Lab 4 - Caching

Presented by - Darshana Patil

ARCHITECTURE & ETL DATA FLOW (REDIS → MONGODB)

1. Node.js connects both redis and mongoDB.
2. Request sends to Node.js server.
3. server checks the cache (redis)
4. If there is a data available in cache (cache hit) it will fetch it from cache in

   minimum time.
5. If data is not availabe (cache miss) then it goes to mongoDB data base - get the data and store in cache (redis) - return data. (takes more time)

LEARNING

- Less time to fetch data, so Faster in performance.
- Learn how to implement different cache patterns and how they act on data.
- Different Cache pattern has different purpose so we can use it as per the needs.
- for eg. TTL has the expiring data feature which will gives us the updated data.
- likewise cache-aside etc. shows how lesser the timeit takes than comparing to the database call.

CHALLENGES -

- What work well ?
  - connecting redis and mongoDB.
  - inserting the 2000 records in mongo
- What/ where i faced challenges ?
  - at first Running and testing all these cache pattern was a challenge for me but then when i debugg it many times i started getting the differences in output (timing).
  - implementing these cache and how they are different from each other was a challenge but i find it more interesting to implement them and specially run them on postman and track the timing difference.