10주차

13장 검색어 자동 완성 시스템

1단계 : 문제 이해 및 설계 범위 설정

빠른 응답 속도

- 사용자가 검색어를 입력함에 따라 자동완성 검색어도 충분히 빨리 표시되어야한다. 페이스북 검색어 자동완성 시스템에 관한 문서에 따르면 시스템 응답속도는 100밀리초 이내여야 한다.

연관성

- 자동완성 검색어는 사용자가 입력한 단어와 연관된 것이어야함

정렬

- 시스템의 계산 결과는 인기도 등의 순위 모델에 의해 정렬되어야함

고가용성

- 시스템의 일부에 장애가 발생하거나, 느려지거나, 예상치 못한 네트워크 문제가 발생해도 시스템은 계속 사용 가능해야 한다

개략적인 규모 추정

- 일간 능동 사용자는 천만명으로 가정
- 평균적으로 한 사용자는 매일 10건의 검색을 수행
- 질의 할때 마다 평균적으로 20바이트의 데이터를 입력한다고 가정
    - 문자 인코딩 방법으로는 ASCII를 사용하고 1문자이다.
    - 질의문의 평균 4글자하고 한 글자는 5문자로 구성, 질의는 평균 20Byte값을 갖음
- 검색창에 글자를 입력할 때마다 클라이언트는 검색어 자동완성 백엔드에 요청을 보낸다
- 대략 초당 24000건의 질의가 발생할 것이다

2단계 : 개략적 설계안 제시 및 동의 구하기

- 데이터 수집 서비스
  - 사용자가 입력한 질의를 실시간으로 수집하는 시스템이다.
- 질의 서비스
  - 주어진 질의에 다섯개의 인기검색어를 정렬해 놓는 서비스이다
  - query : 질의문을 저장하는 필드
  - frequency : 질의문이 사용된 빈도를 저장하는 필드

3단계 : 상세 설계

트라이구조

- 데이터 베이스를 이용해 가장 인기 있었던 5개의 질의문을 골라내는 방법은 효율적이지 않다. 이문제는 트라이를 사용하여 해결하는 데 트라이는 문자열들을 간략하게 저장할 수 있고 문자열을 꺼내는 연산에 초점을 맞추어 설계된 자료구조이다.
- 트리형태의 자료구조이다
- 이 트리의 루트노드는 빈 문자열을 나타낸다
- 각 노드는 글자를 하나 저장하여 26개의 자식노드를 가질 수 있디
- 각 트리 노드는 하나의 단어, 또는 접두어 문자열을 나타낸다

데이터 수집 서비스

- 사용자가 타이핑을 하면 실시간으로 데이터를 수정했는데 이때 발생하는 문제점은 다음과 같다
- 매일 수천만 건의 질의가 입력될텐데 그때마다 트라이를 갱신하면 질의 서비스는 심각하게 느려질 것이다
- 일단 트라이가 만들어지고나면 인기검색어는 그다지 바뀌지 않을 것이다. 트라이는 갱신을 적게해도된다
- 위에서 눈치챘겠지만 데이터 일정 수준 쌓이면 인기 검색어는 그다지 자주 바뀌지 않으므로 트라이는 자주 갱신할 필요가 없다.
  - 물론 실시간성이 중요한 서비스마다 즉 요구사항에 따라 달라질 순 있겠다.
- 데이터 수집 방식

    - 테이터를 수집하는 방식은 여러가지겠지만, 결론적으로 그 데이터를 잘 취합하는 것이 중요하다.
- 트라이 캐시와 영속성 보장
    - 트라이를 캐시하여 분산 시스템을 구성하여 성능을 높이고
    - 영속성을 위해 DB에 저장해두는 것도 생각볼 수 있다.
    - 여기서 document store를 활용하면 성능면에서 적합할 수 있겠다.
    - key-value store도 다음과 같이 활용해볼 수 있겠다.
        - 트라이에 보관된 모든 접두어를 해시 테이블 키로 변환
        - 각 트라이 노드에 보관된 모든 데이터를 해시 테이블 값으로 변환

질의 서비스

이제 위에서 설계한 시스템을 기반으로 질의 서비스를 만들면 된다.

자세한 내용은 생략하고 성능면에서 어떤 것들을 개선할 수 있을지만 살펴보자

- 트라이 캐시
- 브라우저 캐싱
- 데이터 샘플링

검색어 삭제(필터링)

비속어가 포함된 질의어는 자동완성 결과에서 제외해야한다.

이럴 때 트라이 캐시 앞 단에 filter layer를 별개로 두어 부적절한 질의어를 반환하지 않게 조정해보자

layer를 별도로 주면 필터 규칙을 수정하기 편리하고 먼저 필터 규칙으로 일시적으로 사용자에게 노출하지 않고, 실제 DB에서 물리적으로 검색어를 삭제하는 것은 비동기적으로 진행하면 된다.