

Elastic search

13 personnes

Elastic search comparé a Solr

Les 2 sont basés sur Lucene et fonctionne en REST api

Les 2 sont proches dans l'écriture et le fonctionnement

- > Elastic search permet de définir une hierarchie entre les éléments (arborescence)
- > Solr permet d'effectuer des regroupements

Search api répète le fonctionnement de Solr sur certaine partie, et le fait de facon plus lente et mal (car en php/mysql)

Il convient de l'utiliser uniquement sur l'indexation (et pas la recherche)

Elastic search est beaucoup plus simple a utiliser :

pas de fichier de configuration, il s'organise en fonction du contenu qui lui est envoyé à indexer

installation beaucoup plus claire que solr

"agrégation" de recherche en plus dans elactic (à approfondir)

Toutes recherche :

L'utilisation des multi-coeur permet d'établir plusieurs index différents.

Cela sert par exemple dans le cas d'un site multilingue (spell différents, auto-complétion)

Solr et Elastic search est rapide en lecture.

Pour l'écriture, cela fonctionne de la façon suivante (par default) :

un spool de contenu est créé (file d'attente), et il convient à l'utilisateur de configurer quand elle passera en stade d'indexation (tout les X documents, toutes les x minutes, etc...)

cela nécessite donc un workflow d'indexation pour les gros sites de contenus :

Mise en spool > Indexation > Gestion du cache (purge varnish par ex. sur les contenus ré-indexés)

Pour indexer une page integralement (et non une entité par exemple), on peut utiliser Nutch, qui parcourera un site, et permettra d'indexer le contenu des pages dans Solr.

<https://nutch.apache.org/>

Site comparatif entre les 2 technos : <http://solr-vs-elasticsearch.com/>

L'ergonomie d'Elastic search est plus intuitive que celle de Solr.

Luke : voir comment se comporte Lucene, diagnostiquer

On voit de plus en plus de site affichant des résultats affichés par les moteurs de recherche, et non des informations venant directement des bases de données

Cela permet d'avoir une réponse effectuée par des moteurs plus rapide que Mysql, et surtout pouvant être mis en cluster et en cache.

On a alors un CMS permettant de créer un contenu, et des moteurs permettant l'affichage

rapide de ce contenu.

Dans un rendu de recherche via views, on peut identifier les terms provenant du moteur de recherche, avec ceux qui nécessite le chargement d'une entité :

ils ont affichés entre parenthèses

Résultat :

les 2 moteurs fonctionnent relativement de la même façon, il convient donc d'utiliser le moteur de recherche avec lequel on est le plus à l'aise

Elastic search est beaucoup plus récent, et est surtout beaucoup plus simple à utiliser

L'utilisation d'Elastic search est en fortement en hausse actuellement, et devrait être plébiscitée au sein de la communauté Drupal (présenté au DrupalCon d'Austin en juin)