Requirements of a Newsfeed System's Design

Get introduced to the requirements and estimation to design a newsfeed system.

We'll cover the following

- Requirements
  - Functional requirements
  - Non-functional requirements
- Resource estimation
  - Traffic estimation
  - Storage estimation
  - Number of servers estimation
- Building blocks we will use

Requirements

To limit the scope of the problem, we'll focus on the following functional and non-functional requirements:

Functional requirements

- **Newsfeed generation:** The system will generate newsfeeds based on pages, groups, and followers that a user follows. A user may have many friends and followers. Therefore, the system should be capable of generating feeds from all friends and followers. The challenge here is tha there is potentially a huge amount of content. Our system needs to decide which content to pick for the user and rank it further to decide which to show first.

- **Newsfeed contents:** The newsfeed may contain text, images, and videos.
- **Newsfeed display:** The system should affix new incoming posts to the newsfeed for all active users based on some ranking mechanism. Once ranked, we show content to a user with higher-ranked first.

Non-functional requirements

- **Scalability:** Our proposed system should be highly scalable to support the ever-increasing number of users on any platform, such as Twitter, Facebook, and Instagram.
- **Fault tolerance:** As the system should be handling a large amount of data; therefore, partition tolerance (system availability in the events of network failure between the system's components) is necessary.
- **Availability:** The service must be highly available to keep the users engaged with the platform. The system can compromise strong consistency for availability and fault tolerance, according to the PACELC theorem.
- **Low latency:** The system should provide newsfeeds in real-time. Hence, the maximum latency should not be greater than 2 seconds.

Resource estimation

Let's assume the platform for which the newsfeed system is designed has 1 billion users per day, out of which, on average, 500 million are daily active users. Also, each user has 300 friends and follows 250 pages on average. Based on the assumed statistics, let's look at the traffic, storage, and servers estimation.

Traffic estimation

Let's assume that each daily active user opens the application (or social media page) 10 times a day. The total number of requests per day would be:

$500M \times 10 = 5$ billions request per day $\approx 58K$ requests per second.

Traffic estimation for the newsfeed system

Storage estimation

Let's assume that the feed will be generated offline and rendered upon a request. Also, we'll precompute the top 200 posts for each user. Let's calculate storage estimates for users' metadata, posts containing text, and media content.

1. **Users' metadata storage estimation:** Suppose the storage required for one user's metadata is 50 KB. For 1 billion users, we would need $1B \times 50KB = 50TB$.

   We can tweak the estimated numbers and calculate the storage for our desired numbers in the following calculator:

Storage Estimation for the Users' Metadata.

Number of users (in billion)	1
Required storage for one users' met adata (in KBs)	50
Total storage required for all users (i n TBs)	*f* 50

2. **Textual post's storage estimation:** All posts could contain some text, we assume it's 50KB on average. The storage estimation for the top 200 posts for 500 million users would be:

   $200 \times 500M \times 50KB = 5PB$

3. **Media content storage estimate:** Along with text, a post can also contain media content. Therefore, we assume that $1/5th$ posts have videos and $4/5th$ include images. The assumed average image size is 200KB and the video size is 2MB.

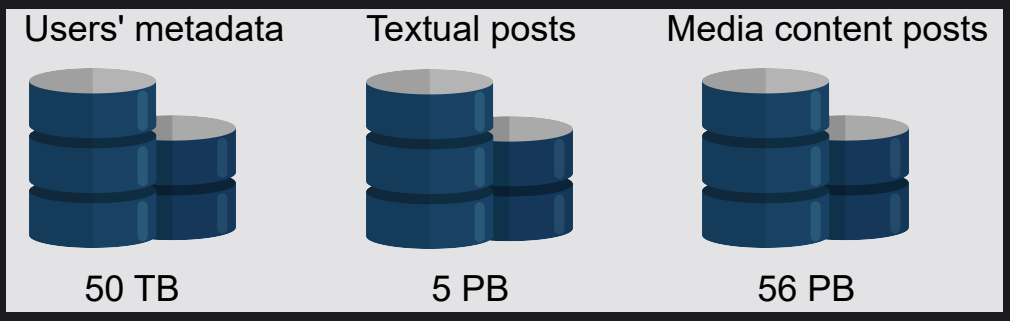
   Storage estimate for 200 posts of one user:

   $(200 \times 2MB \times \frac{1}{5}) + (200 \times 200KB \times \frac{4}{5}) = 80MB + 32MB = 112MB$

   Total storage required for 500 million users' posts:

   $112MB \times 500M = 56PB$

   So we'll need at least 56PB of blob storage to store the media content.



Storage required for 500 million active users per day (each with approx. 200 posts) by newsfeed system

Storage Estimation of Posts Containing Text and Media Content.

Number of active users (in million)	500
Maximum allowed text storage per post (in KBs)	50
Number of precomputed posts per user (top N)	200
Storage required for textual posts (in PBs)	5

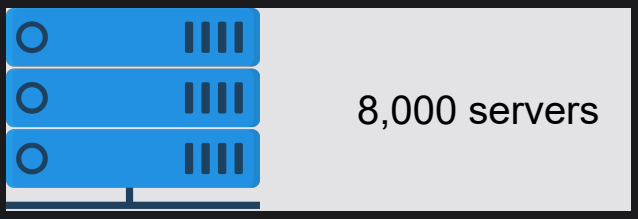
Total required media content storage for active users (in PBs)	56

Number of servers estimation

Considering the above traffic, let's estimate the required number of servers during peak load. Recall that a typical server can serve 64,000 requests per second (RPS). Considering our assumption of using daily active users as a proxy for the number of requests per second for peak load times, we get 500 million requests per second. Then, we use the following formula to calculate the number of servers:

$$Servers\ needed\ at\ peak\ load = \frac{Number\ of\ requests/second}{RPS\ of\ server}$$

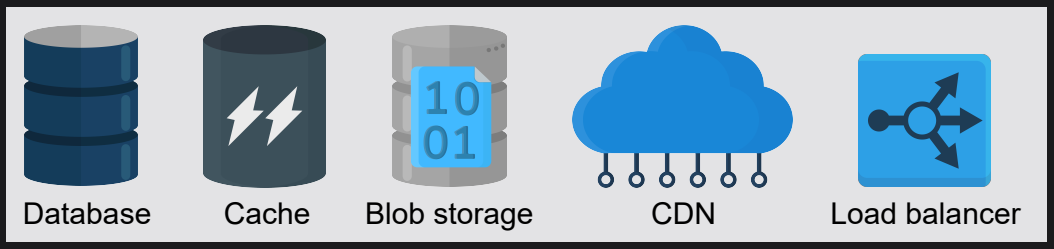
$$Servers\ needed\ at\ peak\ load = \frac{500\ million}{64,000} = 7812.5 \approx 8K\ servers$$



Number of servers required for the newsfeed system

Building blocks we will use

The design of newsfeed system utilizes the following building blocks:



The building blocks to design a newsfeed system

- **Database(s)** is required to store the posts from different entities and the generated personalized newsfeed. It is also used to store users' metadata

and their relationships with other entities, such as friends and followers.

- **Cache** is an important building block to keep the frequently accessed data, whether posts and newsfeeds or users' metadata.
- **Blob storage** is essential to store media content, for example, images and videos.
- **CDN** effectively delivers content to end-users reducing delay and burden on back-end servers.
- **Load balancers** are necessary to distribute millions of incoming clients' requests for newsfeed among the pool of available servers.

In the next lesson, we'll focus on the high-level and detailed design of the newsfeed system.