Twitter
High-Level Solution Design Document

Version: 1.00

Updated: 04/09/2022

Author: Ashish Nikumbh

1. Introduction

Twitter is a microblogging and social networking service on which users post and interact with messages known as "tweets", owned by American company Twitter, Inc. Registered users can post, like, and retweet tweets, while unregistered users only have a limited ability to read public tweets. Users interact with Twitter through browser or mobile frontend software, or programmatically via its APIs.

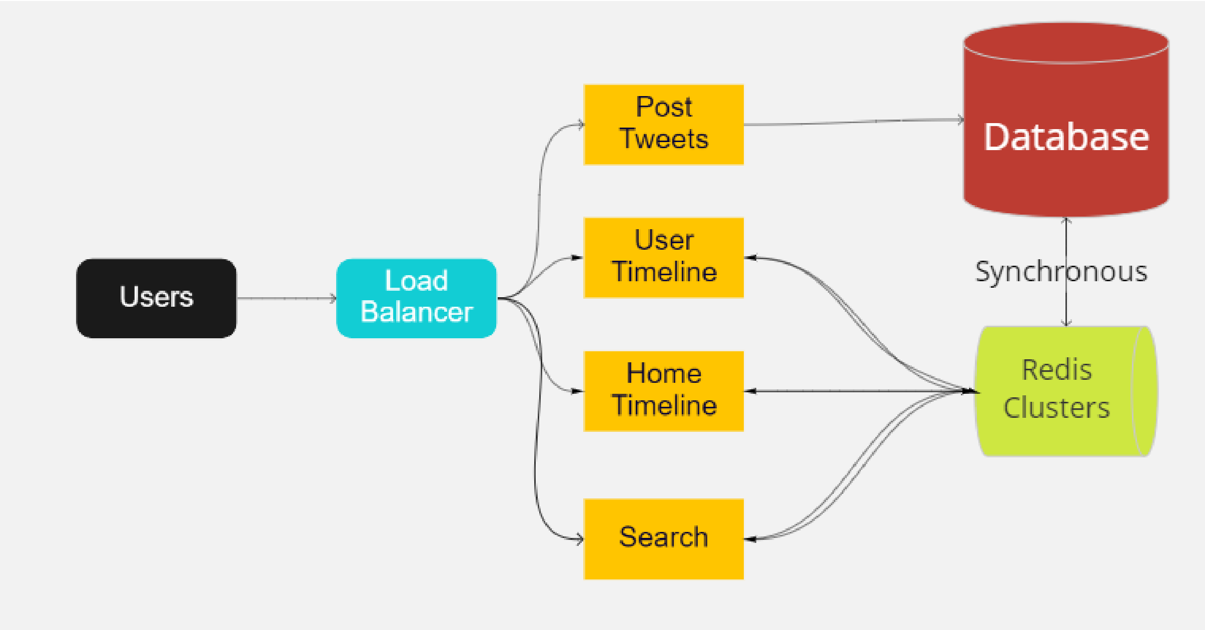
2. Requirement Summary

- Design Twitter application User should be able to tweet
  - Timeline -> Home timeline , User timeline, Search timeline
  - User should also be able to see trending hashtags and topics.

3. Assumptions and Prerequisites

- Twitter is more used for reading purpose then for write tweets. It is more ready-heavy than write heavy. So let's assume:
  - Tweets per sec : 600
  - Reads per sec : 600000
- We will focus more on accessing database then to write in it.

4. High-Level Design



- The interaction happens between main user portal and database (as shown in figure)

4.1. Application Modules

Includes the following:

- User portal where default user can see home timeline
- User can see its own timeline
- User can search on the basis of hashtags
- Server layer where all fetching and retrieving functions takes place

4.2. Transactions and User Flows

- Home Timeline: all the tweets database is fetched when this page shown.
  - Get all the data from REDIS and if required then fetch from database
- User Timeline: only user id related tweets are fetched.
  - Get all the tweets from the REDIS for related user
- Search : only data which are related with the entered hashtags will be fetched
  - Get all tweets from the REDIS
- Post tweets: user can submit the tweets and if hashtags are contained then it should be saved in hashtag table with the id of the tweet. This should be saved first in database and then to redis.