

가상 면접 사례로 배우는
대규모 시스템 설계 기초

이정민

11. 뉴스 피드 시스템 설계

1단계) 문제 이해 및 설계 범위 확정

■ 요구사항 질의/응답

- 지원 대상? 모바일 앱, 웹
- 주요 기능? 뉴스피드 페이지에 새로운 스토리를 올리고 친구들이 올리는 스토리 조회 할 수 있어야함
- 스토리 표시 순서(최신, 토픽 점수, 친밀도)? 단순 시간 역순 흐름
- 한 명 당 최대 가질 수 있는 친구수는? 5,000명
- 트래픽 규모? 매일 천만 명 방문
- 이미지, 비디오 스토리 가능 여부? 스토리는 이미지나 비디오 포함 가능함

2단계) 개략적 설계안 제시 및 동의 구하기

■ 주요 기능

- 피드 발행 : 스토리를 포스팅하면 캐시와 데이터베이스에 기록하고 친구의 뉴스피드에 전송함
- 뉴스 피드 생성 : 모든 친구의 포스팅을 시간 흐름 역순으로 모아서 제공

2단계) 개략적 설계안 제시 및 동의 구하기

■ 뉴스 피드 API

- 피드 발행 API : POST /v1/me/feed - 새 스토리를 포스팅
 - body : 포스팅 내용
 - Authorization 헤더 : API 호출 인증
- 피드 읽기 API : GET /v1/me/feed - 뉴스 피드를 가져 옴
 - Authorization 헤더 : API 호출 인증

2단계) 개략적 설계안 제시 및 동의 구하기

■ 피드 발행

- 포스팅 저장 서비스 : 신규 포스팅을 데이터베이스와 캐시에 저장함
- 포스팅 전송 서비스 : 신규 포스팅을 친구 뉴스피드에 푸시함. 뉴스 피드 데이터는 캐시에 보관
- 알림 서비스 : 친구들에게 신규 포스팅이 게시되었음을 알리거나 푸시 알림을 전송

■ 뉴스 피드 생성

- 뉴스 피드 서비스 : 캐시에서 뉴스 피드를 가져옴
- 뉴스 피드 캐시 : 뉴스 피드를 렌더링할 때 필요한 피드 ID를 보관

3단계) 상세 설계

■ 피드 발행 흐름 상세 설계

- 웹 서버 : 인증 및 처리율 제한 등의 기능 수행 (스팸 및 유해 콘텐츠 방지 - 특정 기간 포스팅 수 제한)
- 포스팅 전송(팬아웃) 서비스 : 포스팅을 사용자의 친구에게 전달하는 과정
 - 쓰기 시점 팬아웃(fanout-on-write) 모델(push)
 - 읽기 시점 팬아웃(fanout-on-read) 모델(pull)

3단계) 상세 설계

■ 피드 발행 흐름 상세 설계 - 쓰기 시점 팬아웃 모델

- 신규 포스팅 기록 시점에 뉴스 피드 갱신. 포스팅 완료 시 해당 사용자의 캐시에 해당 포스팅 기록
- 장점
 - 뉴스 피드가 실시간으로 갱신되며 친구들에게 즉시 전송
 - 새 포스팅이 기록되는 순간에 뉴스 피드가 이미 갱신되므로 뉴스 피드를 읽는 시간이 짧아짐
- 단점
 - 친구가 많은 경우 친구 목록의 모든 사용자의 뉴스 피드를 갱신하는 데 많은 시간이 소요됨 (hotkey)
 - 서비스를 자주 이용하지 않는 사용자의 피드까지 갱신해야 하므로 컴퓨팅 자원이 낭비됨

3단계) 상세 설계

■ 피드 발행 흐름 상세 설계 - 읽기 시점 팬아웃 모델

- 요청기반 모델로서 피드를 읽는 시점에 뉴스피드를 갱신함. 사용자 로딩 시점에 포스트를 가져옴
- 장점
 - 비활성화된 사용자, 또는 서비스에 거의 로그인하지 않는 사용자의 경우 적합함 (로그인 전까지 자원 소모 없음)
 - 데이터를 친구 각각에 푸시하는 작업이 없어 핫키 문제가 생기지 않음
- 단점
 - 뉴스 피드를 읽는데 많은 시간이 소요될 수 있음

3단계) 상세 설계

■ 피드 발행 흐름 상세 설계 - 피드 발행 설계 방안

- 대부분의 사용자에게 대해서는 푸시 모델 사용
- 친구나 팔로어가 아주 많은 사용자의 경우에는 풀 모델 사용
- 안정 해시를 통해 요청과 데이터를 고르게 분산하여 핫키 문제를 줄임
- p. 191) 그림 11-5

3단계) 상세 설계

- 피드 읽기 흐름 상세 설계

- p. 193) 그림 11-7

3단계) 상세 설계

■ 캐시 구조

- 뉴스 피드 : 뉴스 피드의 ID를 보관
- 콘텐츠 : 포스팅 데이터 보관. 인기 데이터는 별도 보관
- 소셜 그래프 : 사용자 간 관계 정보를 보관
- 행동(action) : 포스팅에 대한 사용자의 행위에 관한 정보를 보관. 포스팅에 대한 좋아요, 답글 등
- 횟수(counter) : 좋아요 횟수, 응답 수, 팔로어 수, 팔로잉 수 등의 정보를 보관

4단계) 마무리

■ 추가 논의사항

- 데이터베이스 규모 확장
 - 수직적 규모 확장 vs 수평적 규모 확장, SQL vs NoSQL, 주-부(master-slave) 다중화
 - 복제본(replica)에 대한 읽기 연산, 일관성 모델(consistency model), 데이터베이스 샤딩(sharding)
- 웹 계층(web tier)을 무상태로 운영하기
- 가능한 한 많은 데이터를 캐시할 방법
- 여러 데이터 센터를 지원할 방법
- 메시지 큐를 사용하여 컴포넌트 사이의 결합도 낮추기
- 핵심 메트릭(key metric)에 대한 모니터링, 피크타임 QPS, 새로고침 시 지연시간 등