

YouTube System Design

Functional Requirements

- Upload video
- Watch video

Non-Functional Requirements

- **Reliability of videos**
 - No crash or corruption
 - Video player works
- **1B Daily users**
 - Watch 10 videos/day per user
 - Most are watchers, so 1:50 ration of uploaders to watchers
 - 2% of 10B videos = 200M videos

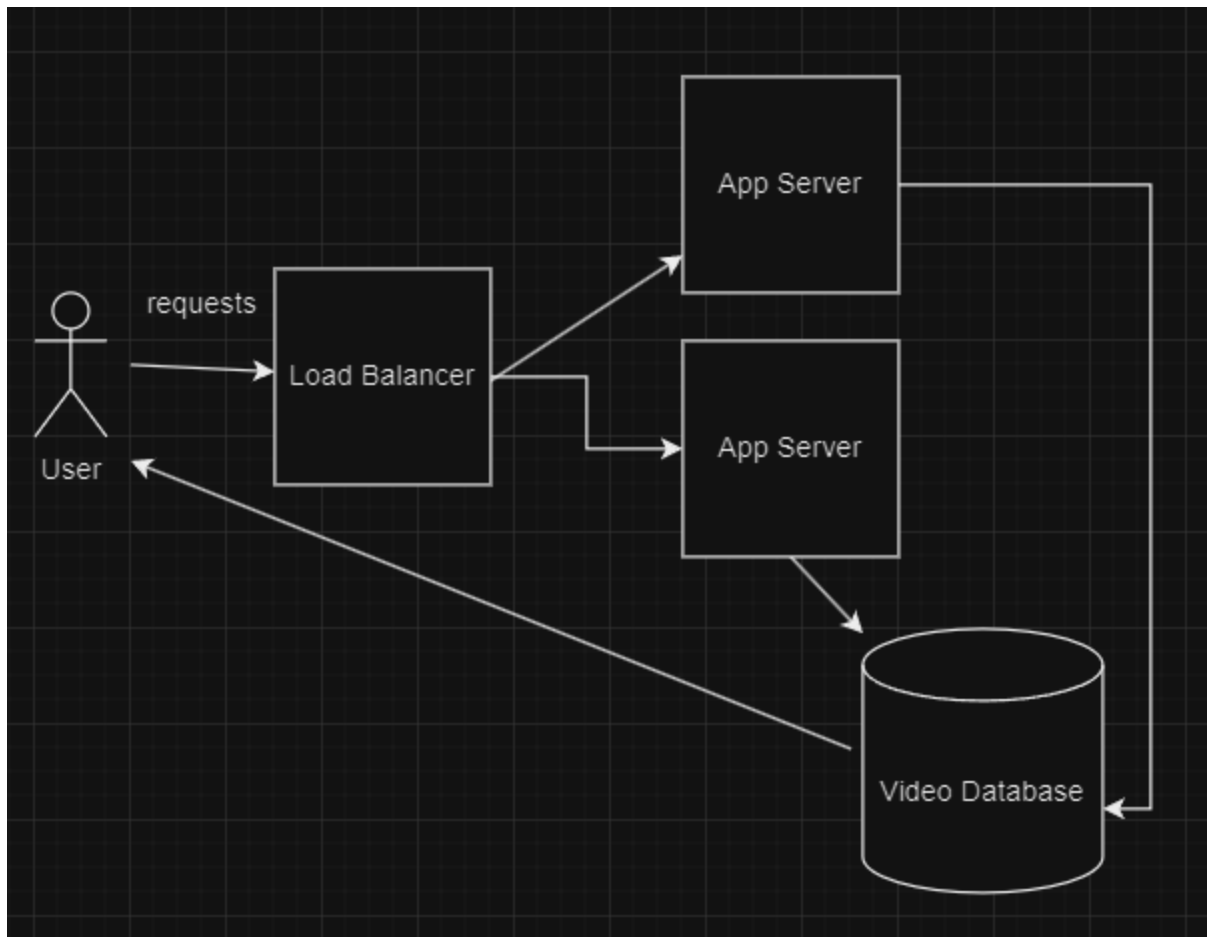
Overall Assumptions

- 1B daily users
- 10 videos/day per user
- Ration of uploaders to watchers 1:50, so 200M videos

Storage Estimation

- Average video about 10minutes = 1GB
- 200M GB for uploads

High-Level Design



App server – handles communication between user and video database

DB for storing all videos and metadata

Load Balancer to handle all of users connection requests to video database

Upload(video,videoID,userID)

GetFeed(uid)

Subscribe(uid,user2id)

Play(vid)

Pause(vid)

Load(vid)

Skip(skipTime,vid)

Search(text, list of related videos)

Database – relational database like a graph to maintain following/followers videos

End to End Flow

Goal: UserA wants to search for VideoA

1. UserA searches for VideoA
 - a. Connects to video database and tries to find relevant information
 - b. Video database sends back data
2. List of videos are shown
3. UserA clicks on VideoA
 - a. VideoA is loaded from database
4. UserA is taken to a new page where VideoA is loaded
5. UserA can play,pause, and skip

Goal: UserA clicks video on their feed of VideoA

1. App Server shows userA a feed of videos
 - a. AppServer collects relevant videos from userA and connects to video database to gather those videos
2. UserA clicks on VideoA
 - a. VideoA is loaded from database
3. UserA is taken to a new page where VideoA is loaded
4. UserA can play,pause, and skip