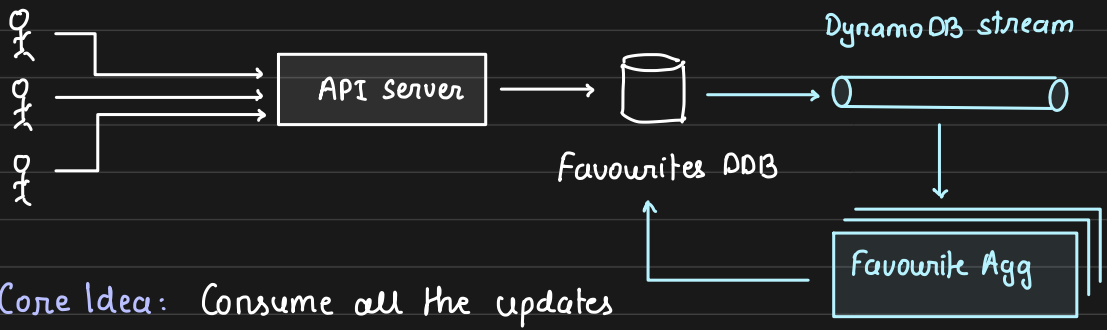
   is just a simple GET (KV lookup)

2. Marking a favourite or unmarking are pointed operations

Thus we see, all user operations happen in O(1) time

Feature request : Show list of restaurants ordered by FAV counts

Hence we would need to "aggregate" favourites and keep the favourite count handy at a restaurant level.



Core Idea: Consume all the updates happening in Favourites table and aggregate them in a different table

Aggregated Favourites

restaurant_id : partition key

time_window : sort key

↳ ALL_TIME , AUG 2022 , 20220801

favourite_count

updated_at

A	ALL_TIME	729
A	AUG 2022	42
B	ALL_TIME	10072

Aggregator function would

- receive a batch of updates happening in Favourites table
- filter INSERT and REMOVE events
- atomically, increment or decrement the count per restaurant
  - ↳ use ADD instead of SET
  - ↳ use DDB transactions to batch update 100 writes

Isn't Lambda costly?

Not a frequent action at all

Fav per day ~ 7000

Each execution ~ 25ms

Mem consumption 64 MB

} < $1 per month

Advantages

- extremely cost efficient for given user behaviour
- APIs response time unaffected because ASYNC
- better than traditional approaches
  - daily full table scan (not realtime)
  - synchronous updak to Aggregakd Favourik table
    ↳ (APIs would become slower)