ELASTICSEARCH MAINTENANT EN VERSION 1.4

firm1

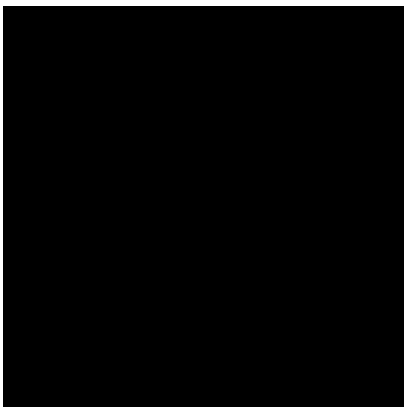
29 octobre 2015

Zeste de savoir

Table des matières

1 Introduction

Elasticsearch, comme son nom pourrait l'indiquer, est un moteur d'indexation *open source*, et très certainement l'un des plus puissants qui existe aujourd'hui. Il y a quelques semaines, il est passé en version 1.4 et c'est l'occasion de vous présenter un peu cet outil, ses fonctionnalités, et à quel type de besoin il répond. Nous allons aussi dans cet article présenter ce qui se fait du côté de la concurrence afin d'avoir une vision plus ouverte du monde des moteurs de recherches.

->  <-

2 Les principaux atouts

2.1 Moteur de recherche vs Moteur d'indexation

Si l'on vous demande de donner le nom de quelques moteurs de recherche, vous parlerez certainement de *Google Search*, *Bing*, *DuckDuckGo*, etc. Et vous aurez bien raison car un moteur de recherche est tout simplement une application (souvent web) qui permet de retrouver des liens, des images, des documents, etc. Bref, des ressources en rapport avec certains mot-clés.
Cependant, pour pouvoir donner des résultats pertinents, un moteur de recherche doit savoir à l'avance où sont les ressources qu'on pourrait lui demander. Pour le savoir, de nombreux moteurs de recherche ont des robots qui parcourent Internet à la recherche de nouvelles ressources. Ils se basent donc sur des **moteurs d'indexation**, dont le rôle est de collecter des ressources, et d'extraire des mot-clés les plus significatifs. Un moteur d'indexation n'est donc qu'un sous-ensemble du moteur de recherche.

Tandis que les géants du Web utilisent des moteurs d'indexation propriétaires, dans le monde de l'*open source*, *Apache Lucene*, une bibliothèque d'indexation *full-text* développée en Java s'est fait une grosse réputation, et est devenue aujourd'hui le standard sur lequel se basent les meilleurs moteurs d'indexation. C'est le cas d'Elasticsearch, lui aussi basé sur *Apache Lucene*, qui est aujourd'hui un des meilleurs moteurs d'indexation du marché.

2.2 Du côté technique

Sous le capot, Elasticsearch est développé en **Java**, et fonctionne donc sur toutes les plateformes qui disposent d'une JVM. Pour interagir nativement avec Elasticsearch, les interfaces disponibles sont l'**API Java** et le format **JSON**. Le moteur d'indexation a de quoi communiquer aisément avec un cluster Big Data grâce à son connecteur **Hadoop** disponible en téléchargement sur le site officiel. Le moteur sait aussi se connecter aux bases de données relationnelles et NoSQL.

2.3 Des fonctionnalités inédites

2.3.1 La réplication des données

Dans un cluster[^cluster] Elasticsearch, lorsque vous avez plusieurs nœuds[^noeud], les données stockées sur ces derniers sont répliquées entre eux. Ceci permet entre autres de conserver l'intégralité des données en cas de perte d'un nœud.

La réplication est faite de manière automatique. Rajouter un nœud ou un shard[^shard] déclenche la réplication automatique.

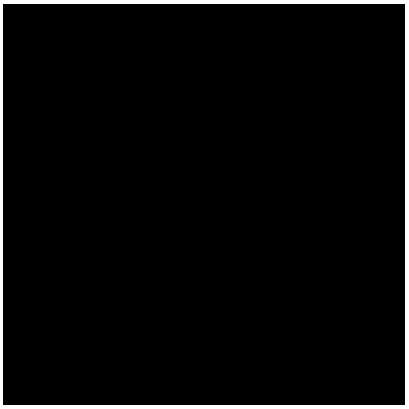


Figure 2.1 – Un cluster Elasticsearch avec 4 nœuds, 5 shards et 2 répliquas, depuis le plugin Head

2.3.2 La recherche en temps réel et contextuelle

La recherche dans Elasticsearch est l'une des plus performantes du marché. On parle de recherche distribuée. Quand on lance une recherche sur le nœud principal, ce dernier va renvoyer la recherche sur les autres nœuds et les résultats seront renvoyés au demandeur.

L'une des particularités du moteur est qu'il regroupe les éléments indexés en rapprochant selon le contexte de la donnée. Les documents en français par exemple seront regroupés ensemble, pour faire plus vite les rapprochements.

2.3.3 Les facettes

Elasticsearch supporte les facettes, qui sont des regroupements de résultats de recherche. Ce qui permet aux utilisateurs d'avoir une vue agrégée de leurs données. Il existe plusieurs types de facettes disponibles dans Elasticsearch, parmi lesquelles :

- *Filter* : renvoie le nombre de hits correspondant à un filtre.
- *Geo distance* : regroupe les données par intervalle de distance géographique.
- *Query* : renvoie le nombre de hits correspondant à une requête.
- *Terms* : renvoie les termes les plus fréquents.
- *Statistical* : permet de calculer les données de type somme, minimum, moyenne, maximum, variance, etc. sur des données de type numériques.

2.4 Une forte communauté

L'un des atouts majeurs du projet ElasticSearch est sa communauté. Ce qui participe à obtenir un écosystème plutôt intéressant. Bien qu'il y ait moins de 5 vrais contributeurs sur le dépôt officiel, nombreux sont ceux à proposer tout un tas de *plugins* différents pour connecter le moteur d'indexation avec les outils du marché et exploiter au maximum le moteur. C'est ainsi qu'on retrouve des pointures telles que :

- **Logstash** : un outil qui permet de centraliser et d'analyser les *logs* des applications.
- **Kibana** : qui vous permet de visualiser vos *logs* de manière *user-friendly*.
- **Marvel** : un outil de supervision pour votre cluster.

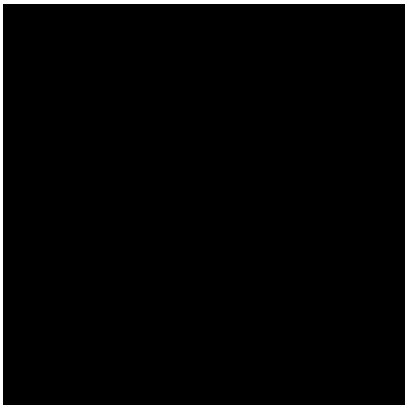


Figure 2.2 – Interface Marvel

À côté de ça, la communauté met à disposition des connecteurs qui permettent d'interagir avec de nombreuses API (Amazon, Azure, Google Twitter, RabbitMQ, MongoDB, Wikipédia, etc.) ainsi qu'avec les langages Groovy, Python, JavaScript, SQL, etc.

Pour ceux qui habitent en France, la communauté Elasticsearch organise très souvent des *meetups* sur Paris. C'est un des meilleurs moyens de vous tenir informé et de rencontrer les utilisateurs d'Elasticsearch.

2.5 Les apports de la version 1.4

L'équipe de développement d'Elasticsearch est très active. Les nouveautés apportées dans la version 1.4 visent essentiellement :

- La stabilité : la découverte des machines du cluster sur le réseau en *Multicast* a été améliorée.
- La consommation mémoire : l'utilisation de la mémoire a été revue et réduite au strict minimum lors d'une requête.
- Les performances.

Des bugs ont aussi été corrigés, dont quelques bugs plutôt contraignants. On avait par exemple le bug qui empêchait de faire une sauvegarde à chaud d'un index en plein chargement.

Pour en savoir plus, je vous invite à lire la note de *release*.

3 Les reproches qui reviennent le plus

3.1 Manque de sécurité

La principale critique que l'on adresse à Elasticsearch est son manque de sécurité (on a le même reproche aussi chez la concurrence). En effet, lorsque vous installez Elasticsearch sur un réseau ouvert, si vous ne mettez pas en place un pare-feu, il sera accessible par tout le monde sur le réseau. Là où ça devient **dangereux**, c'est qu'étant donné qu'il est essentiellement REST, un simple appel à l'API permet de créer ou supprimer des index, sans possibilité de savoir qui a réalisé l'action, car il n'y a aucun processus d'authentification à l'API.

Un simple `curl -XDELETE http://localhost:9200/monindex` peut réduire à néant les travaux d'indexation de plusieurs jours.

Pour pallier ce genre de problème, certaines solutions existent mais ne sont pas toujours satisfaisantes.

— Limiter les accès sur le port 9200 (Http) et 9300 (Transport) aux machines qui ont réellement besoin de se connecter à l'API. Ce qui limite grandement les possibilités de travail.
— Le plugin Jetty, qui permet de limiter les accès en Http (port 9200), mais il reste possible d'attaquer votre cluster via le port 9300.

Cependant, lors de la présentation à Paris de cette dernière mouture, l'équipe Elasticsearch a annoncé travailler sur un outil dédié à la sécurité du moteur d'indexation, du nom de Shield. Il n'est pas encore disponible, mais les fonctionnalités à venir sont alléchantes.

-> <-

3.2 La stabilité

La stabilité du cluster est aussi l'un des points critiqués d'Elasticsearch. Les bonnes pratiques de mise en place d'un cluster ne sont pas toujours très claires, et il n'est pas rare d'observer une séparation d'un cluster lors d'un problème réseau.

La stabilité est aussi un des points sur lesquels se sont penchés les développeurs dans la version 1.4.

4 Les principaux concurrents

4.1 Solr

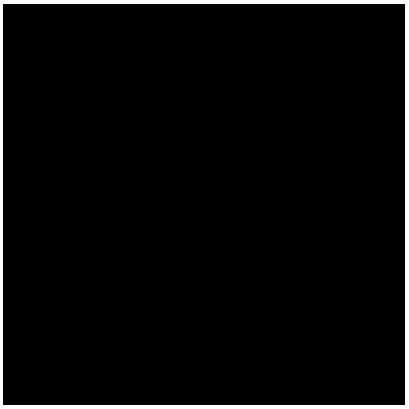


Figure 4.1 –

Solr, est à quelque choses près, identique au moteur Elasticsearch. Basé lui aussi sur *Lucene*, Solr est également un moteur *open source* développé en Java. Les fonctionnalités d'Elasticsearch sont similaires à celles de Solr. Ils souffrent tous les deux du même problème de sécurité.

Depuis la levée de fonds de l'entreprise Elasticsearch, on remarque une nette augmentation des contributions au code du projet Elasticsearch, et une diminution de l'activité du côté de Solr. Elasticsearch a certainement un écosystème plus important que celui de Solr.

Quoi qu'il en soit, Solr est clairement le plus gros concurrent d'Elasticsearch aujourd'hui sur tous les terrains. Pour information, Zeste de Savoir utilise le moteur de recherche Solr.

4.2 Xapian

Xapian est un moteur de recherche *open source* lui aussi. Mais contrairement aux autres, il est écrit en C++. Il y a tout de même un ensemble de *bindgins* pour Python, Ruby, Java, PHP et Perl.

Xapian, n'est pas aussi performant et scalable que Elasticsearch ou Solr, et ne dispose pas des fonctionnalités avancées telles que la vue par facette. Il a tout de même l'avantage d'être assez flexible et il sait indexer autant du contenu Web que du contenu sur le disque dur.

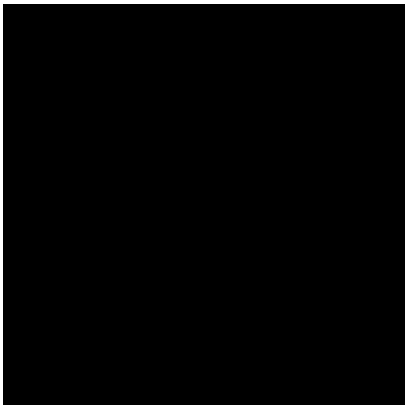


Figure 4.2 –

5 Aller plus loin ?

-  Dépot du projet Elasticsearch
- Site Officiel Elasticsearch
- Télécharger et installer Elasticsearch
- Pourquoi le moteur de recherche d'orange utilise Elasticsearch ?
- La bataille des géants Solr et Elasticsearch

[JVM] : Java Virtual Machine [^cluster] : Un cluster Elasticsearch est un ensemble de nœud. [^noeud] : Un nœud au sens Elasticsearch est une instance du service. [^shard] : Un index Elasticsearch peut stocker une grande quantité de données. Lorsque l'index est trop gros, les recherches se verront ralenties. Elasticsearch permet de diviser un index en plusieurs morceaux appelés shard. *Un* shard* est une instance Lucene qui permet de stocker un document. Par défaut, un index Elasticsearch a cinq *shard*.