Elasticsearch

Advanced Topics on NoSQL databases

A4 - S8

ESILV
nicolas.travers (at) devinci.fr

Chapter 1

Setting Movies dataset

For this practice work, we will use the elasticsearch indexing software.

1.1   cURL

cURL is a small executable which send/receive HTTP requests in command lines. You can specify headers by adding parameters to your command (XPUT/XGET/XPOST, -H"Content-Type...", --databinary...).

You can download cURL (already installed under Linux and MacOSX) here (better to use the SSL embedded version):
`https://curl.haxx.se/download.html`.

1.2   Elasticsearch & Kibana

The installation steps are little bit straightforward.

- Local installation: `https://www.elastic.co/downloads`

- Docker : name "elasticsearch-kibana" (@nshou v 6.5.4)

1.3   Import dataset with cURL

- Download the `movies_elastic.json.zip` data file[1], and unzip it

- Import it in a **command line** with:

```
curl -XPUT localhost:9200/_bulk -H"Content-Type: application/json" --data-binary @movies_elastic.json
```

---
[1]`https://devinci-online.brightspace.com/d2l/le/content/15379/viewContent/18377/View`

Chapter 2

DSL queries

For each query, you can write in two formats (when possible):

- With the HTTP GET method, with the q= parameter

- With the HTTP POST method, with the JSON document param "*query.json*" (called DSL queries):

```
curl -XPOST "localhost:9200/movies/movie/_search" -H"Content-Type: application/json"
          -d @query.json -o output.json
```

And then open this JSON document in your Web browser.

- You can also make those queries directly in Kibana in the "Dev Tools"

2.1   Simple queries

2.1.1 Every movies which title matches 'Star Wars' (match query),

2.1.2 Try with exact match (match_phrase),

2.1.3 Star Wars movies and Directors equal to 'George Lucas' (boolean query),

2.1.4 Movies were 'Harrison Ford' played,

2.1.5 Movies were 'Harrison Ford' played with a plot containing 'Jones',

2.1.6 Movies were 'Harrison Ford' played with a plot containing 'Jones' but plots without containing 'Nazis'

2.1.7 Movies of 'James Cameron' which rank is better than 1000 (boolean + range query)

2.1.8 Movies of 'James Cameron' which rating <u>must</u> be higher than 5

2.1.9 Movies of 'James Cameron' which rating must be higher than 5 and which genre must not be 'Action' nor 'Drama'

2.1.10 Movies of 'J.J. Abrams' which released date is mandatory between 2010 and 2015 (filtered query)

2.2   Aggregate queries

We wish now to do some aggregate queries on the index in order to extract some statistics.

2.2.1   Complex queries

2.2.1 Give for each year the number of movies,

2.2.2 For each category (genres), give the number of movies. To take into account the whole text, you need to use "keyword" after the required field.

2.2.3 Give the average rating of movies,

2.2.4 Give the average rating of George Lucas' movies,

2.2.5 Count the number of movies for the given ranges of rating: 0-1.9, 2-3.9, 4-5.9...),

2.2.6 Number of distinct directors in adventures movies,

2.2.2 Hard queries

2.2.1 Give the average rating per genre,

2.2.2 Give min, max and average rating for each genre,

2.2.3 Give average movie's rank and average movie's rating for each director. Sort the result decreasingly on average rating,

2.2.4 Give the terms occurrences extracted from each movie's title. The text value requires a specific mapping on the dataset stored in elasticsearch, see: `https://www.elastic.co/guide/en/elasticsearch/reference/current/fielddata.html`.

2.2.5 Most significant terms in plots of George Lucas movies,