

8주차

뉴스 피드 시스템 설계

뉴스피드를 설계할 때 고려할 점

- 웹? 앱? 둘다 지원 해야하는 지?
- 가장 중요한 기능으로는?
- 새로운 스토리를 올릴 수 있어야하고 친구들이 올리는 스토리도 볼 수 있어야함
- 순서는?
- 최대 몇명의 친구를 가질것인지?
- 트래픽 규모는?

피드발행

- 사용자가 스토리를 포스팅하면 해당 데이터를 캐시와 데이터 베이스에 기록

뉴스피드 생성

- 지면 관계 상 뉴스피드는 모든 친구의 포스팅을 시간흐름 역순으로 모아서 만든다

뉴스피드 API

- 클라이언트가 서버와 통신하기 위해 사용하는 수단이다. HTTP 프로토콜 기반이고, 상태 정보를 업데이트하거나, 뉴스피드를 가져오거나, 친구를 추가하는 등의 다양한 작업을 수행하는 데 사용한다. 지금부터 이 가운데 가장 중요한 두가지를 배울 것이다.

피드발행 API

- POST/v1/me/feed
- 인자 body : 포스팅 내용, Authorization 헤더 : API 호출을 인증하기 위해 사용한다

뉴스피드 API

- Authorization 헤더 : API 호출을 하기 위해 사용한다

피드발행

- 사용자 : 모바일 앱이나 브라우저에서 새 포스팅을 올리는 주체
- 로드밸런서 : 트래픽을 웹 서버들로 분산
- 웹 서버 : HTTP 요청을 내부 서비스로 중계하는 역할을 담당한다
- 포스팅 저장 서비스 : 새 포스팅을 데이터베이스와 캐시에 저장한다
- 포스팅 전송 서비스 : 새 포스팅을 친구의 뉴스피드에 푸시한다. 뉴스 피드 데이터는 캐시에 보관하여 빠르게 읽어갈 수 있도록 한다
- 알림서비스 : 친구들에게 새포스팅이 올라 왔음을 알려거나 푸시알림을 보내는 역할을 담당한다

뉴스피드 생성

- 사용자 : 뉴스피드를 읽는 주체
- 로드밸런서 : 트래픽을 웹서버들로 분산한다
- 웹 서버: 트래픽을 뉴스피드 서비스로 보낸다
- 뉴스피드 서비스 : 캐시에서 뉴스피드를 가져오는 서비스이다
- 뉴스 피드 캐시 : 뉴스 피드를 렌더링할 때 필요한 피드 ID를 보관한다.

웹서버

- 클라이언트와 통신
- 인증 Authorization 헤더에 토큰을 포함하는 사용자만 포스팅가능
- 처리율제한
- 스팸 차단, 유해한 콘텐츠가 자주 올라오는 것을 방지

팬아웃 서비스

- 어떤 사용자의 새 포스팅을 그 사용자와 친구 관계에 있는 모든 사용자에게 전달하는 과정
- 쓰기 시점 팬아웃
- 읽기 시점 팬아웃

쓰기 시점 팬아웃

- 새로운 포스팅을 기록하는 시점에 뉴스피드를 갱신, 포스팅이 완료되면 해당 사용자의 캐시에 해당 포스팅이 기록된다
- 장점 : 뉴스피드가 실시간으로 갱신되고 친구 목록에 있는 사용자에게 즉시 전송된다, 새 포스팅이 기록되는 순간에 뉴스피드가 이미 갱신되므로, 뉴스피드를 읽는 데 드는 시간이 짧아 진다.
- 단점 : 친구가 많은 사용자의 경우 친구목록을 가져오고 그 목록에 있는 사용자 모두의 뉴스 피드를 갱신하는 데 많은 시간이 소요 될 수 있다, 서비스를 자주 이용하지 않는 사용자의 피드까지 갱신해야하므로 컴퓨팅 자원이 낭비가 된다

읽기 시점 팬아웃

- 피드를 읽어야하는 시점에 갱신된다 , 온디맨드 형식
- 사용자가 본인 홈페이지나 타임라인은 로딩하는 시점에 새로운 포스트를 가져오게 됨
- 장점 : 비활성화된 사용자, 또는 서비스에 거의 로그인하지 않는 사용자의 경우에는 이 모델이 유리함, 로그인 전까지는 어떤 컴퓨팅 자원도 소모하지 않음, 데이터를 친구 각각에 푸시하는 작업이 필요없으므로 핫키 문제가 발생하지 않는다
- 단점 : 뉴스피드를 읽는데 많은 시간이 소요된다

뉴스피드를 빠르게 가져오는 것은 중요하고 팔로워 수가 많을 때는 사용자의 포스팅을 필요할 때 가져오도록 읽기시점 팬아웃 모델을 사용한다

피드읽기 흐름 상세 설계

1. 사용자가 뉴스피드를 읽으려고 요청
2. 로드밸런서가 요청을 웹 서버 가운데 하나로 로드 밸런싱을 진행
3. 웹 서버는 뉴스 피드 서비스 호출
4. 뉴스 피드 서비스는 뉴스피드 캐시에서 ID 목록을 획득
5. 뉴스피드에서 표시할 사용자이름, 사용자 사진, 이미지 , 콘텐츠등을 사용자 캐시와 포스팅 캐시에서 가져와 안전한 뉴스피드를 만든다
6. 생성된 뉴스피드를 JSON 형태로 클라이언트에게 전송, 클라이언트는 해당 피드를 렌더링함