Moteur d'indexation

Semifir

Enzo Radnaï / Adrien Vossough

https://www.semifir.com/

Introduction

Elasticsearch qui est un **serveur d'indexation** et de recherche des données. Il est basé sur la bibliothèque **Apache Lucene**.

Moteur d'indexation

L'indexation des données dans les bases relationnelles

L'indexation des données dans les bases relationnelles :

Une base de données relationnelle permet d'entreposer des données sous forme de tableaux.
Les données sont ordonnées par ordre d'enregistrement.

id	titre	auteur	Contenu complet
1	Docker et les microservices	Adrien Vossough	Dans ce cours nous allons parler […]
2	Amazon Cloud	Antoine Fissot	Amazon est à l'origine du cloud pour […]
3	Teams et les outils MS Office	Aurélie Dufour	Microsoft Office est un ensemble d'outils […]
4	Certification Java	Enzo Radnaï	Le passage de la certification de niveau […]
5	Le framework Angular	Benoit Evraere	Les problématiques JavaScript pour le […]

L'indexation des données dans les bases relationnelles

La recherche de documents se passe ainsi :

```
SELECT *
FROM cours
WHERE titre='Le framework Angular';
```

cours

id	titre	auteur	Contenu complet
1	Docker et les microservices	Adrien Vossough	Dans ce cours nous allons parler […]
2	Amazon Cloud	Antoine Fissot	Amazon est à l'origine du cloud pour […]
3	Teams et les outils MS Office	Aurélie Dufour	Microsoft Office est un ensemble d'outils […]
4	Certification Java	Enzo Radnaï	Le passage de la certification de niveau […]
5	Le framework Angular	Benoit Evraere	Les problématiques JavaScript pour le […]

La base de données va rechercher parmi toutes les entrées celles qui correspondent à la recherche.

Le problème est dans le cas où nous avons énormément d'entrées dans la table, la rechercher deviendra fastidieuse. (1 millions d'entrées, 1 millions de tests)

L'indexation des données dans les bases relationnelles

Pour accélérer une recherche, il existe l'indexation de colonnes.
La base de données va créer une copie d'une colonne dans une structure, généralement en arbre B (B-Tree)

id	titre	auteur	Contenu complet
1	**Docker et les microservices**	Adrien Vossough	Dans ce cours nous allons parler […]
2	**Amazon Cloud**	Antoine Fissot	Amazon est à l'origine du cloud pour […]
3	**Teams et les outils MS Office**	Aurélie Dufour	Microsoft Office est un ensemble d'outils […]
4	**Certification Java**	Enzo Radnaï	Le passage de la certification de niveau […]
5	**Le framework Angular**	Benoit Evraere	Les problématiques JavaScript pour le […]

```
SELECT *
FROM cours
WHERE titre='Le framework Angular';
```

Amazon Cloud	2
Certification Java	4
Docker …	1
Le framework Angular	5
Teams et les outils …	3

Amazon Cloud
Docker …
Teams et les outils MS Office

L'indexation des données dans les bases relationnelles

L'indexation permet un gain de temps considérable lors de recherches.

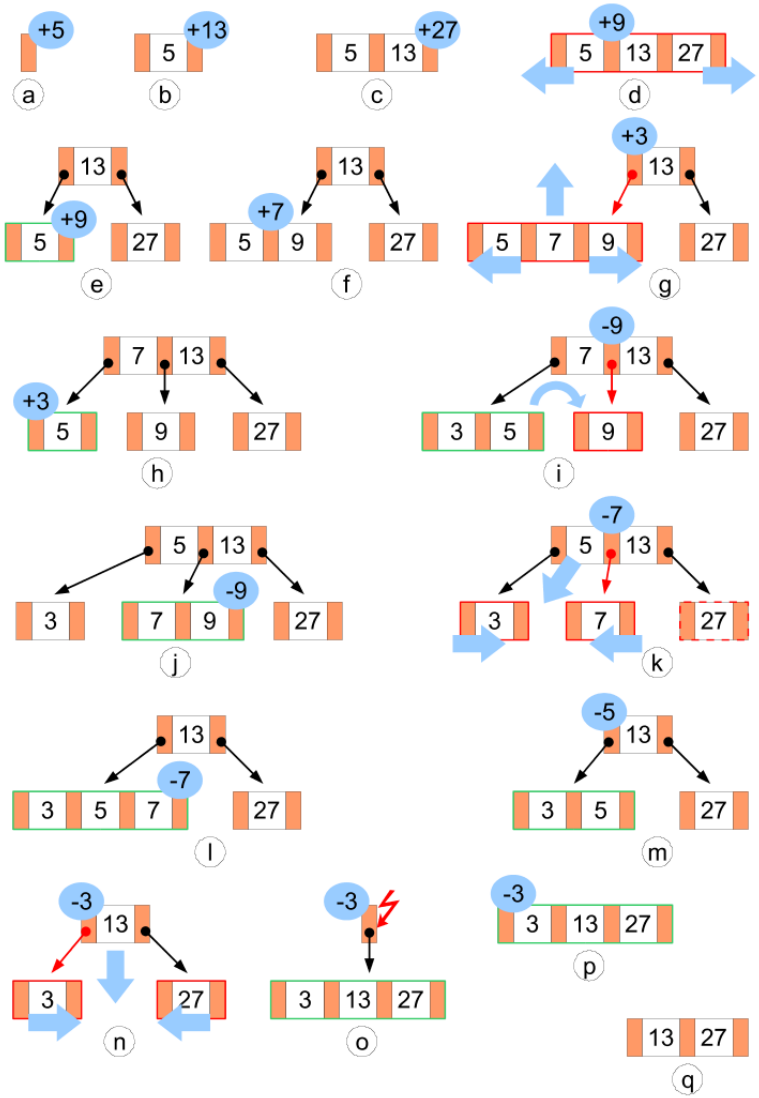
Nombre de ligne	Nombre d'itération	Gain en %
5	3	40
10	4	60
50	6	88
100	7	93
500	9	98,2
1 000	10	99
5 000	13	99,74
10 000	14	99,86
50 000	16	99,968
100 000	17	99,983

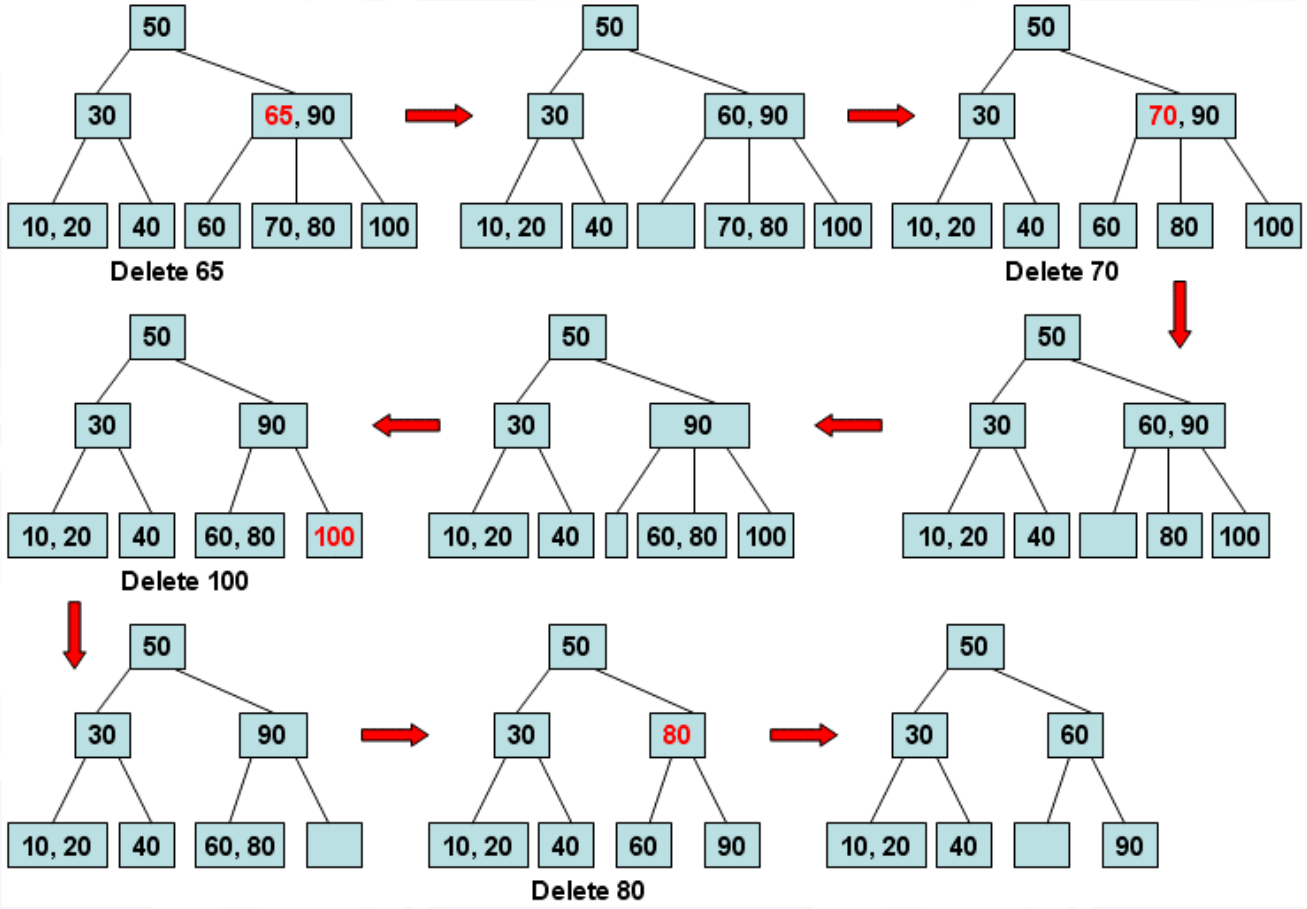
Inconvénients :

- Chaque requête en écriture est ralentie car les données sont dupliquées et rangées dans l'arbre B
- Cela fait gonfler la base de données

L'indexation des données dans les bases relationnelles

Exemple d'ajout ou suppression d'un élément dans un arbre B :





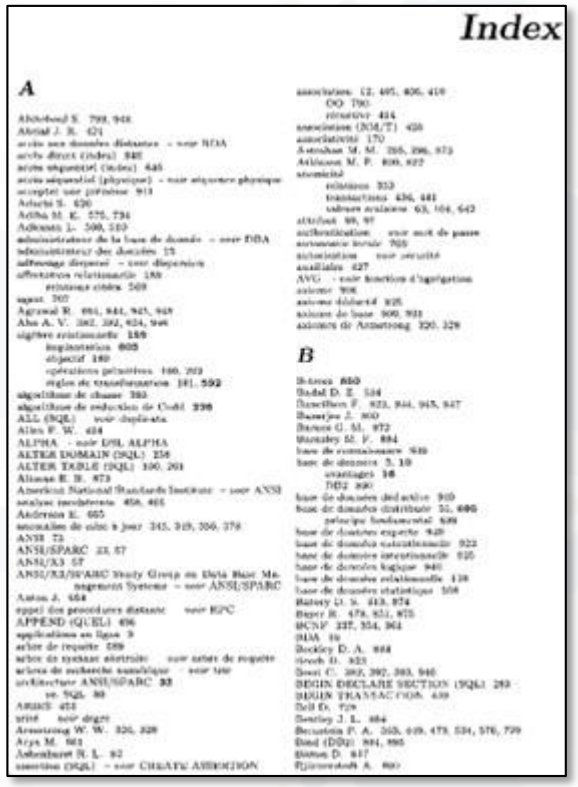
L'indexation inversée

Une indexation inversée permet de retrouver un ensemble de document grâce à une information.

Documents

Récupération du texte

Découpage des mots



Retrouver les documents contenant le mot football sera rapide.

L'indexation inversée : construction

Construction de l'index inversée :

Documents

1	Le langage Java propose deux bibliothèques dédiées à la conception d'interfaces graphiques.
2	Java SWING a été conçue pour pallier les principales insuffisances de la bibliothèque Java AWT.

Extraction des termes

Terme	Id
Le	1
langage	1
Java	1
propose	1
deux	1
bibliothèques	1
dédiées	1
à	1
la	1
conception	1
d'interfaces	1
graphiques	1

Terme	Id
Java	2
SWING	2
a	2
été	2
conçue	2
pour	2
pallier	2
les	2
principales	2
insuffisances	2
de	2
la	2
bibliothèque	2
Java	2
AWT	2

L'indexation inversée : construction

Termes non triés

Terme	Id
Le	1
langage	1
Java	1
propose	1
deux	1
bibliothèqu	1
dédiées	1
à	1
la	1
conception	1
d'interfaces	1
graphiques	1
Java	2
SWING	2
a	2
été	2
conçue	2
pour	2
pallier	2
les	2
principales	2
insuffisances	2
de	2
la	2
bibliothèque	2
Java	2
AWT	2

Tri

Termes triés

Terme	Id
à	1
bibliothèqu	1
conceptior	1
dédiées	1
deux	1
d'interface	1
graphique	1
Java	1
la	1
langage	1
Le	1
propose	1
a	2
AWT	2
bibliothèque	2
conçue	2
de	2
été	2
insuffisances	2
Java	2
Java	2
la	2
les	2
pallier	2
pour	2
principales	2
SWING	2

Regroupement des termes

Termes regroupés

Terme	Id	fréquence
a	2	1
à	1	1
AWT	2	1
bibliothèque	2	1
bibliothèques	1	1
conception	1	1
conçue	2	1
de	2	1
dédiées	1	1
deux	1	1
d'interfaces	1	1
été	2	1
graphiques	1	1
insuffisances	2	1
Java	1	1
Java	2	2
la	1	1
la	2	1
langage	1	1
Le	1	1
les	2	1
pallier	2	1
pour	2	1
principales	2	1
propose	1	1
SWING	2	1

L'indexation inversée : construction

Termes identiques groupés

Terme	Id	fréquence
a	2	1
à	1	1
AWT	2	1
bibliothèque	2	1
bibliothèques	1	1
conception	1	1
conçue	2	1
de	2	1
dédiées	1	1
deux	1	1
d'interfaces	1	1
été	2	1
graphiques	1	1
insuffisances	2	1
Java	1	1
Java	2	2
la	1	1
la	2	1
langage	1	1
Le	1	1
les	2	1
pallier	2	1
pour	2	1
principales	2	1
propose	1	1
SWING	2	1

Création de l'index final

Index

Terme	{id, fréquence}
a	{2, 1}
à	{1, 2}
AWT	{2, 1}
bibliothèque	{2, 1} {1, 1}
conception	{1, 1}
conçue	{2, 1}
de	{2, 1}
dédiées	{1, 1}
deux	{1, 1}
d'interfaces	{1, 1}
été	{2, 1}
graphiques	{1, 1}
insuffisances	{2, 1}
Java	{1, 1} {2, 2}
la	{1, 1} {2, 1}
langage	{1, 1}
Le	{1, 1}
les	{2, 1}
pallier	{2, 1}
pour	{2, 1}
principales	{2, 1}
propose	{1, 1}
SWING	{2, 1}

L'indexation inversée : Recherche

Recherche dans l'index de :

Bibliothèque ET langage

Index

Terme	{id, fréquence}
a	{2, 1}
à	{1, 2}
AWT	{2, 1}
bibliothèque	{1, 1} {2, 1}
conception	{1, 1}
conçue	{2, 1}
de	{2, 1}
dédiées	{1, 1}
deux	{1, 1}
d'interfaces	{1, 1}
été	{2, 1}
graphiques	{1, 1}
insuffisances	{2, 1}
Java	{1, 1} {2, 2}
la	{1, 1} {2, 1}
langage	{1, 1}
Le	{1, 1}
les	{2, 1}
pallier	{2, 1}
pour	{2, 1}
principales	{2, 1}
propose	{1, 1}
SWING	{2, 1}

Document 1 et 2

Document 1

Résultat :
Document 1

Il est possible de retirer des mots dont la fréquence est trop élevées pour accélérer la recherche :
l', d', le, la, les, du, vous, il, mon, ton, son, …

L'indexation inversée : Recherche limitation

L'index précédent fonctionne pour la recherche de mots simples, mais dans le cas de groupes de mots ?

Dans le cas d'une recherche tel que " elastic stack", nous :

- Amazon ***Elastic*** Compute Cloud has a ***stack*** *of technologies…*

Ce résultat ne correspond pas à nos attentes.

L'indexation inversée avec n-gramme : Recherche

Il existe la méthode des "**n-gramme"** ou le texte est transformé ainsi :

Le langage Java propose deux bibliothèques dédiées à la conception d'interfaces graphiques.

Le langage	**langage Java**	**Java propose**	**propose deux**
deux bibliothèques	**bibliothèques dédiées**	**dédiées à**	**…**

Les mots sont doublés entre les signes de ponctuation.
Chaque mot et bi-gramme sont référencés.

Terme
à
à la
bibliothèques
bibliothèques dédiées
conception
conception d'interfaces
…

Une recherche de plus de 2 mots tel que :
"Propose deux bibliothèques" **devient** "Propose deux" AND " bibliothèques"

Uni-gramme : 1 mot
Bi-gramme : séquence de 2 mots

L'indexation inversée avec n-gramme : Recherche

Pour accélérer la recherche, il est possible encore une fois d'extraire les mots trop fréquents, mais cela ne change pas le fait qu'avec la technique des "n-grammes", l'index devient très volumineux.

Si nous prenons 200 000 termes uniques avec tous les n-grammes entre 1 et 5 mots, nous arrivons à $3,2 \times 10^6$ entrée.

L'indexation inversée avec proximité : Recherche

Les moteurs de recherches utilisent aujourd'hui un indice de position, cela permet de déduire la proximité de deux mots dans un même document.

Exemple d'index avec la position :

terme	{document: position}
à	{1 : 8} {2 : 1,5}
bibliothèques	{1 : 6, 14} {2 : 18}
conception	{1 : 10, 15}
dédiées	{1 : 7}
deux	{1 : 5}
d'interfaces	{1 : 11, 18}
graphiques	{1 : 12}
Java	{1 : 3, 16} {2 : 12, 19}
La	{1 : 9, 13} {2 : 17}
Langage	{1 : 2, 17} {2 : 4, 25}
Le	{1 : 1}
propose	{1 : 4}

L'utilisateur peut définir une proximité des mots ou laisser le moteur lui donner des résultats le plus proche d'une requête tel que : "Le langage Java"

Le résultat sera en priorité le **document 1** car les 3 mots ont un delta moins élevé que celui du document 2

L'indexation inversée avec proximité : Recherche

Les moteurs de recherches utilisent aujourd'hui un indice de position, cela permet de déduire la proximité de deux mots dans un même document.

Exemple d'index avec la position :

terme	{document: position}
à	{1 : 8} {2 : 1,5}
bibliothèques	{1 : 6, 14} {2 : 18}
conception	{1 : 10, 15}
dédiées	{1 : 7}
deux	{1 : 5}
d'interfaces	{1 : 11, 18}
graphiques	{1 : 12}
Java	{1 : 3, 16} {2 : 12, 19}
La	{1 : 9, 13} {2 : 17}
Langage	{1 : 2, 17} {2 : 4, 25}
Le	{1 : 1}
propose	{1 : 4}

L'utilisateur peut définir une proximité des mots ou laisser le moteur lui donner des résultats le plus proche d'une requête tel que : "Le langage Java"

Le résultat sera en priorité le **document 1** car les 3 mots ont un delta moins élevé que celui du document 2