Bases de données documentaires et distribuées

Introduction à la recherche d'information

Auteurs: Raphaël Fournier-S'niehotta, Michel Crucianu, Marin Ferecatu
(fournier@cnam.fr, michel.crucianu@cnam.fr, marin.ferecatu@cnam.fr)

Département d'informatique
Conservatoire National des Arts & Métiers, Paris, France

Plan du cours

Motivation

- Un moteur de recherche est une application spécialisée dans la recherche, qui s'appuie sur un index.
- On a vu les avantages par rapport aux moteurs des bases de données
- La question naturelle : pourquoi ne pas utiliser directement le moteur de recherche comme gestionnaire des documents.
- Pourquoi s'embarasser de MongoDB alors qu'un moteur permet des recherches puissantes, efficaces, ainsi que le stockage et l'accès aux documents.

Motivation

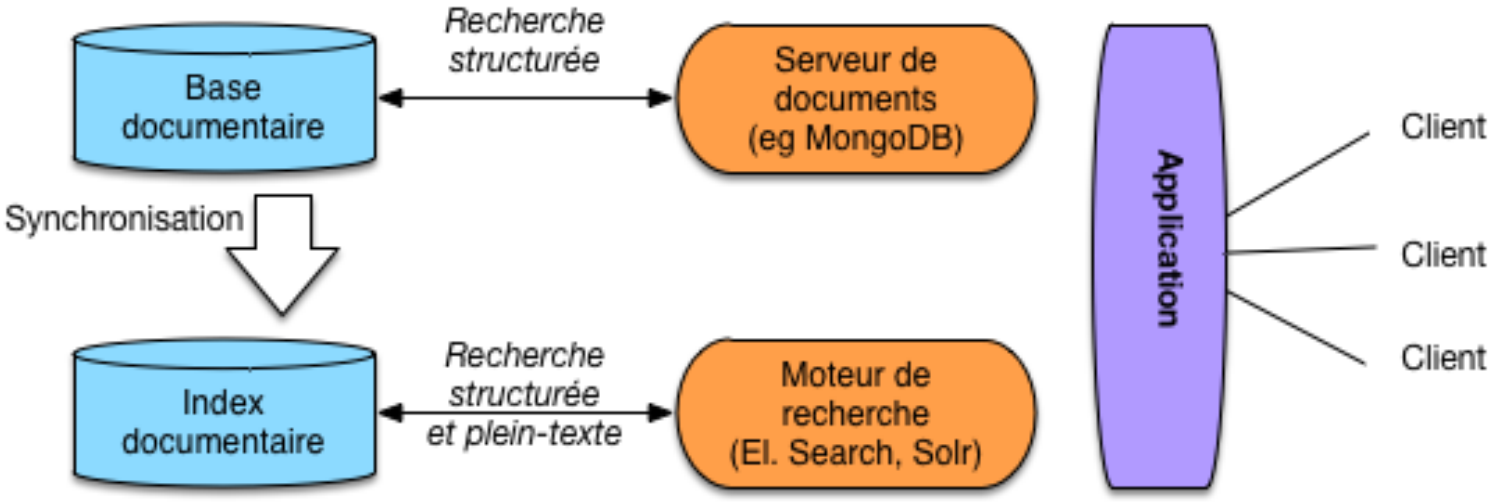
- Un moteur de recherche est entièrement consacré à la recherche (donc à la lecture) la plus efficace possible de documents.
- il s'appuie pour cela sur des structures compactes, compressées, optimisées (les index inversés)
- En revanche, ce n'est pas un très bon outil pour les autres fonctionnalités d'une base de données.
- Le stockage par exemple n'est ni aussi robuste ni aussi stable,
- Il faut parfois reconstruire l'index à partir de la base originale.
- De même, les modifications fréquentes sont moins bien supportées
- Cela tient à la structure même des indexes (donc temps réel délicat)

Intégration

- L'utilisation la plus courante consiste donc à utiliser un moteur de recherche comme un **complément d'un serveur de base de données (relationnelle ou documentaire)**
- On confie aux moteurs de recherche des tâches que le serveur BD ne sait pas accomplir
- Essentiellement : les recherches non structurées.
- Dans le cas des bases NoSQL, l'absence fréquente de tout langage de requête fait du moteur de recherche associé un outil indispensable.
- Les moteurs sont aussi efficaces pour les requêtes structurées !

Inconvénient

- Il faut maintenir plusieurs systèmes
- il faut propager les données de l'un à l'autre
- Il existe des "rivers", configurables pour cela (cours suivants)


Base documentaire
Recherche structurée
Serveur de documents (eg MongoDB)
Application
Client
Client
Client
Synchronisation
Index documentaire
Recherche structurée et plein-texte
Moteur de recherche (El. Search, Solr)

Moteurs de recherche

- Nous allons vous présenter la pratique des moteurs de recherche
- Deux systèmes : Elastic Search, Solr
- Lucene : bibliothèque Java pour indexer et chercher du texte
- Les deux projets reposent sur Lucene
- Solr a transformé la librairie java Lucene en un serveur Web,
- Elasticsearch a métamorphosé Lucene en un serveur web distribué.
- Différences : entreprise / fondation, distribution des données, langage, etc.
- http://solr-vs-elasticsearch.com/

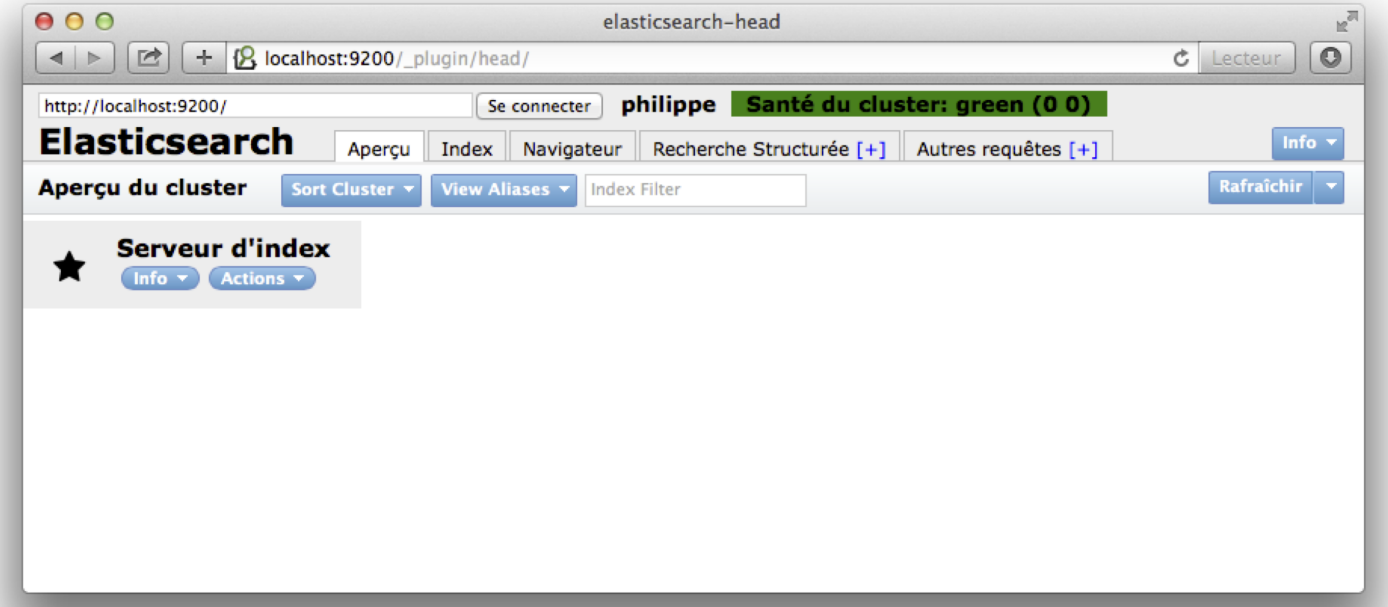
Moteurs de recherche

- Faciles d'installation tous les deux

```
docker pull snasello/elasticsearch
docker run –rm -it -p 9200:9200 snasello/elasticsearch –cluster.name=foobar
docker exec foobar /elasticsearch/bin/plugin –install mobz/elasticsearch-head
```

Test d'installation


```
{
"status" : 200,
"name" : "Serveur d'index",
"cluster_name" : "philippe",
"version" : {
"number" : "1.5.0",
"...": "...",
"lucene_version" : "4.10.2"
},
"tagline" : "You Know, for Search"
}
```


elasticsearch-head
localhost:9200/_plugin/head/
Lecteur
http://localhost:9200/
Se connecter
philippe
Santé du cluster: green (0 0)
Elasticsearch
Aperçu
Index
Navigateur
Recherche Structurée [+]
Autres requêtes [+]
Info
Aperçu du cluster
Sort Cluster
View Aliases
Index Filter
Rafraîchir
Serveur d'index
Info
Actions

ElasticSearch

- Toutes les interactions avec un serveur ElasticSearch passent par une interface REST basée sur JSON.
- ElasticSearch organise les données selon trois niveaux:
  - l'index regroupe des chemins d'accès à un collection de documents;
  - le type désigne le format du document indexé;
  - l'identifiant sert de clé d'accès à un document;
  - chaque document a un numéro de version.
- Pour indexer un de nos films dans l'index nfe204, avec le type movies, on exécute la commande suivante:

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

- Le PUT crée une "ressource" (au sens Web/REST du terme)
- Réponse :

```
{"_index":"nfe204","_type":"movies","_id":"movie:1","_version":1,"created":true}
```


elasticsearch-head
localhost:9200/_plugin/head/
Lecteur
http://localhost:9200/
Se connecter
philippe
Santé du cluster: green (1 1)
Elasticsearch
Aperçu
Index
Navigateur
Recherche Structurée [+]
Autres requêtes [+]
Info
Aperçu du cluster
Sort Cluster
View Aliases
Index Filter
Rafraîchir
nfe204
size: 7.03ki (7.03ki)
docs: 1 (1)
Info
Actions
Serveur d'index
Info
Actions
0

ElasticSearch

- Pour récupérer la ressource créée :

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