

Elasticsearch

Presented By: Scott Harris and Noah Dunn

Introductory Information

- System Name: **Elasticsearch**
- Primary Database Model: **Search Engine Database**
- Host Company: **Elasticsearch B.V (AKA Elastic)**.
- URL: [Link to the Company](#)



The Authors

Scott Harris:



Noah Dunn:

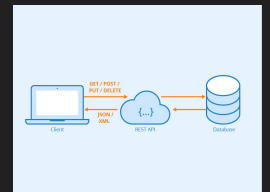


System Overview

- History
 - Based on Apache Lucene
 - First released in 2010
- Intended Usage
 - REST API
 - Retrieving Small Doses of Information
 - Website Searching
 - Focus on Read Operations
- Special Feature(s)
 - Fuzzy or Incomplete Querying
 - Optimized for Data Retrieval
 - Document Store



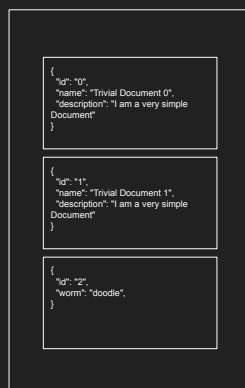
Lucene



Document Store

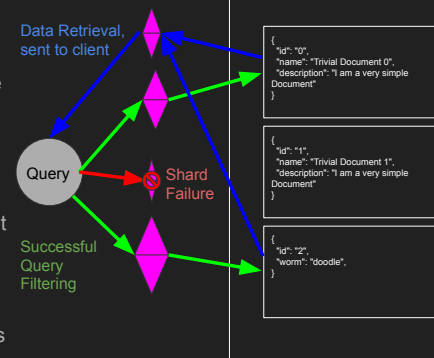
- Elasticsearch makes use of JSON Document-Store format
- The 'Database' consists of the entire collection of all Documents
- Much like Mongo or other NoSQL solutions, these documents are unstructured and can contain a wide array of keys and values

Elasticsearch Instance



Sharding Visual

- All Database Processing in Elasticsearch is done via what are called 'shards'
- Shards are processing units of various sizes that can handle retrieval for queries
- Shards can merge together or split for increased distribution or increased processing power
- Shards provide 'fault-tolerance'. If one process fails, the other shards will pick up the slack



INSERT Query Example

```
35.245.53.182:9200/people/doc/

POST 35.245.53.182:9200/people/doc/

Body
none form-data x-www-form-urlencoded raw binary JSON(application/json)

1: {
2:   "name": "Beah Dunn",
3:   "age": "Older",
4:   "job": "Professional Cheese Grater"
5: }
```

SELECT Query Example

```
GET 35.245.53.182:9200/people/_search

Body
1: {
2:   "query": {
3:     "match": "Older"
4:   }
5: }
```

```
{
  "took": 6,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 2.160473,
    "hits": [
      {
        "_index": "people",
        "_type": "doc",
        "_id": "1",
        "_score": 2.160473,
        "_source": {
          "name": "Guy Fieri",
          "home": "Flavortown",
          "job": "Chef"
        }
      }
    ]
  }
}
```

Multi-Conditional Querying

```
GET 35.245.53.182:9200/people/_search

Body
1: {
2:   "query": {
3:     "bool": {
4:       "must": [
5:         { "match": { "name": "Scott Harris" } },
6:         { "match": { "home": "Texas" } }
7:       ]
8:     }
9:   }
10: }
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

{
  "took": 5,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 3.3631094,
    "hits": [
      {
        "_index": "people",
        "_type": "doc",
        "_id": "AUX7E-Sv1S2DA0Ssq1W",
        "_score": 3.3631094,
        "_source": {
          "name": "Scott Harris",
          "home": "Texas",
          "job": "Professional Idiot"
        }
      }
    ]
  }
}
```

Fuzzy Matching Example

```
GET 35.245.53.182:9200/people/_search

Body
1: {
2:   "query": {
3:     "bool": {
4:       "must": [
5:         { "match": { "name": "Scott Harris" } },
6:         { "match": { "home": "Texas" } }
7:       ]
8:     }
9:   }
10: }
```

```
GET 35.245.53.182:9200/people/_search

Body
1: {
2:   "query": {
3:     "bool": {
4:       "must": [
5:         { "match": { "name": "Scott Harris" } },
6:         { "match": { "home": "Texas" } }
7:       ]
8:     }
9:   }
10: }
```

Links, Description, Relevant Documentation

- Official Website
 - <https://www.elastic.co/>
- Querying Documentation
 - <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>
- A More Complex Look at Sharding
 - <https://buildingvts.com/elasticsearch-architectural-overview-a35d8910e515>
- Guide: Installation to Querying
 - <https://medium.com/@factoryhr/elasticsearch-introduction-implementation-and-example-17dd66c35c35>
- Link to Where to Get Install Packages
 - <https://www.elastic.co/guide/en/elasticsearch/reference/current/install-elasticsearch.html>

Challenges, Issues, Things to keep in mind

- Things Elasticsearch doesn't do well or at all
 - Deletion (Finicky)
 - Joining different documents (Not possible from our research)
 - Bulk insertion
 - There is no custom client offering. Everything is done in REST calls, which can be a negative to those unfamiliar with HTTP or REST.
- Potential DevOPS Issues
 - Google Cloud consumes credit for an instance very quickly
 - Local Installs have to be manually started for use
 - Shard allocation can lead to consuming a large portion of CPU
- When Querying
 - There are at least two ways to write each query: POST body call, and URL modification
 - Do what you find easier