

# 11장

## 뉴스 피드 시스템 설계

- 뉴스피드란?
  - 지속적으로 업데이트 되는 정보들을 모아둔 것

### 1단계 문제 이해 및 설계 범위 확정

---

1. 뉴스 피드 페이지에는 새로운 스토리를 올릴 수 있어야 한다.
2. 친구들이 올리는 스토리를 볼 수 있어야 한다.
3. 포스트는 시간 흐름 역순으로 정렬된다.
4. 한 명의 사용자는 최대 5,000명의 친구를 가질 수 있다.
5. 어플리케이션은 매일 천만 명이 방문한다.
6. 피드에 이미지나 비디오 스토리도 올릴 수 있다.

### 2단계 매력적 설계안 제시 및 동의 구하기

---

#### 뉴스 피드 API

1. 클라 - 서버 간 통신을 위해 사용하는 수단.
2. HTTP 프로토콜 기반이고 상태 정보를 업데이트 하거나, 뉴스 피드를 가져오거나, 친구를 추가하는 등의 작업에 활용됨.
  - 피드 발행 API  
새 스토리를 포스팅하기 위한 API이며 HTTP POST 형태로 요청을 보낸다.
  - 피드 읽기 API  
뉴스 피드를 가져오는 API

## 피드 발행

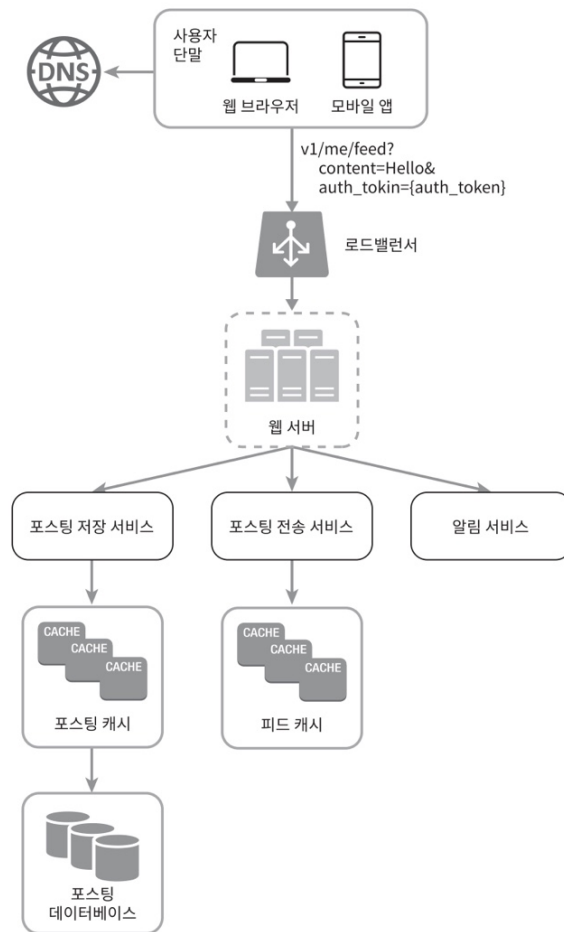


그림 11-2

## 피드 생성

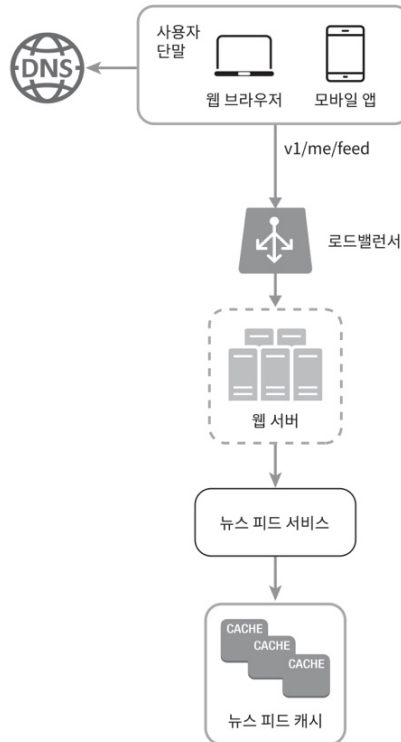


그림 11-3

## 3단계 상세 설계

### 피드 발행 흐름 상세 설계

- 웹 서버
  - 클라 - 서버 간 통신 뿐 아니라 인증, 처리율 제한 등의 기능도 수행함
  - 특정 기간 동안 한 사용자가 올릴 수 있는 포스팅의 수에 제한을 둬
- 포스팅 전송(팬아웃) 서비스
  - 어떤 사용자의 새 포스팅을 그 사용자와 친구 관계에 있는 모든 사용자에게 전달하는 과정
    - 쓰기 시점에 팬아웃하는 모델
      - 포스팅이 완료되면 곧바로 해당 사용자의 캐시에 포스팅을 기록한다.
        - 장점

1. 뉴스 피드가 실시간으로 갱신되며 친구 목록에 있는 사용자에게 즉시 전송
2. 뉴스 피드를 읽는 데 드는 시간이 줄어들

- 단점

1. 친구가 많은 사용자의 경우 친구 목록을 가져오고, 목록에 있는 사용자들의 뉴스 피드를 갱신하는 데에 많은 시간이 소비 됨 (핫키hotkey 문제)
2. 서비스를 자주 이용하지 않는 사용자의 피드까지 갱신해야 함

- 읽기 시점에 팬아웃하는 모델

피드를 읽는 시점에 뉴스 피드를 갱신한다.

- 장점

1. 비활성화된 사용자 혹은 서비스를 거의 이용하지 않는 사용자의 경우 로그인 시점 이전까지는 어떤 자원도 낭비하지 않음
2. 데이터를 친구 각각에 푸시하는 작업이 필요 없으므로 핫키hotkey 문제가 발생하지 않음

- 단점

1. 뉴스 피드를 읽는 데에 많은 시간이 걸림

- 보완

두 방법의 장점만 취해서 구조를 설계한다.

- 대부분의 사용자에게는 푸시 모델을 사용한다.
- 친구나 팔로어가 많은 경우 해당 사용자의 뉴스 피드를 읽는 경우 풀 모델을 사용하도록 한다.
- 안정 해시를 통해 요청과 데이터를 고르게 분산한다.

- 팬아웃 서비스의 동작 과정

1. 그래프 데이터베이스에서 친구 ID 목록을 가져온다.
2. 사용자 설정에 따라 데이터 걸러낸다.

3. 친구 목록과 새 스토리의 포스팅 ID를 메시지 큐에 넣는다.
4. 팬아웃 작업 서버가 메시지 큐에서 데이터를 꺼내어 뉴스 피드 데이터를 뉴스 피드 캐시에 넣는다. 이 때 메모리 정보량을 줄이기 위해 캐시에 포스팅 정보 중 <포스팅 ID, 사용자 ID>를 매핑하여 저장한다.

## 피드 읽기 흐름 상세 설계

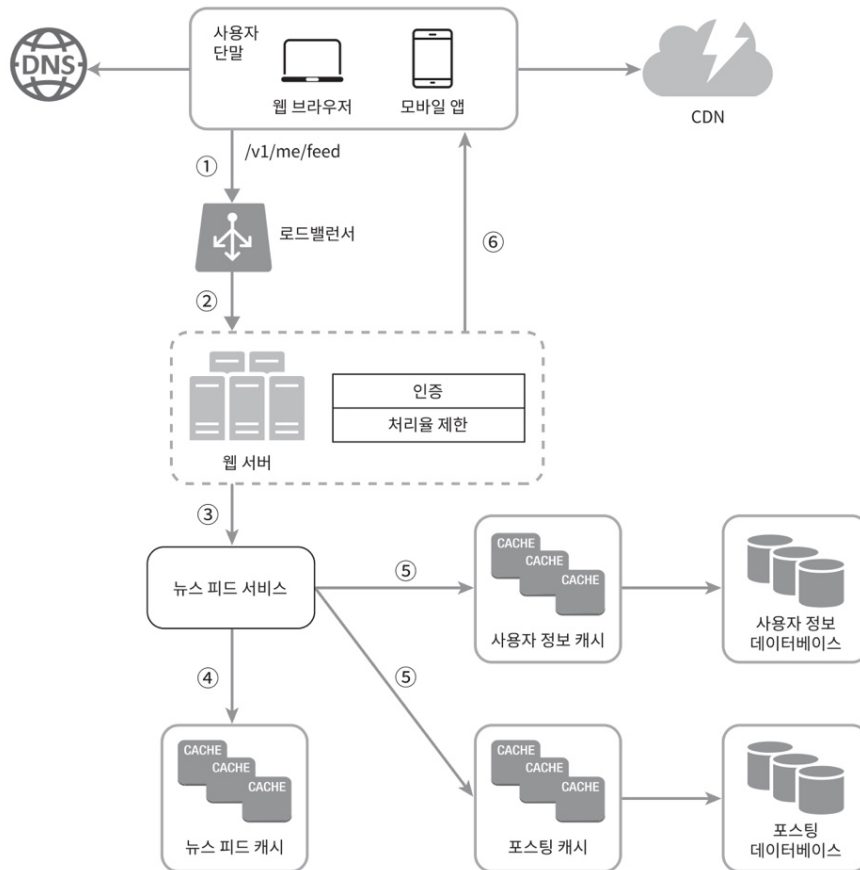


그림 11-7

## 캐시 구조



그림 11-8

## 4단계 마무리

- 논의하면 좋은 것
  - 데이터베이스 규모 확장
  - 웹 계층을 무상태로 운영하기
  - 가능한 한 많은 데이터를 캐시할 방법
  - 여러 데이터 센터를 지원할 방법
  - 메시지 큐를 사용하여 컴포넌트 사이의 결합도 낮추기
  - 핵심 메트릭에 대한 모니터링