

Bases de données
documentaires et distribuées,
<http://b3d.bdpedia.fr>

BD documentaires et moteurs de
recherche: Elasticsearch

Moteurs de recherche

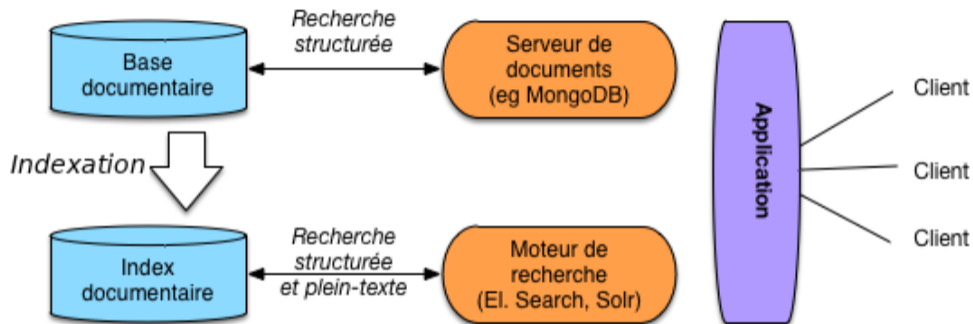
Moteurs libres

- ▶ Apache Solr (Lucene)
- ▶ **ElasticSearch (Lucene)**
- ▶ Sphinx
- ▶ Xapian

Moteurs commerciaux

- ▶ Google Search Appliance
- ▶ Exalead, Qwant
- ▶ Amazon CloudSearch
- ▶ Microsoft Azure Search, Bing

Bases documentaires et moteur de recherche



Bases documentaires et moteur de recherche

Un scénario typique :

- ▶ une recherche par mot-clé dans un site
- ▶ Les données du site sont gérées classiquement par une base documentaire (MySQL, Postgres, MongoDB)
- ▶ On extrait de la base tous les textes à **indexer** et on en fait des documents ES/Solr, à qui on les confie
- ▶ Ce dernier se charge alors de répondre quand un utilisateur emploie un champ de recherche.

Bases documentaires et moteur de recherche

Un système comme Elasticsearch est consacré à la **recherche**, c'est-à-dire la lecture la plus efficace possible de documents.

Il s'appuie pour cela sur des structures compactes, compressées, optimisées (les **index inversés**)

Ce n'est pas (pas forcément) un très bon outil pour les autres fonctionnalités d'une base de données :

- ▶ Le stockage par exemple n'est ni aussi robuste ni aussi stable.
- ▶ Mises à jour coûteuses et asynchrones

Présentation d'Elasticsearch

Elasticsearch est un moteur de recherche basé sur Lucene

- ▶ grande communauté d'utilisateurs
- ▶ open source, profite de la recherche dans le domaine sur Lucene
- ▶ utilisé par de grands opérateurs du Web sur des collections immenses

En fait, plusieurs composants dont :

- ▶ Logstash, l'ETL, pour extraire, transformer et charger les données
- ▶ ElasticSearch, le moteur lui-même
- ▶ Kibana, pour produire des tableaux de bords de surveillance

Installer Elasticsearch 2.4 et le plugin Kopf

```
docker run -d --name es1 -p 9200:9200 -p 9300:9300 elasticsearch:2.4 \  
-Des.index.number_of_shards=1 \  
-Des.index.number_of_replicas=0
```

```
$ docker ps -a
```

```
$ docker exec <containerId> plugin install lmenezes/elasticsearch-kopf
```

```
# Adresse IP de votre Elasticsearch
```

```
$ ip="$(docker inspect --format '{{ .NetworkSettings.IPAddress }}' es1)"
```

```
$ echo $IP
```

Un premier document

Télécharger la liste des films sur

<http://webscope.bdpedia.fr/index.php?ctrl=xml> (Exports, Listes des films complets, un fichier JSON par film ZIPpé)

```
$ unzip movies-json.zip
```

```
$ cd movies-json/
```

```
<EDITER le fichier pour enlever le champ id>
```

Notre premier document

```
{
  "title": "Vertigo",
  "year": 1958,
  "genre": "drama",
  "summary": "Scottie Ferguson, ancien inspecteur de police, est sujet au vertige depuis qu'il a
              vu mourir son collègue. Elster, son ami, le charge de surveiller sa femme,
              Madeleine, ayant des tendances suicidaires. Amoureux de la jeune femme Scottie
              remarque pas le piège qui se trame autour de lui et dont il va être la victime."
  "country": "DE",
  "director": {
    "_id": "artist:3",
    "last_name": "Hitchcock",
    "first_name": "Alfred",
    "birth_date": "1899"
  },
  "actors": [
    {
      "_id": "artist:15",
      "first_name": "James",
      "last_name": "Stewart",
      "birth_date": "1908",
      "role": "John Ferguson"
    }
  ],
}
```

Un premier document

Utiliser l'API REST pour mettre le document dans Elasticsearch

```
$ curl -X PUT http://localhost:9200/nfe204/movies/movie:1 --data-binary @movie_1.json
```

constater que l'on a bien un index appelé nfe204

```
$ firefox http://localhost:9200/_plugin/kopf/
```

Ensuite, récupérer le document avec curl :

```
$ curl -X GET http://localhost:9200/nfe204/movies/movie:1
```

D'autres documents

```
$ wget http://b3d.bdpedia.fr/files/movies-elastic.json
```

```
# Utiliser l'API REST et l'interface bulk pour déposer les documents dans Elasticsearch
```

```
$ curl -XPUT http://localhost:9200/_bulk --data-binary @movies-elastic.json $
```

```
# constater avec Kopf que l'on a 4849 documents importés dans l'index nfe204.
```
