

14장 유튜브 설계

유튜브의 요구사항

- MAU :2억명
- 매일 재생되는 비디오수 : 50억
- 미국 성인 중 73%가 유튜브 이용
- 5천만명의 창장자
- 유튜브 광고 수입 15조원
- 모바일 인터넷 트래픽 37% 소요
- 80개 언어 이용

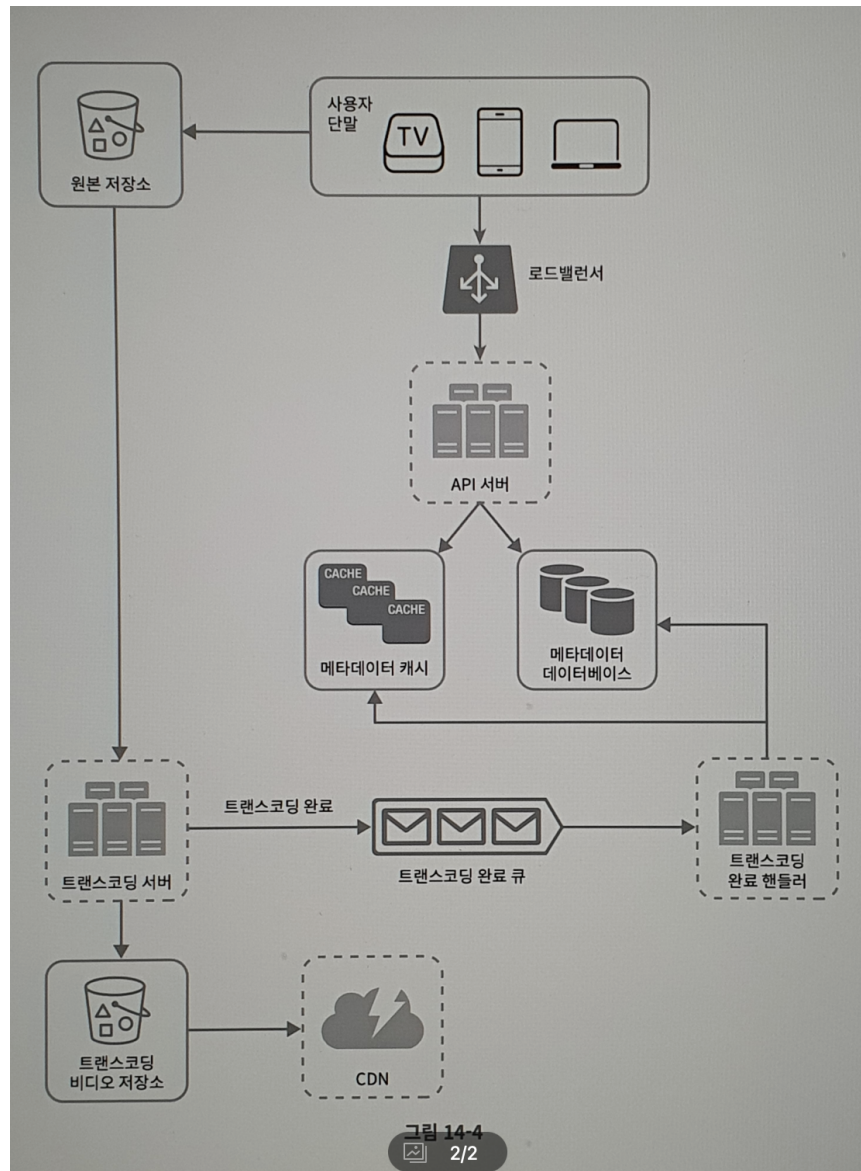
1단계 문제 이해 및 설계 범위

- 주요 기능 : 비디오 업로드, 시청
- 클라이언트 : 모바일 앱, 웹 브라우저, 스마트 TV
- DAU : 5백만
- 평균 사용 시간 : 30분
- 다국어
- 다양한 해상도
- 암호화
- 비디오 파일 크기 제한 : 1GB
- 클라우드 사용 O

규모 추정 : 매일 새로 요구되는 저장 용량 5백만 x 10% x 300MB \Rightarrow 150TB

CDN 비용 추정 : 5백만 * 5 비디오 * 0.3GB * 0.02달러 \rightarrow 150,000달러

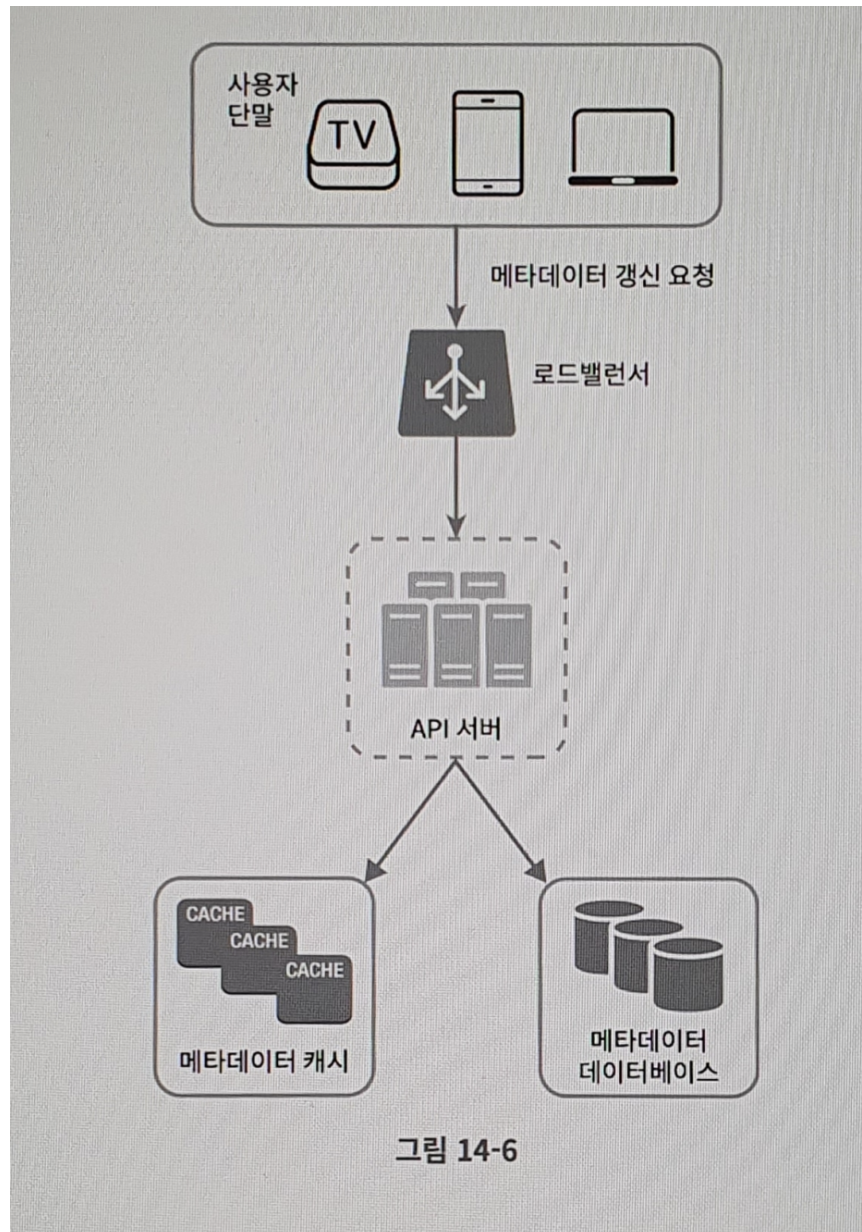
2단계 - 계략적인 설계안 제시 및 동의 구하기



비디오 업로드 절차

- 메타데이터 DB : 비디오의 메타데이터, 샤딩이나 다중화를 적용하여 가용성 및 성능 요구사항을 충족
- 메타데이터 캐시 : 성능을 높임
- 원본 저장소 : 원본 비디오 파일 저장소
- 트랜스코딩
 - 서버 : 적절한 비디오 포맷과 해상도로 변환하는 절차
 - 비디오 저장소 : 스트리밍 용 트랜스코드가 완료된 저장소

- 완료 큐와 핸들러 : 완료 이벤트를 받아서 보관, 갱신할 작업 서버들



비디오 스트리밍 프로토콜

- MPEG-DASH : Moving Picture Experts Group - Dynamic Adaptive Streaming over HTTP
- Apple Http Live Streaming
- MS Smooth Streaming
- Adobe HTTP Dynamic Streaming (HDS)

3단계 - 상세 설계

비디오 트랜스코딩

- 저장공간의 효율성과 클라이언트의 지원 포맷, 네트워크의 대역폭(클라이언트의 유동성) 등의 이유로 다양한 화질(bitrate)와 포맷으로 트랜스코딩해야 한다.
- 인코딩 포맷 = 컨테이너(.avi, .mp4)와 코덱(H.264, VP9) 으로 구성됨

DAG

- 비순환 그래프 : 작업의 워크플로를 비순환 그래프의 형태로 함
- 비디오 : 섬네일 추출, 인코딩, 워터마크, 오디오, 메타데이터를 병렬적으로 작업하여 병합하는 형태

비순환 트랜스코딩 아키텍처

- 전처리기 → DAG 스케줄러 → 자원관리자 → 작업 실행 서버 → 인코딩된 비디오
 - 작업 실행 서버와 전처리기는 임시저장소에 데이터를 보관한다.
 - 전처리기
 - 비디오 분할 (GOP) 몇 초 정도의 독립적으로 재생가능한 비디오로 분할
 - DAG 생성 : 개발자가 입력한 형식대로 DAG 워크플로를 생성함
 - GOP와 비디오 메타데이터를 temp storage에 보관한다. (작업 실패시 이를 이용하여 재시도)
 - DAG 스케줄러 : 오디오, 비디오, 메타데이터 추출 혹은 인코딩한다
 - 자원관리자 : 작업 큐와 스케줄러로 구성되며 적절함 워커 서버에 작업을 할당한다.
 - 작업 실행 서버 : 작업 서버다 .. 별 내용 없음
 - 임시저장소 : 메타데이터는 인메모리, 오디오/비디오 파일은 BLOB Storage에 저장해둔다.

시스템 최적화

- 속도 최적화
 - 비디오 파일을 한번에 올리는 것은 비효율적이므로 작은 GOP를 나눠 클라이언트에서 업로드할 수 있도록 구현한다.

- 업로드 센터를 근거리 지정 - CDN를 업로드 센터로 이용하여 제공할 수 있다.
- 모든 절차를 병렬화 : 하나의 GOP를 전체 절차를 밟고 다음 작업으로 넘어가는 것은 비효율적으로 파이프라인의 각 스텝별 사이에 메세지 큐를 뒤 동시적(병렬적)으로 작업을 할 수 있게 한다.
- 안정성 최적화
 - 미리 사인된 업로드 URL : API서버를 업로드 서버로 쓰기에는 그 용도로 적합하지 않다. 그러므로 미리 사인된 업로드 URL을 만들고 이를 클라이언트에서 직접 업로드하도록 변경한다.
 - 비디오 보호 : DRM 시스템(FairPlay, Widevine, PlayReady)를 사용한다. AES로 암호화 한다. 워터마크를 표기한다.
- 비용 최적화
 - 룬테일 분포를 따라서 인기 있는 동영상은 CDN에서 제공하고, 나머지는 비디오 서버를 통해 제공한다.
 - 특정 지역에만 인기있는 비디오들은 그 지역의 CDN에 옮긴다
 - ISP와 직접 제휴하여 CDN을 구성한다.
- 오류처리
 - 회복 가능 오류 : 트랜스 코딩에 실패한 경우 회복이 가능하다 하지만 재차 실패하는 경우 사용자에게 적절하게 오류코드를 반환해야한다.
 - 회복 불가능 오류 : 비디오 포맷이 잘못된 경우
 - 패턴
 - 업로드 오류 : 재시도
 - 비디오 분할 오류 : 비디오를 서버에 전송하고 서버가 GOP 생성
 - 트랜스코딩 오류 : 재시도
 - 전처리 오류 : DAG 재생성
 - DAG 스케줄러 오류 : 리스케줄링
 - 작업 관리자 큐에 장애 발생 : 사본(replica) 이용
 - 작업 서버 장애 : 다른 서버에 해당 작업을 재시도
 - API 서버 장애 :무상태로 구성
 - 메타데이터 캐시 서버 장애 : 다중화

- 메타데이터 DB : 클러스터 구성

마무리

1. API 계층 규모 확장 방안
2. 데이터베이스 규모 확장 방안 (샤딩, 다중화)
3. 라이브 스트리밍 : 응답 지연이 낮아야함, 작은 단위의 데이터를 실시간으로 빨리 처리해야함
4. 비디오 삭제 방안 : 신고와 미리 필터로 검사