엘라스틱서치(Elasticsearch) 소개

순천향대학교 컴퓨터공학과

이 상 정

학습 내용

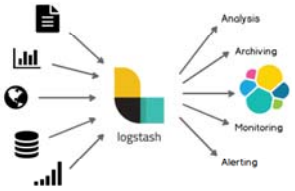
1. 엘라스틱서치 소개
2. 컴퓨터 네트워크 복습 – HTTP
3. JSON 과 REST
4. 엘라스틱서치 사용 예

1. 엘라스틱서치 소개

엘라스틱서치(Elasticsearch) 소개 (1)

- 분산화되고 확장 가능한 실시간 검색 및 분석 엔진 (distributed, scalable, real-time search and analytics engine)
  - 전문(full-text) 검색, 구조화된 데이터의 실시간 분석
  - 분산시스템
    - 여러 개의 노드에 데이터를 색인(indexing)하고 검색, 저장
    - 노드는 독립적으로 실행되는 프로세스
    - 데이터 저장소(data store)
      - 수백 대의 서버로 scale out
      - PB 급의 데이터 저장
  - 아파치 루씬(Apache Lucene) 사용하여 개발
    - Java로 개발된 검색 라이브러리
    - https://lucene.apache.org/
  - JSON 문서 기반 저장 및 검색, HTTP 프로토콜의 RESTful API 지원

엘라스틱서치(Elasticsearch) 소개 (2)



- ELK 스택으로 제공
  - E (Elasticsearch): 데이터 저장 및 검색
  - L (Logstash): 데이터 수집
    - 로그스태시(Logstash)는 데이터의 입력, 변환, 출력을 실시간 파이프라인으로 처리하는 오픈 소스 데이터 수집 엔진
    - 다양한 입력 소스에서 동시에 데이터를 수집(Ingest)하여 변환한 후 자주 사용하는 "스태시(Stash)-보관소"(엘라스틱서치)로 전송
  - K (Kibana): 데이터 시각화 및 리포팅 분석
    - 키바나(Kibana)는 엘라스틱서치와 연동하여 동작하는 데이터 시각화 (리포팅) 및 분석

엘라스틱서치 사용 사례

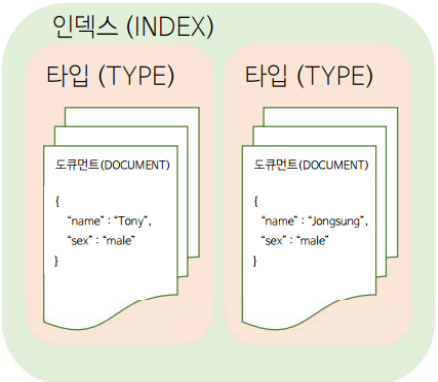
- 위키피디아 (WIKIPEDIA)
  - 전문검색(FULL TEXT SEARCH)를 수행
  - 실시간 타이핑 검색
  - 추천 검색어 기능
- 스택 오버플로우 (STACK OVERFLOW)
  - 검색내용과 결과를 통합해 유사한 질문과 해답을 연결
- 깃허브 (GITHUB)
  - 1,300억 줄이 넘는 소스 코드를 검색하는 데 사용
- 더 가디언 (THE GUARDIAN)
  - 방문객의 로그 분석
  - 소셜 데이터 생성 및 분석으로 실시간 응대
  - 기사에 대한 반응 분석
- 골드만 삭스 (GOLDMAN SACHS)
  - 매일 5TB가 넘는 데이터를 저장
  - 주식 시장의 변동 분석에 사용

몽고 DB (MongDB)와 비교

	몽고 DB 4.7.0	엘라스틱서치 1.0
구분	데이터 저장소	검색 엔진 어플리케이션
개발언어	C++	자바
운영체제	리눅스, OS X, 솔라리스, 윈도우	자바 가상 머신이 설치된 모든 OS
접속 방식	자체 프로토콜	RESTFul/HTTP API
문서 구조	JSON	JSON

주요 용어 소개 (1)

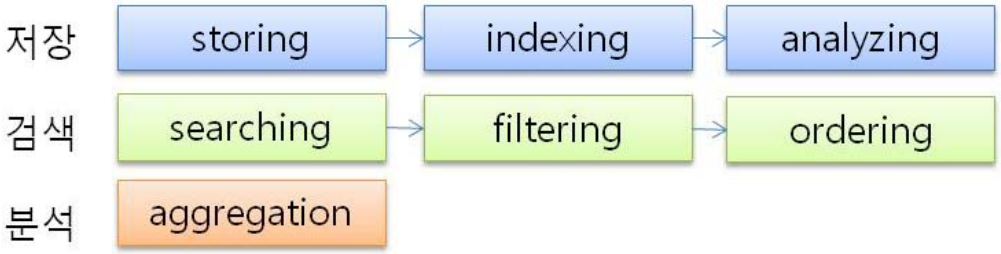
- 인덱스(index), 명사
  - 관련된 다큐먼트(document)를 저장하는 장소
  - 관계형 데이터베이스에서 *데이터베이스*에 해당
- 색인(index, indexing), 동사
  - 검색과 질의를 위해 문서를 인덱스로 변환하여 저장하는 과정
- 데이터 구조는 인덱스, 타입, 다큐먼트 단위로 구성



주요 용어 소개 (2)

- vs. RDBMS

RDBMS	database	table	row	column
elasticsearch	index	type	document	field



2. 컴퓨터 네트워크 복습 – HTTP

HTTP 개요

- HTTP (HyperText Transfer Protocol)는 웹의 애플리케이션 계층 프로토콜

- 클라이언트/서버 모델
  - 클라이언트는 웹 객체들을 요청하고 받아서 디스플레이 하는 브라우저 (browser)
  - 서버 는 요청에 응답하여 객체들을 보내는 웹 서버

HTTP 요청 메시지 (HTTP Request Message)

- 두 유형의 HTTP 메시지: 요청(request), 응답(response)
- HTTP 요청 메시지
  - ASCII 텍스트

요청 라인(request line)
(GET, POST,
HEAD commands)

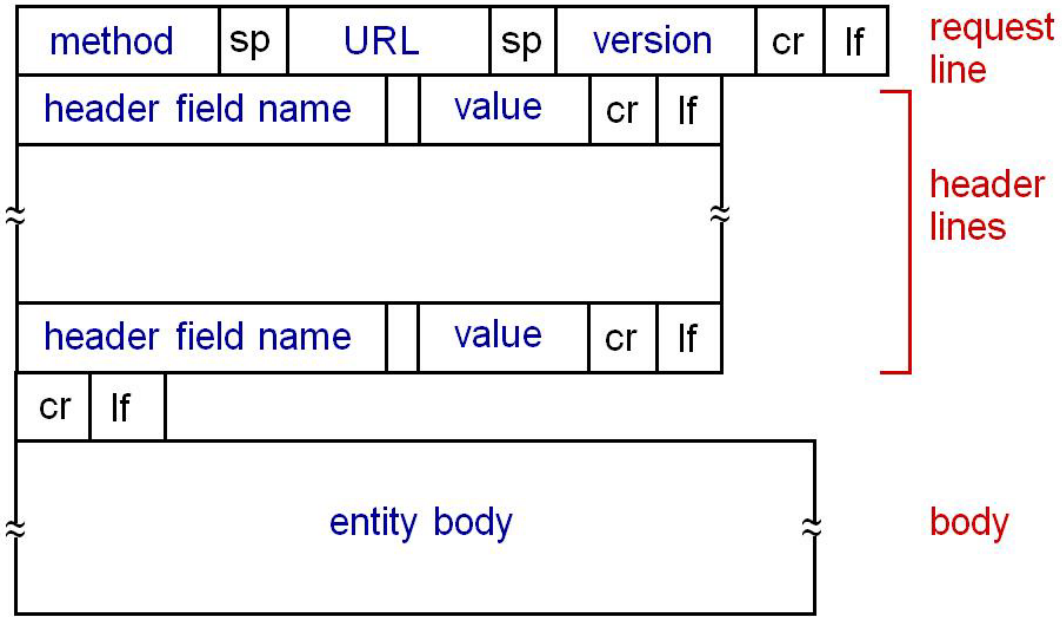
carriage return character
line-feed character

헤더 라인
(header lines)

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

CR(Carriage Return),
LF(Line Feed)
메시지의 끝을 표시

HTTP 요청 메시지: 일반 포맷



폼 입력 업로드 (Uploading Form Input)

- 웹 페이지는 폼 입력을 포함
  - 사용자가 브라우저의 폼 필드(form field)에 입력하여 서버에 전달
- POST 메서드 (POST method)
  - 입력은 개체 몸체(entity body)로 서버에 업로드
- URL 방식
  - GET 메서드(method) 사용
  - 입력은 요청 라인의 URL 필드로 서버에 업로드

```
www.somesite.com/animalsearch?monkeys&banana
```

메서드 유형 (Method Type)

- HTTP/1.0
  - GET
  - POST
  - HEAD
    - GET 메서드과 유사하나 서버가 응답 시 요청된 객체는 보내지 않음
- HTTP/1.1
  - GET, POST, HEAD
  - PUT
    - URL 필드에 명시된 경로로 개체 몸체 안의 파일을 업로드
  - DELETE
    - URL 필드에 명시된 파일을 삭제

HTTP 응답 메시지 (HTTP Response Message)

상태 라인(status line)
프로토콜(protocol)
상태 코드(status code)
상태 문장(status phrase)

헤더 라인(header lines)

데이타, e.g.,
요청된 HTML file

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02
   GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-
   1\r\n
\r\n
data data data data data ...
```

HTTP 응답 상태 코드 (HTTP Response Status Code)

- 응답 메시지의 상태 라인에서 표시
- 일반적인 상태 코드
  - 200 OK
    - 요청이 성공되었고, 요청된 객체가 이 메시지로 보내짐
  - 301 Moved Permanently
    - 요청된 객체가 이동되었고, 새로운 위치는 메시지의 "Location:" 헤더로 표시
  - 400 Bad Request
    - 서버가 요청을 이해할 수 없다는 일반 오류 코드
  - 404 Not Found
    - 요청된 문서가 서버에 존재하지 않음
  - 505 HTTP Version Not Supported
    - 요청된 HTTP 프로토콜 버전을 서버가 지원하지 않음

3. JSON과 REST

JSON 소개

JSON (Javascript Object Notation)

- 경량의 데이터 교환(객체 직렬화) 형식
- XML과 비교하여 데이터 용량이 줄고, 빠른 전송 및 처리 속도
- JSON 데이터는 key/value (name/value) 형식으로 구성
- 기본 자료형: 수, 문자열, 참/거짓, 객체, 배열

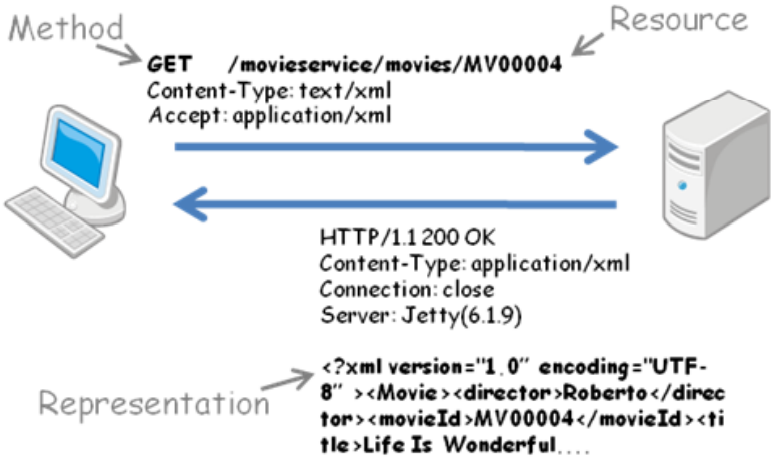
자료형	표현 방법	예
수(number)	정수 또는 실수	"number" : 1
문자열(string)	큰 따옴표로 묶음	"name" : "송중기"
참/거짓(boolean)	true 또는 false	"isResult" : true
객체(object)	여러 개의 key/value를 입력하여 중괄호로 묶음	{"name" : "송중기", "gender" : "남자"}
배열(array)	여러 개의 object를 대괄호로 묶음	{ "employees" : [{"name" : "송중기", "gender" : "남자"}, {"name" : "싸이", "gender" : "남자"}, {"name" : "김태희", "gender" : "여자"}] }

JSON 예

```
{
  "_index" :    "megacorp",
  "_type" :     "employee",
  "_id" :       "1",
  "_version" : 1,
  "found" :     true,
  "_source" :   {
      "first_name" :  "John",
      "last_name" :   "Smith",
      "age" :         25,
      "about" :       "I love to go rock climbing",
      "interests":  [ "sports", "music" ]
  }
}
```

REST 소개

- REST (REpresentaional State Transfer) 아키텍처
  - 웹을 활용한 분산 시스템 소프트웨어 아키텍처
  - URI로 자원을 기술하고, HTTP로 전송
    - HTTP 메서드로 자원의 제어, 명령의 결과는 XML, JSON 등으로 응답
  - 2000년 Roy Fielding의 박사학위 논문에서 제안



REST 소개 – HTTP 메서드

- 자원에 대한 명령은 CRUD로 압축
  - C : Create. 자원을 생성, SQL의 Insert
  - R : Read. 자원의 정보를 읽기, SQL의 Select
  - U : Update. 자원의 정보를 업데이트, SQL의 Update
  - D : Delete. 자원의 삭제, SQL의 Delete
- REST는 HTTP 메서드로 원격의 자원을 제어하는 명령을 기술
  - C: POST, R: GET, U: PUT, D: DELETE
- 명령의 결과 데이터는 HTTP 몸체(body)로 전송
  - 데이터 형식은 XML, JSON 등으로 표현

HTTP, CRUD, SQL 비교

HTTP METHOD, CRUD, SQL 비교

HTTP METHOD	CRUD	SQL
GET	READ	SELECT
PUT	UPDATE	UPDATE
POST	CREATE	INSERT
DELETE	DELETE	DELETE

3. 엘라스틱서치 사용 예

직원 데이터 관리 예 – 색인 (indexing)

- 직원의 다큐먼트를 색인 (저장)

```
PUT /megacorp/employee/1
{
    "first_name" : "John",
    "last_name" :  "Smith",
    "age" :        25,
    "about" :      "I love to go rock climbing",
    "interests": [ "sports", "music" ]
}
```

- 직원에 대한 상세 정보를 각각의 다큐먼트로 저장
- 각 다큐먼트는 자신만의 id를 가짐
- 다큐먼트의 타입(type)은 employee
- employee 타입은 megacorp 인덱스(index)에 포함

직원 데이터 관리 예 – 조회 (retrieving)

- 직원의 다큐먼트를 조회(retrieving)

```
GET /megacorp/employee/1
```

```
{
  "_index" :   "megacorp",
  "_type" :    "employee",
  "_id" :      "1",
  "_version" : 1,
  "found" :    true,
  "_source" :  {
      "first_name" :  "John",
      "last_name" :   "Smith",
      "age" :         25,
      "about" :       "I love to go rock climbing",
      "interests":  [ "sports", "music" ]
  }
}
```

직원 데이터 관리 예 – 기본 검색 (basic search)

- 기본 문법

```
GET /megacorp/employee/_search
```

- 문자열 검색

```
GET /megacorp/employee/_search?q=last_name:Smith
```

- DSL (domain-Specific Language)로 질의

```
GET /megacorp/employee/_search
{
    "query" : {
        "match" : {
            "last_name" : "Smith"
        }
    }
}
```

sql 비교

```
select *
from employee
where last_name = "Smith"
```


직원 데이터 관리 예 – 필터된 검색 (filtered search)

- 질의의 범위를 좁힘

```
GET /megacorp/employee/_search
{
    "query" : {
        "filtered" : {
            "filter" : {
                "range" : {
                    "age" : { "gt" : 30 }
                }
            },
            "query" : {
                "match" : {
                    "last_name" : "smith"
                }
            }
        }
    }
}
```

sql 비교

```
select *
from employee
where age > 30
and last_name = "Smith"
```

직원 데이터 관리 예 – 전문 검색 (full-text search)

- 연관 점수(relevance score)에 따라 일치(match)한 다큐먼트들을 정렬

```
GET /megacorp/employee/_search
{
    "query" : {
        "match" : {
            "about" : "rock climbing"
        }
    }
}
```

```
{
   ...
   "hits": {
      "total":      2,
      "max_score":  0.16273327,
      "hits": [
         {
            ...
            "_score":         0.16273327,
            "_source": {
               "first_name":  "John",
               "last_name":   "Smith",
               "age":         25,
               "about":       "I love to go rock climbing",
               "interests": [ "sports", "music" ]
            }
         },
         {
            ...
            "_score":         0.016878016,
            "_source": {
               "first_name":  "Jane",
               "last_name":   "Smith",
               "age":         32,
               "about":       "I like to collect rock albums",
               "interests": [ "music" ]
            }
         }
      ]
   }
}
```


직원 데이터 관리 예 – 구문 검색 (phrase search)

- 정확히 일치하는 구문만 검색

```
GET /megacorp/employee/_search
{
    "query" : {
        "match_phrase" : {
            "about" : "rock climbing"
        }
    }
}
```

```
{
   ...
   "hits": {
      "total":      1,
      "max_score":  0.23013961,
      "hits": [
         {
            ...
            "_score":         0.23013961,
            "_source": {
               "first_name":  "John",
               "last_name":   "Smith",
               "age":         25,
               "about":       "I love to go rock climbing",
               "interests": [ "sports", "music" ]
            }
         }
      ]
   }
}
```

직원 데이터 관리 예 – 구문 검색 (phrase search)

- 검색 결과에서 매칭된 부분 하이라이트하기

```
GET /megacorp/employee/_search
{
    "query" : {
        "match_phrase" : {
            "about" : "rock climbing"
        }
    },
    "highlight": {
        "fields" : {
            "about" : {}
        }
    }
}
```

```
"hits": [
   {
      ...
      "_score":         0.23013961,
      "_source": {
         "first_name":  "John",
         "last_name":   "Smith",
         "age":         25,
         "about":       "I love to go rock climbing",
         "interests": [ "sports", "music" ]
      },
      "highlight": {
         "about": [
            "I love to go <em>rock</em> <em>climbing</em>"
         ]
      }
   }
]
```


직원 데이터 관리 예 – 분석 (analytics)

- 어그리게이션(aggregation, 집계/종합) 을 사용하여 직원들의 취미 조사

```
GET /megacorp/employee/_search
{
  "aggs": {
    "all_interests": {
      "terms": { "field": "interests" }
    }
  }
}
```

```
{
   ...
   "hits": { ... },
   "aggregations": {
      "all_interests": {
         "buckets": [
            {
               "key":       "music",
               "doc_count": 2
            },
            {
               "key":       "forestry",
               "doc_count": 1
            },
            {
               "key":       "sports",
               "doc_count": 1
            }
         ]
      }
   }
}
```

참고 자료

- Elasticsearch: The Definitive Guide, Getting Started
  - You Know, for Search...
    - https://www.elastic.co/guide/en/elasticsearch/guide/current/index.html
- 시작하세요! 엘라스틱서치, 김종민, 위키북스, 2015
  - 01. 엘라스틱서치
- 엘라스틱서치 입문
  - http://www.slideshare.net/seunghyuneom/elastic-search-52724188
- 엘라스틱서치 적용 및 활용
  - http://www.slideshare.net/JunyiSong1/elasticsearch-45936425
  - Elasticsearch: The Definitive Guide 요약