Elasticsearch as a new search engine for Invenio and more

Johnny Mariéthoz

Johnny.Mariethoz@rero.ch

RERO DOC: in summary

- CDS Invenio v1.1 + RERO patches
- 26’000 records + 170’000 print media issues
- Theses, Dissertations, Books, Newspapers, Journals, Postprints, Preprints, Reports, Maps, Audio Recordings, Music Scores, Print Media
- scientific and heritage documents
- 51 institutions
- use of Multivio (http://multivio.org) as document viewer

What's (almost) new in RERO DOC

- hierarchical facets
- filters and filter storage
- snippets with stemming
- digitized press navigation

http://doc.rero.ch

Elasticsearch

Elasticsearch the engine

- features
  - distributed
    - just start a new node with the same cluster name
- (nested) document oriented
- schema free (need mapping for complex data)
- apache2 open source license
- built on the top of Lucene

Glossary

- index
  - like a database in a relational database
- type or document type
  - like a table in a relational database
- document
  - like a row in a table in a relational database
- field
  - similar to a column in a table in a relational database

Concepts

- cluster (node == machine)
- shards (Lucene instance)
- replicas (replication)
- mapping (field constraints)
- filters (boolean) and query (ranking)
- analyzers
  - charfilters (html strip)
  - tokenizers (whitespace)
  - tokenfilter (lowercase)

Simple to use

Start the cluster

```
./bin/elasticsearch
```

Initiate a connection

```
from pyelasticsearch import ElasticSearch
con = ElasticSearch("http://localhost:9200")
```

Delete an Index

```
con.delete_index(”invenio")
```

Create an Index

```
con.create_index(”invenio")
```

Simple to use

Index a record

```
data = {
        'recid': 1,
        'language' : 'eng',
        'title': [{
            'title': 'my great title',
            'subtitle': 'my great subtitle',
            }]
        }

con.index(index="invenio", doc_type="records", doc=data,
    id=data['recid'])
```

Simple to use

Refresh for direct access

```
con.refresh(index=”invenio")
```

Get a record

```
con.get(index=”invenio", doc_type=”records", id=1)
```

Several search queries

```
con.search(index=”invenio", query=”great")
con.search(index=”invenio", query=”title.subtitle:great")
con.search(index=”invenio", query=”title.\*:great")
```

Mapping I

Declaration and identifier

```
mapping = {
    ”records": {
        "properties" : {
            #force recid type to integer for default sorting
            "recid" : {
                "type" : "integer"
                },
```

Mapping II

Language

```
"language": {
        "type": "string",
        "fields": {
          "facet_language": {
            "type": "string",
            "index": "not_analyzed"
          }
        },
      },
```

Mapping III

Title

```
"title": {
    "properties": {
      "subtitle": {
        "type": "string"
      },
      "title": {
        "type": "string",
        "analyzer": "english",
        "fields": {
          "sort_title": {
            "type": "string"
          }}}}},
```

Mapping IV

Send Mapping to the Server

```
            }
        }
    }
con.put_mapping(index=”invenio", doc_type=”records",
    mapping=mapping)
```

Support Special Search Query

```
#only with mapping
con.search(index="test", query="title.title:great")
```

Query and filters

- demo with marvel & sense

Query and filters

```
"size": 4,
  "fields": ["title.title", "title.subtitle"],
  "query": {
    "filtered": {
      "query": {
        "query_string": {
          ”query": "abstract.summary:results”
      }},
      "filter": {
        "bool": {
          "must": [
            {"term": {
              "language.facet_language": "eng"
            }}]}}},
```

Query and filters

```
"sort": [
    {
      "title.title.sort_title": {
        "order": "desc"
      }}],
  "aggs": {
    "authors": {
      "terms": {
        "field": "authors.full_name.facet_authors",
        "size": 10
      }}},
```

Query and filters

```
"highlight": {
    "fields": {
      "title.title": {
        "number_of_fragments": 1,
        "fragment_size": -1
      },
      "abstract.summary": {
        "number_of_fragments": 3,
        "fragment_size": 20
      }
    }
  }
}
```

Invenio and elasticsearch

Useful features for Invenio

- parent - children relation
- facets
- filters
- nested documents (python dictionaries)
- regexp facets
- python language supported
- highlight (snippets)
- many more (similar doc, search completion, …)

Invenio (pu) specific setup

- three different indexes
  - one index for the records (metadata)
  - one index for the collections (hierarchical facets)
  - one index for fulltext
  - signal connections to update indexes
- field alias with different analyzers
  - for sorting
  - for facetting
  - for searching

JsonAlchemy configuration

```
language:
    elasticsearch:
        mapping: {
            "language": {
                "type": "string",
                    "fields": {
                        "facet_language": {
                            "type": "string",
                            "index": "not_analyzed”}}}}
        facets: {
            "language": {
                "terms" : {
                    "field" : "facet_language",
                    "size": 10,
                    "order" : { "_count" : "desc" }}}}
```

Invenio/pu + demosite (demo)

- search view
  - https://github.com/jma/elasticsearch_view
- specific demosite package
  - https://github.com/jma/invenio-demosite/tree/pu-elasticsearch

Digitalized press (demo)

Digitalized press

- parent-children records organization
- ES based ajax requests through a flask/wsgi API
- date range filters
- fulltext search only
- weekday facets
- hierarchical navigation for print media

Misc: user statistics

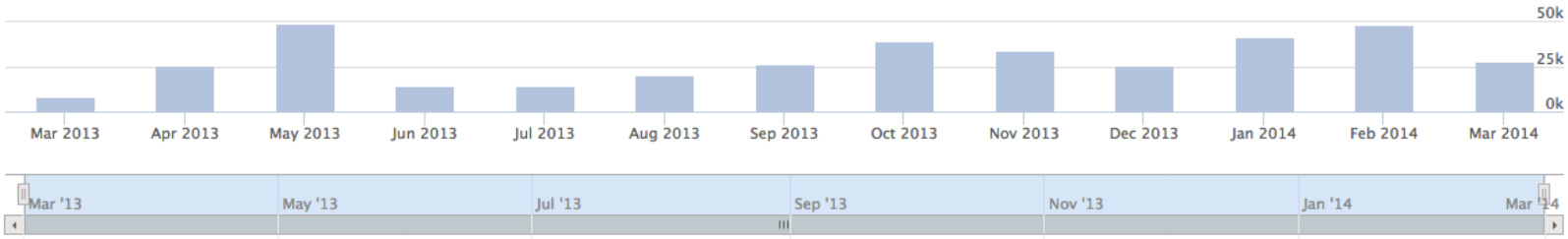
- prototype
  - kibana + Elasticsearch
  - based on apache logs
  - needs Elasticsearch instance public access
- final version
  - angular + flask API + Elasticsearch
  - based on apache logs
  - Expose only read-only ES function using Flask-restfull API

Filtres

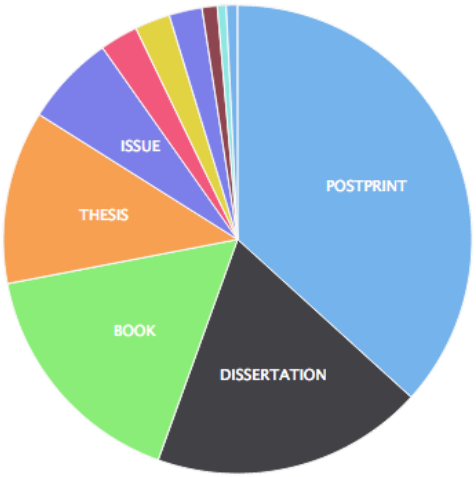
✖ language:French

Zoom 1m 3m 6m YTD 1y All

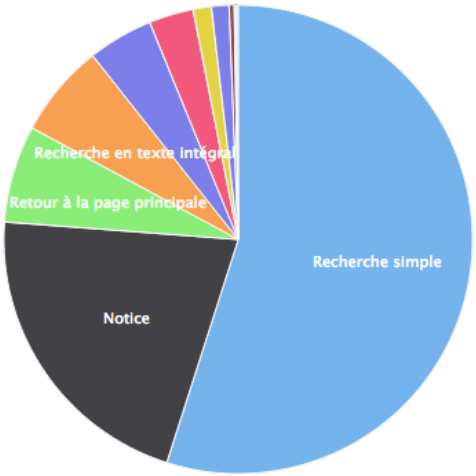
From Mar 1, 2013 To Mar 1, 2014



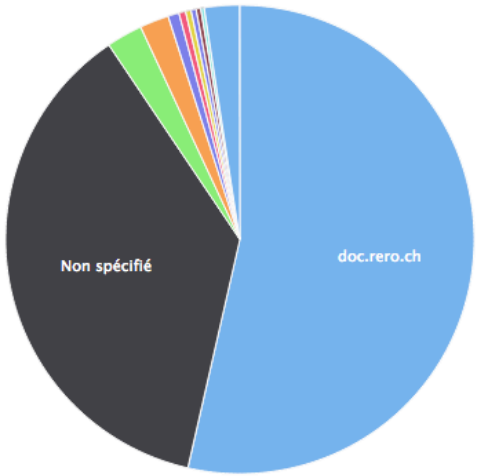
Type de document Graph Table



Type de contenu Graph Table



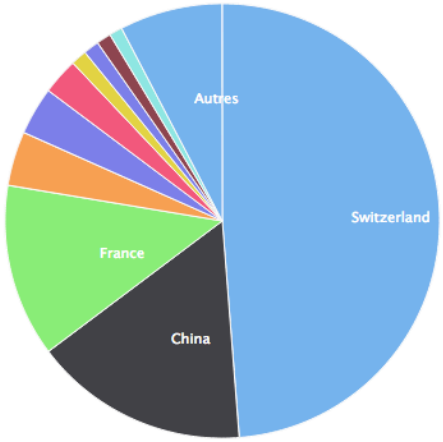
Référant Graph Table

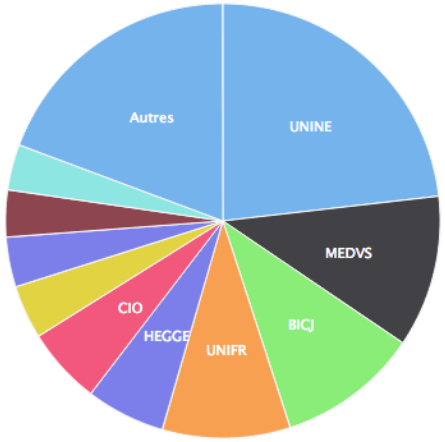


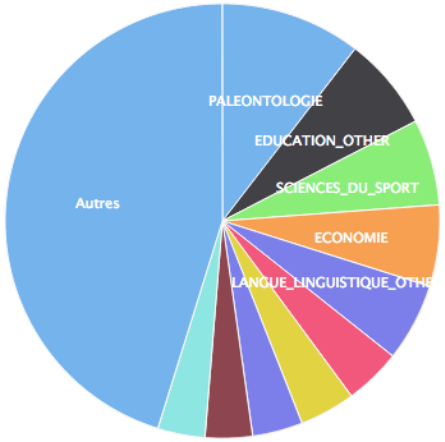
Pays Graph Table

institution Graph Table

Domaine CDU Graph Table

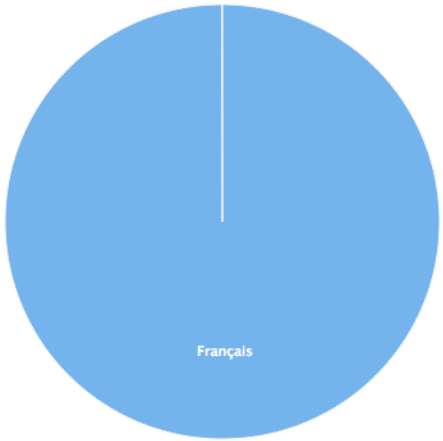






Langue

Graph Table



Fichier téléchargé

FleuryN-these.pdf?ln=frve	10.7%
these_BancilaV.pdf?ln=frv	10.5%
CarnelliA-these.pdf?ln=fr	5.3%
HuckL-these.pdf?ln=frvers	3.7%
BergerA-these.pdf?ln=frve	3.7%
DreyfusP-these.pdf?ln=frv	3.3%
AcciettoC-these.pdf?ln=fr	2.5%
these_TorglerJ.pdf?ln=frv	2.1%
these_BoffiElAmariE.pdf?l	2.1%

Notice détaillée

32150	1.0%
12852	1.0%
18186	0.5%
11876	0.4%
208866	0.4%
205530	0.2%
31283	0.2%
12529	0.2%
6302	0.2%

Server settings

- 6 GB of RAM for ES
- file descriptors
  - /etc/security/limits.conf (64’000)
- number of shards (1), replicas (1)
- index size (128 GB)
- number of document (~200’000)
- fulltext search response time (few seconds to display the results page)

Kiitos! Any questions?