Workflow System

Thursday, 10 December 2020 7:19 PM

1. Gather Requirements
2. Estimate scale
3. Schema + API's
4. Design Goals
5. Remove bottlenecks and scale according to the requirement

Video ingestion System

1. Requirements.

- Any signed in user can upload a video
- filter by language, category
- generate thumbnail, preview.
- The video must adhere to copyright claims.
- support geo-restrictions
- supported formats: mp4, mov
- size: 100 MB
- verify video quality (720p)
- scale down the video to 320p, 480p

2. Scale

- 2 Billion users
- 1% users uploads video every day

TPS: $\frac{2 \times 10^9}{100} = 20M$ video/day. $= \frac{2 \times 10^7}{24 \times 3600} \sim 250$ /sec

- 50% users consume a video

TPS: ~ 10 K/sec

Storage: 200M uploads × 100 MB/upload

$= 2 \times 10^{10}$ MB

$= 20$ PB

$\frac{2 \times 10^5}{24 \times 3600}$

3. Schema

User
- type
- user_id

Video
- user_id
- video_id
- languages
- categories
- geo_loc^n
- format
- resolution
- thumbnail
- preview

video_language
- video_id
- lang

video_category
- cat_id
- video_id

category
- id
- name

4. Design Goals

- Partition
- Consistent = ? Yes
- Availability = ? Yes
- latency: can be high
- scale: horizontally.

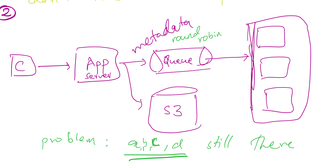
5. Initial Design

C → API → S → RDS

1. copyright check (hash)
2. apply country laws (separate system)
3. format check (metadata)
4. check all video frames (app svc)
5. convert in multiple formats. (self svc)
6. generate thumbnail (app svc)
7. generate preview (ss)
8. convert to lower res (ss)
9. save

(a) bulky code
(b) each step can take different time.
(c) can fail without altering
- d start the pipeline again.
- traffic will be uneven (Queue)
- ① don't need to respond immediately

②



problem: app svc still there

state
- curr_step
- prev_step
- next_step
- request
- response