

검색어 자동완성 시스템

보통 검색어 자동완성(autocomplete, typeahead, search-as-you-type, incremental search)
가장 많이 이용된 검색어 k개를 자동완성하여 출력하는 시스템 설계이다.

1단계 문제 이해 및 설계 범위 확정

역시 시작은 질문으로 시작한다.

Q: 자동완성될 검색어의 첫 부분이어야 하나요? 아니면 중간 부분?

W: 첫 부분으로 한정

Q: 몇 개의 자동완성 검색어가 표시되어야 하나요?

W: 5개

Q: 5개를 고르는 기준은 무엇인가요?

W: 질의 빈도에 따라 정해지는 검색어 인기 순위를 기준

Q: 맞춤법 검사 기능도 제공?

W: 아뇨, 지원하지 않습니다. 다만 다국어 지원 고려

Q: 얼마나 많은 사용자를 지원해야 합니까?

W: 일간 능동 사용자(DAU) 천만(10million)명

요구사항을 정리하면

- **빠른 응답 속도**
- **연관성**
 - 자동완성되어 출력되는 검색어는 사용자가 입력한 단어와 연관된 것
- **정렬**
 - 시스템의 계산 결과는 인기도 등의 순위 모델에 의해 정렬
- **규모 확장성**
 - 시스템은 많은 트래픽을 감당할 수 있도록 확장 가능
- **고가용성**
 - 시스템의 일부는 장애가 발생하거나, 느려지거나, 예상치 못한 네트워크 문제가 생겨도 시스템은 계속 사용 가능

규모 추정하면 다음과 같다.

- 일간 능동 사용자(DAU)는 천만명
- 평균적으로 한 사용자는 매일 10건의 검색 수행
- 질의할 때마다 평균적으로 20바이트의 데이터를 입력
- 검색창에 글자를 입력할 때마다 클라이언트는 검색어 자동완성 백엔드에 요청을 보낸다. 따라서 평균적으로 1회 검색당 20건의 요청이 백엔드로 전달.
- 대략 초당 24,000건의 질의 발생($10,000,000 * 10 * 20 / 24 / 3600$)
- 최대 QPS = QPS * 2 = 48,000

2단계 개략적 설계안 제시 및 동의 구하기

개략적 시스템 설계는 2가지로 나눌수 있다.

데이터 수집 서비스(data gathering service)

사용자가 입력한 질의를 실시간으로 수집하는 시스템

질의 서비스(query service)

주어진 질의에 다섯 개의 인기 검색어를 정렬해 내놓는 서비스

아래는 빈도 테이블을 표현한다.

질의	빈도
twitch	1
twitter	2

빈도 테이블을 만드는 행위

= 데이터 수집 서비스

만들어진 빈도 테이블을 활용해 Top5 제공하는 행위

= 질의 서비스

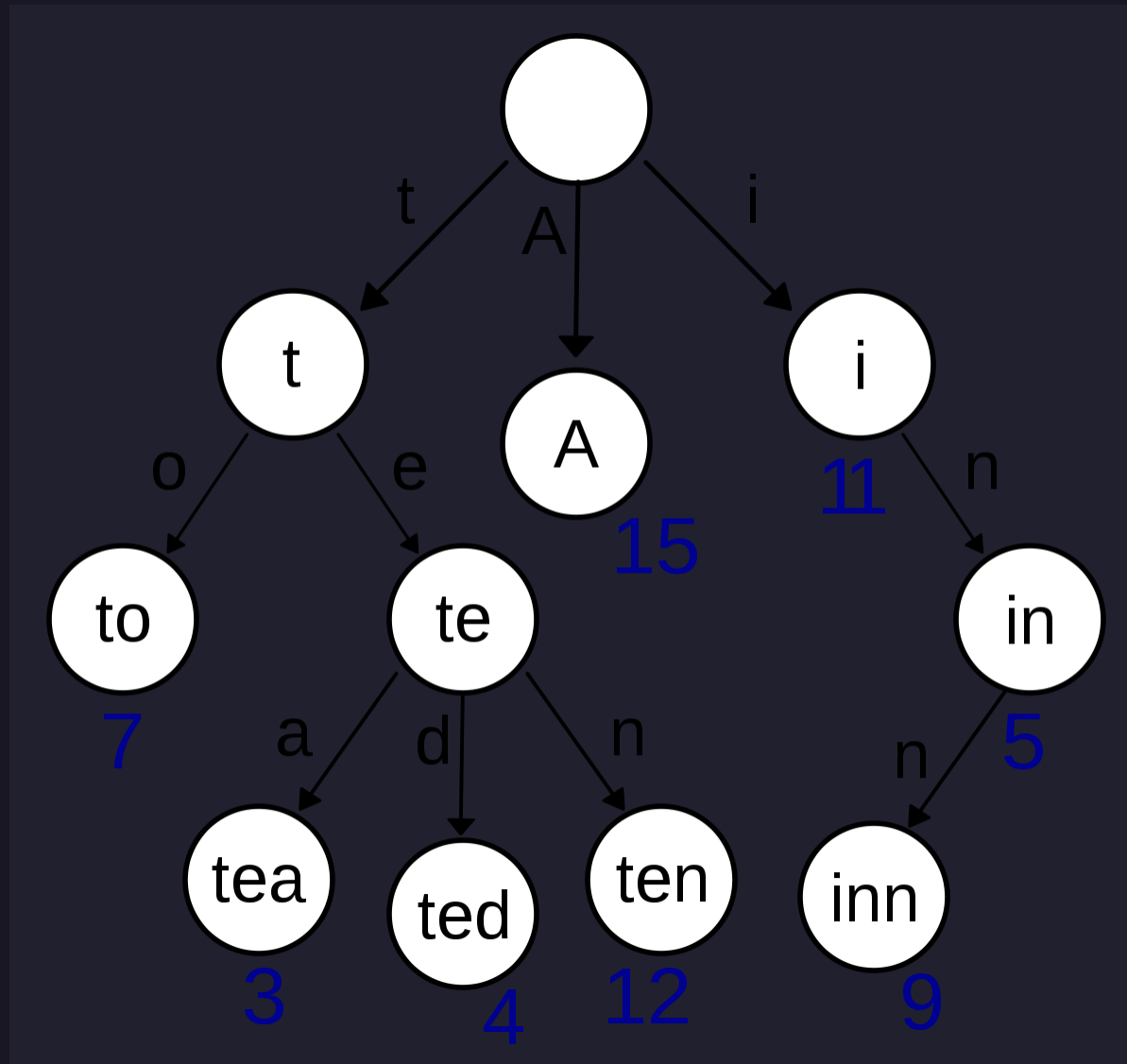
3단계 상세 설계

이야기 하고 싶은 컴포넌트는 다음과 같다.

- 트라이(trie) 자료구조
- 데이터 수집 서비스
- 질의 서비스
- 규모 확장이 가능한 저장소
- 트라이 연산

트라이 자료구조

- 트라이는 트리 형태의 자료구조
- 이 트리는 루트 노드는 빈 문자열을 나타냄
- 각 노드는 글자(character) 하나를 저장, 26개의 자식 노드
- 각 트리 노드는 하나의 단어, 또는 접두어 문자열(prefix string)



시간복잡도는 다음과 같다.

- 해당 접두어를 표현하는 노드 찾기 시간복잡도 - $O(p)$
- 해당 노드부터 시작하는 하위 트리를 탐색해 모든 유효 노드 찾기 - $O(c)$
- 가장 인기 있는 검색어 K개 찾기 - $O(c \log c)$

만약 최악의 경우

모두 탐색해야 한다. 이럴때는 접두어 최대 길이 제한 또는 각 노드에 인기 검색어 제시

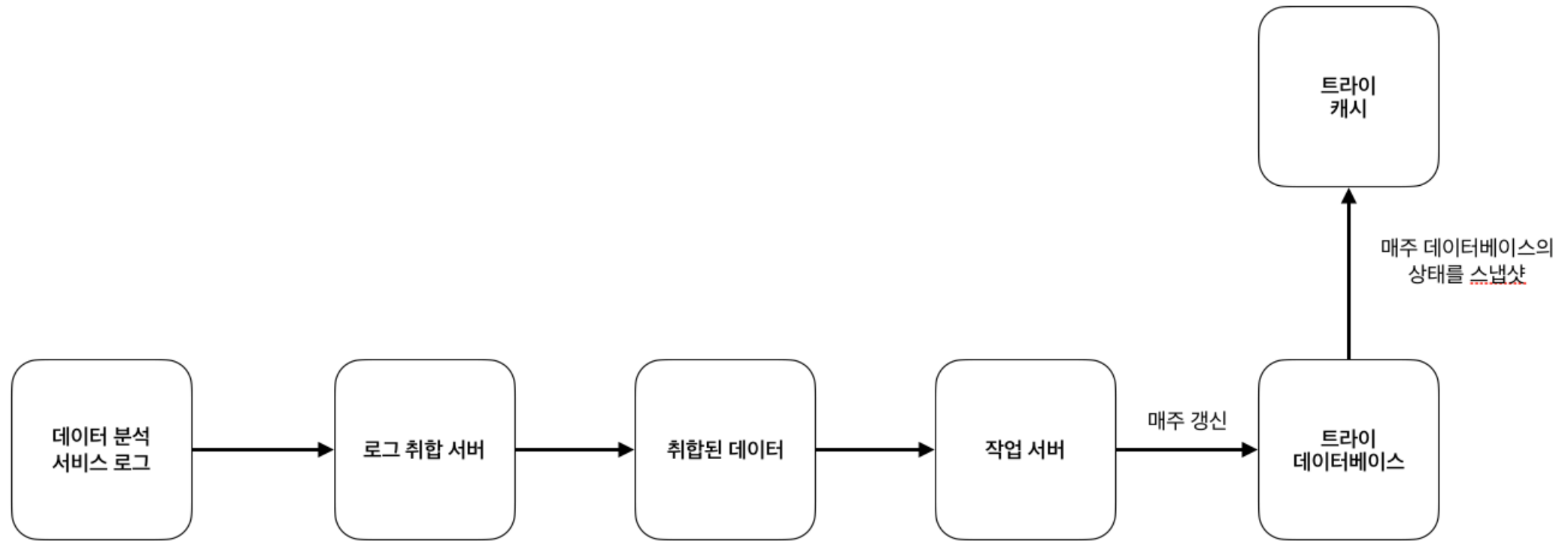
데이터 수집 서비스

타이핑을 할 때마다 실시간으로 데이터 수정하는 API 를 요청하게 되는데, 실용적이지 못함

- 매일 수천만 건의 질의가 입력될 텐데, 그때마다 트라이를 갱신하면 질의 서비스는 심각하게 느려질 것.
- 일단 트라이가 만들어지고 나면 인기 검색어는 그다지 자주 바뀌지 않을 것. 그러니 트라이는 그렇게 자주 갱신할 필요가 없다.

그래서 이 문제를 어떻게 해결할 수 있을까?

해결하기 위한 간단한 설계안은 다음과 같다.



각각의 컴포넌트를 간단히 살펴보면 다음과 같다.

데이터 분석 서비스 로그

검색창에 입력된 질의에 관한 원본 데이터를 보관. 오직 추가될 뿐 수정은 이루어지지 않음

로그 취합 서버

용례에 따라 다르지만. 로그를 취합하여 빈도(frequency) 필드 표현

작업 서버

주기적으로 비동기적 작업(job)을 실행하는 서버 집합. 트라이 자료구조를 만들고 트라이 DB에 저장하는 역할 담당

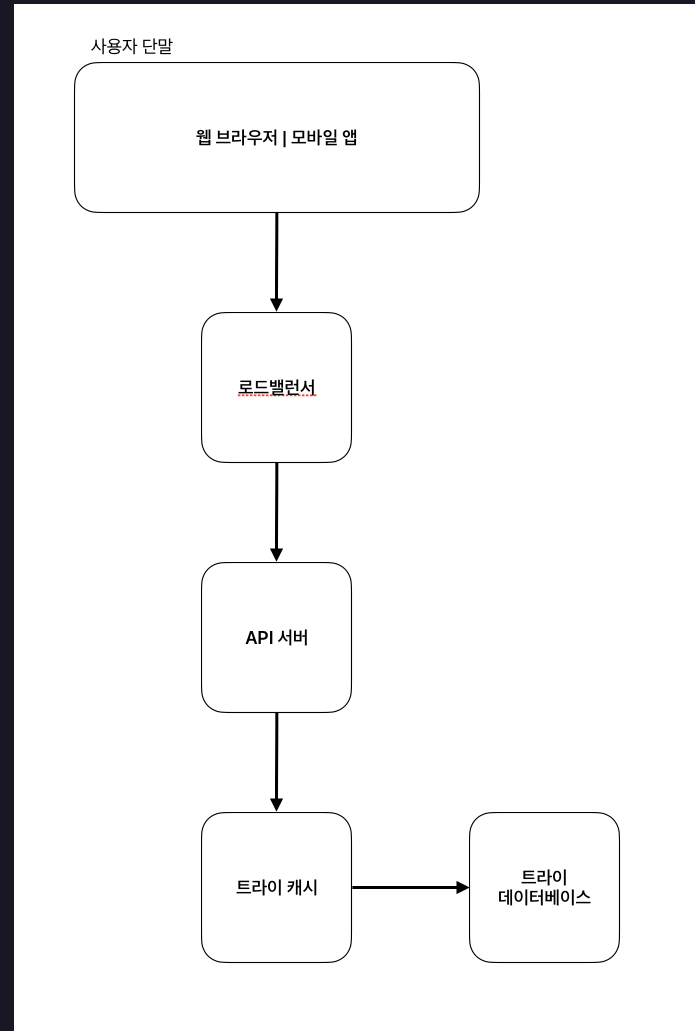
트라이 캐시

분산 캐시 시스템으로 트라이 데이터를 메모리에 유지하여 읽기 연산 성능을 높이는 구실

트라이 데이터베이스

몽고디비와 같은 문서 저장소 / 키-값 저장소 활용하여 구현할 수 있다.

질의 서비스



순서

1. 검색 질의가 로드밸런서로 전송
2. 로드밸런서는 해당 질의를 API 서버로 보낸다.
3. API 서버는 트라이 캐시에서 데이터를 가져와 해당 요청에 대한 자동완성 검색어 제안 응답을 구성
4. 데이터가 트라이 캐시에 없는 경우에는 데이터를 데이터베이스에서 가져와 캐시를 채운다.

여기서 질의 서비스가 좀더 빨라지기 위한 개선안으로

1. AJAX 요청

2. 브라우저 캐싱

- cache-control 헤더 값에 private는 해당 응답이 요청을 보낸다. 로컬 캐시!
- max-age=3600 - 한시간동안 유효.

3. 데이터 샘플링

- 모든 질의 결과를 로깅하도록 해 놓으면 CPU 자원과 저장공간을 엄청나게 소진하게 된다.
이때 데이터 샘플링으로 N개 요청 중 1개만 로깅하도록 한다.

트라이 연산

트라이는 검색어 자동완성 시스템의 핵심 컴포넌트

4단계 마무리

상세 설계 후 면접관이 물어볼 수 있다?

Q. 다국어 지원이 가능하도록 시스템을 확장하려면 어떻게 해야 될까?

Q. 국가별로 인기 검색어 순위가 다르다면 어떻게 해야 할까?

Q. 실시간으로 변하는 검색어의 추이를 반영하려면 어떻게 해야 할까?