Twitter High-Level System Design

The Core Features :

We are going to consider the following features on Twitter

- The user should be able to tweet
- The user should be able to see Tweet Timeline(s)
- **Timeline:** This can be divided into two parts
    - *User timeline:* User sees his/her own tweets and tweets user retweet. Tweets that users see when they visit their profile.
    - *Home Timeline:* This will display the tweets from people users follow.
- The user should be able to follow another user.
- Users should be able to tweet millions of followers within a few seconds.

Database Design:

We can use a relational database like MySQL and you can consider two tables **user table (id, username)** and a **tweet table[id, content, user(primary key of user table)]**. User information will be stored in the user table and whenever a user will tweet a message it will be held in the tweet table. Two relations are also necessary here. The user can follow each other, the other is each feed has a user owner. So there will be a one-to-many relationship between the user and the tweet table.

Limitation of Architecture

We will have to do a big select statement in the tweet table to get all the tweets for a specific user whomsoever he/she is following, and that's also in chronological order. Doing this every time will create a problem because the tweet table will have huge content with lots of tweets. We need to optimize this solution to solve this problem and for that, we will move to the high-level solution for this problem. Before that let's understand the characteristics of Twitter first.

User Table, Tweet Table, and Followers Table.

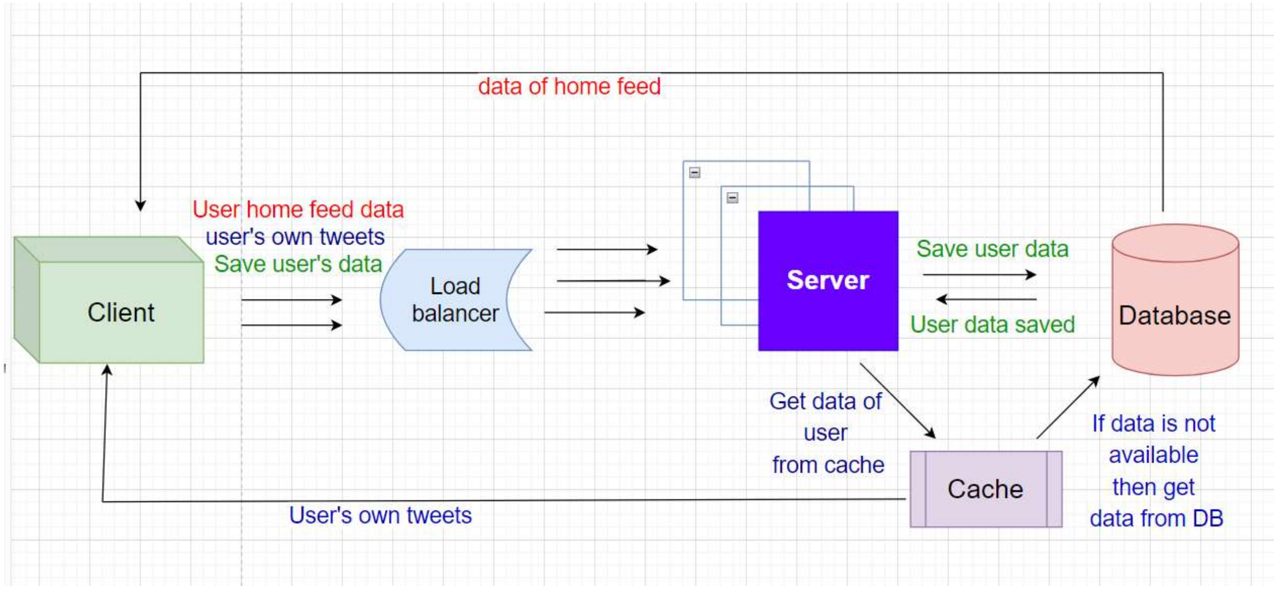
- Whenever a user will create a profile on Twitter the entry will be stored in the User table.
- Tweets tweeted by a user will be stored in the Tweet table along with the User_id. Also, the User table will have **1 too many relationships** with the Tweet table.
- When a user follows another user, it gets stored in Followers Table, and also caches it Redis. The User table will have **1 too many relationships** with the Follower table.

Estimates :

Twitter has 300M daily active users. On average, every second 6, 000 tweets are tweeted on Twitter. Every second 6, 00, 000 Queries are made to get the timelines. Each user has on average 200 followers and some users like some celebrities have millions of followers.

Architecture :

Twitter is read-heavy so we need a system that allows us to read the information faster and also can scale horizontally.



To get the User Timeline simply go to the user table get the user_id, match this user_id in the tweet table and then get all the tweets. This will also include retweets, save retweets as tweets with original tweet references. Once this is done sort the tweet by date and time and then display the information on the user timeline.

A user's Home Timeline contains all the latest tweets of the person and the pages that the user follows. Well, here you can simply fetch the users whom a user is following, for each follower fetch all the latest tweets, then merge all the tweets, sort all these tweets by date and time and display them on the home timeline.

Drawbacks of Architecture :

This solution has some drawbacks. The Twitter home page loads much faster and these queries are heavier on the database so this huge search operation will take much more time once the tweet table grows to millions.