Rao, Praveena

CrAFT DEMO-TWITTER

## Requirements

* Able to post message(C,U,D).
* Able to follow/Unfollow people.
* See messages from people you follow.
* My messages-My timeline.
* Trending- by location?
* Search tweets?

### Scale

* 10K+ users
* Writes – 10 tweets/sec
* Reads – 100 tweets/sec

## Data Base Design

#### User - Table

UserId, name, age

#### Tweets - Table

T\_id, userId, tweet

#### Follow - Table

Id, userId, followUserId

## Cache -Redis

HasMap of UserId and Stack.

Home Timeline

User\_1 : [tweet11, tweet21, tweet31]

User\_2 : [tweet11, tweet21, tweet32]

User Timeline

User\_1 : [tweet11, tweet12, tweet13]

User\_2 : [tweet21, tweet22, tweet23]

User – Celebrity

User 1 : [C1, C2, C3]

## Computing Home Timeline

1. Get all followers
2. Get Latest tweets
3. Sort by time

## Trending

Tweet

->filter(remove most common ones like #fun, #food, #travel)

->parse

->Geo-Location

->Count Hashtag

->Rank

## Performance and special cases

* Celebrity followers Cache.
* When Celebrity tweets don’t fan out. Load it along with user home timeline.
* Get the celebrity timeline and read the tweets.
* Don’t compute home timeline for inactive users.

## Design

Load Balancer.

SpringBoot rest micro services.

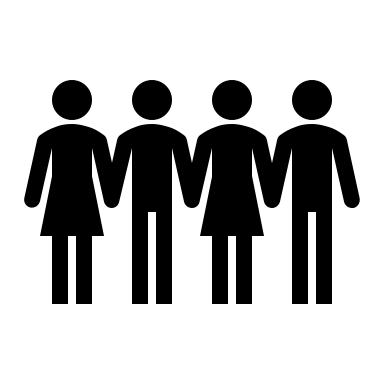
Use pagination for home and user timeline.

Elastic search to index the tweets and search.

LDAP integration for user management.

Redis cache to fanout and store user timeline.

Any SQL/NoSQL Db to store data.



TweetService

DB

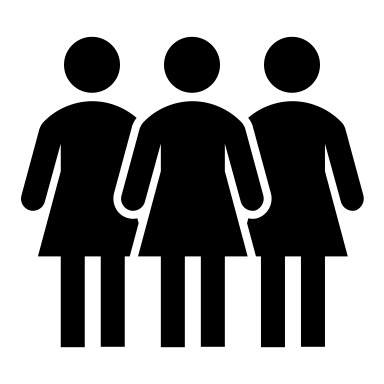
DB

Load Balancer

Users

UserService

Redis



FeedService

FANOUT

LDAP

SearchService

Elastic