2장. 자료 색인

일래스틱 서치 색인

-정의 : 일렉스틱 서치가 논리적인 자료를 저장하는 논리 공간. 여러 세그먼트로 나누어질수 있음

노드:색인= M:N, 색인:샤드=1:N

-샤드, 레플리카: 색인은 샤드 여러개로 구성, 각각은 다큐먼트 집합의 일부를 포함.

물리적 노드가 여러개 일수록 샤드,레플리카의 배치를 나눌경우 효율적.

레플리카가 많을수록 결함 허용률이 좋아짐

샤드는 운용 중에 제한적인 증가 가능, 레플리카는 운용중에서 동적으로 조정 가능.

-색인 생성 : 색인이 존재하지 않을 경우 자동생성.

수동생성이 필요한 경우는 색인 구조나 샤드숫자 변경과 같은 추가설정이 필요할 경우.

PUT, DELETE 사용을 통해 색인 구성 가능.

매핑구성

-정의 : 다큐먼트의 각 필드는 타입에 따라 분석되며, 이러한 분석된 정보를 매핑이라 함.

-핵심타입

문자열: string

속성: term_vector, omit_norms, analyzer, index_analyzer, search_analyzer, norms.enalbed, norms.loading, position_offset_gap, index_options, ingnore_above

숫자 : byte, short, integer, long, float, double

속성: precision_step, ignore_malformed

날짜 : date

속성: format, precision_step

매핑구성

- 핵심타입

  부울 : bool

  바이너리 : binary

  복수필드 : 필드 정의에 fields 객체를 추가하여 구성

  IP : ip

    속성 : precision_step

  token_count : token_count

    속성: 숫자타입과 동일 + anlayzer

공통속성: index_name, index, store, boost, null_value, copy_t o, include_all

매핑구성

- 타입 결정 메커니즘 : 일래스틱서치는 기본적으로 다큐먼트를
정의하는 JSON을 살펴 다큐먼트 구조를 추측

필드 타입 추측 활성화/비활성화 : dynamic : true,false

숫자 타입 : numeric_detection : true,false

날짜 타입 : dynamic_date_format : ["yyyy-MM-dd hh:mm"] ...

매핑 구성

-분석기 사용 : 일래스틱서치는 색인시점과 질의시점에 다양한
분석기를 사용가능.

분석기 구조는 토큰 추출기 와 여러 필터로 구성.
(settings 구문을 통해 default 및 사용자 정의가능)

ex) standard, simple, whitespace, stop, keyword, pattern ,language, snowball

-색인 구조 매핑 : 여러 타입으로 색인 구조 정의 가능

매핑구성

-다양한 유사성 모델 : 기본적으로 질의와 일치하는 도큐먼트를 찾아 점수를 부여. 점수가 높을수록 다큐먼트 관점에서 관련성이 높음. 결국 다큐먼트 내용에서 df가 높고, idf가 작을수록 다큐먼트에 높은 점수를 줌

-TF/IDF 기반 알고리즘 지원:
(Okapi BM25, DFR(Divergence from randomness), IB (Information-based)

TF/IDF

- Tf-idf(term frequency-inverse document frequency) 가중치는 언어 자료 내의 특정 문서에서 어떤 단어의 중요도를 평가하기 위해 사용되는 통계적인 수치

-term frequency(단어 빈도)는 단순히 그 문서에서 해당 단어가 나타나는 횟수

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

- Inverse document frequency(역 문서 빈도)는 해당 단어의 일반적인 중요도를 나타내는 수치. 전체 문서의 수를 해당 단어가 포함된 문서들의 수로 나눈 값에 로그를 씌움.

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

TF/IDF

예제)

만약 100개의 단어로 이루어진 어떤 문서에 단어 cow가 세 번 등장한다면, 단어 cow의 term frequency는 0.03(3/100).

만약, 전체 10,000,000개의 문서 중에서 단어 cow가 들어있는 문서들의 숫자가 1,000개라면, idf 값은 ln(10,000,000/1,000) = 9.21이 되어서

최종 tf-idf 가중치는 0.27(0.03 * 9.21).

매핑구성

-출현 목록 형식 : 색인 파일의 기록방법을 변경 가능하므로 필드별 Postings 형식을 명세 가능.

default, pulsing, direct, memory, bloom_default, bloom_pulsing

-doc : 정렬과 패싯을 효과적으로 수행하여 메모리를 효율적 사용하는 컬럼 확장구조로 기록되는 필드

색인과정에서 속도를 높이기 위한 배치 색인

-정의: 하나씩 색인하는 대신 여러 다큐먼트를 색인하는 방법

-대량 색인을 위한 자료 준비 : 여러 요청을 패킷 하나로 병합이 가능하고 단일요청으로 전송이 가능함. 단. 대량색인의 파일크기는 제한이 있으며 http.max_content_length 를 명세함으로써 조정이 가능.

1. 색인에 이미 존재하는 다큐먼트를 추가하고 대체하기 (index)

2. 색인에서 다큐먼트를 삭제하기 (delete)

3. 색인에 다큐먼트의 다른 정의가 없을 경우 새로운 다큐먼트를 추가하기 (create)

-배타를 대량 요청

색인을 좀 더 빠르게 하고 싶다면 UDP를 사용하여 연산처리. 단 자료가 유실될 수 있음.

추가적인 내부 정보로 색인 구조 확장

-정의: 자료를 담기 위해 사용된 필드뿐만 아니라 자료처리 편의성을 위한 추가정보도 다큐먼트에 저장할 수 있음.

추가필드타입:

_uid : 다큐먼트의 식별자와 다큐먼트 타입을 결합한 식별자, _id : 색인과정에서 설정된 식별자

_type : 다큐먼트 타입

_all : 검색의 편의를 위해 다른 모든 필드에 등장하는 자료를 모아 단일 필드에 저장.

_source : 색인과정에서 일래스틱서치에 전송한 원본 JSON 다큐먼트를 저장. (includes, excludes)

_index : 다큐먼트가 저장될 색인에 대한 정보

_size : _source 필드의 원래 크기를 자동으로 색인하고 저장하게 함.

_timestamp : 다큐먼트가 색인된 시점을 저장.

_ttl : Time to Live로 다큐먼트의 생명 주기를 정의.

세그먼트 병합

-정의: 색인은 여러개의 조각(세그먼트)로 나누어 질수 있음. 이러한 세그먼트는 불변성을 가짐
실제 다큐먼트가 삭제되더라도 색인은 남아있고, 세그먼트 병합 발생 시 이를 처리함.

-세그먼트 병합 : 루씬 라이브러리가 여러 세그먼트를 가져와 여기서 찾은 정보를 토대로 새로운 세그먼트를 생성하는 과정. 병합 후 기존 세그먼트들과 삭제를 위해 표시된 다큐먼트가 실제 삭제됨.

-병합 정책 및 방법 : 색인을 생성하는 세그먼트가 많을수록 검색이 느려지고, 사용되는 리소스가 많아지므로 적절한 병합이 필요함. 그러나 병합시 서버 리소스를 소모하므로 적절한 병합속도 조절이 필요.

병합정책: index.merge.policy (tierd, log_byte_size, log_doc)
병합스케줄러: index.merge_scheduler.type (concurrent, serial)
병합속도조절: indices.store.throttle.type (none, merge, all)
indices.store.throttle.max_bytes_per_sec (xx mb)

라우팅 개괄

-정의: 자료를 색인하거나 검색하기 위해 사용할 샤드를 사용자에게 선택하게 지원하는 기능.

-기본색인 : 다큐먼트 식별자의 해시값을 계산해, 이를 기반으로 가용 샤드 중 한곳에 다큐먼트를 저장.

-기본검색 : 사용자가 노드 중 하나에 질의를 하면, 일래스틱 서치는 검색타입에 따라 모든 노드에
다큐먼트의 식별자와 점수를 얻기위해 질의를 수행 이후 필요한 다큐먼트를 포함하고
있는 관련 샤드에 응답에 필요한 다큐먼트를 요청하는 질의를 수행.

-라우팅: 색인과 질의과정에서 사용할 라우팅값을 명세하여, 실제 질의시 이를 통해 필요한 샤드만
질의, 검색함.

단. 샤드 수보다 라우팅값이 많을 수 있으므로 필터 필요.
(단일 샤드에 여러 사용자의 자료가 있으면, 샤드 수보다 라우팅 값이 많아짐)

설정방법: 타입 정의시 _routing (required :true,false), (path: "filed name")