

# NoSQL Matters Paris 2015

Posted on [mars 31, 2015](#)



Ayant gagné une place par le groupe utilisateurs Elasticsearch (que je remercie une nouvelle fois), j'ai eu l'opportunité d'assister pour la première fois à une conférence dédiée au NoSQL. Parmi la centaine de participants, je devais sans nul doute être le plus néophyte. Certes, je connais relativement bien Elasticsearch pour l'avoir mis en œuvre, mais l'écosystème d'Hadoop et des bases de données NoSQL restait pour moi encore très vague.

Ce fut donc l'occasion rêvée d'approfondir mes connaissances sur le sujet et de m'aérer l'esprit.

Cerise sur le gâteau, cette conférence m'aura permis d'apprécier la qualité de speaker de deux anciens collègues, Bruno Guedes et Lucian Precup.

## Keynote

La keynote d'ouverture « *NoSQL : The Good, the Bad and the Ugly* » fut brillamment animée par **Rob Harrop** que je connaissais pour être le co-fondateur de SpringSource. Après avoir donné sa propre définition du NoSQL, Rob s'est attaché à expliquer ce qu'il y'avait de bon et de moins bon dans ce mouvement.

Voici les idées clés que j'ai retenues :

- La diversité du choix et la sophistication des solutions peu à la fois être une force et une faiblesse.
- Une fois n'est pas coutume, l'innovation est venue de l'industrie et non dans la recherche fondamentale / académique.

Nous utilisons des cookies pour vous garantir la meilleure expérience sur notre site web. Si vous continuez à utiliser ce site, nous supposons que vous en êtes satisfait.

OK

retrouver, Rob nous invite à se référer aux articles de Kyle Kingsbury / Jepsen qui torture et pousse à bout de nombreuses solutions NoSQL.

Pour s'initier progressivement au monde du NoSQL, Rob nous conseille de commencer par quelque chose de modeste, comme du cache distribué ou de la recherche fulltext. Par sécurité, le dual run peut alors être envisagé. Rob insiste également sur le fait de d'abord essayer de tirer partie au maximum de sa base de données relationnelle. Et pour cela, je ne peux que vous conseiller l'excellent ouvrage [SQL Performance Explained](#).

Se sont ensuite enchaînées différentes conférences portant sur Hadoop, Spark, Elasticsearch, MongoDB, Druid ou bien encore Cassandra.

## Le retour du SQL



Un des thèmes récurrent de cette conférence fut paradoxalement le **SQL**. En effet, contrairement au monde des bases de données relationnelles, aucun standard de requêtage n'a émergé pour les bases NoSQL. Certes, des frameworks tels qu'[Hibernate OGM](#) ou [Spring Data](#) permettent d'avoir un cadre commun. Mais encore faut-il appréhender leurs API. Lorsqu'ils souhaitent faire parler ces données, le Data Scientist a besoin d'un langage expressif. Souvent venu de la BI, il est familiarisé avec le SQL. Relativement récents, [Hadoop With Queries \(HAWQ\)](#) et [Spark SQL](#) offrent désormais la possibilité d'interroger des péta-octets de données en utilisant le SQL.

Pour nous convaincre de son efficacité, Duy Hai Doan, évangéliste chez Datastax, a réalisé une démo dans laquelle il indexait des tweets en JSON dans une base Cassandra puis les requêtait avec [Spark SQL](#).

Nous utilisons des cookies pour vous garantir la meilleure expérience sur notre site web. Si vous continuez à utiliser ce site, nous supposons que vous en êtes satisfait.

OK

Hive supportant la norme ANSI SQL-92. Associé à Hortonworks, [Pivotal est en train de rendre HAWQ Open Source](#) au sein du consortium [Open Data Platform](#) (ODP).

Au cours de sa présentation « [Back to the future : SQL 92 for Elasticsearch ?](#) », Lucian Precup, CTO d'[Adelean](#), nous explique que de nombreux utilisateurs d'Elasticsearch se demandent comment réécrire telle ou telle requête SQL avec Elasticsearch. Le cas classique est un développeur qui cherche à migrer ses services métiers de recherche implémentés jusqu'alors en SQL ou en HQL / Hibernate Criteria.

Or, à ce jour, Elasticsearch ne supporte pas (encore ?) la syntaxe SQL. Pour interroger Elasticsearch, le développeur doit passer par une de ses API : REST / JSON, Java, JavaScript ...

De la même manière qu'en SQL, il existe plusieurs moyens de requêter le moteur de recherche. Lucian prend l'exemple de 2 requêtes ES renvoyant le même résultat mais dont la 2<sup>ème</sup> est 100x plus rapide que la 1<sup>ère</sup> car le filtre est appliqué avant la recherche.

A partir d'exemples, Lucian montre ensuite les requêtes ES équivalentes aux agrégations (sum, avg, count), aux clauses group by et having et aux jointures du SQL. Pour une requête SQL, jusqu'à 4 opérations ES sont nécessaires afin d'obtenir le même résultat.

### Migrer ou réécrire ?

La session « [From SQL to NoSQL in less than 40mn](#) » animée par Tugdual Grall, évangéliste MongoDB, a particulièrement retenu mon attention. En effet, elle abordait les raisons qui poussent à migrer vers une base NoSQL : haute performance, disponibilité de 99,999% (<10 mn / an) mais surtout, l'effort et les difficultés qu'il fallait consentir pour passer d'une base de données relationnelles à une base NoSQL. Migrer d'une base Oracle vers Mongo n'a rien à voir avec une migration Oracle vers MySQL.

Une telle migration impacte en effet toutes les couches d'une architecture :

- **Storage** : les bases de données relationnelles sont généralement stockées sur un équipement de type SAN alors que les bases de données NoSQL utilisent des disques durs locaux, espace de stockage peu cher.
- **JDBC** : drivers différents, API propriétaire, pool de connections non fournis par le serveur d'application.
- **Transactions** : pas de transactions sur plusieurs documents avec Mongo. Pas forcément gênant car là où il faut 4 requêtes SQL pour insérer un Produit et 3 Caractéristiques, il ne sera nécessaire d'insérer qu'un seul document JSON.
- **SQL / ResultSet** : pas de SQL avec Mongo, pas de données retournées sous forme d'un tableau et pas de JOIN.
- **ORM** : possible avec Hibernate OGM, mais pas nécessaire avec Mongo
- **POJO** : attention au code technique qui pourrait s'y trouver. Une architecture logicielle bien pensée, avec des couches de service et d'accès aux données atténue l'effort de migration.

Nous utilisons des cookies pour vous garantir la meilleure expérience sur notre site web. Si vous continuez à utiliser ce site, nous supposons que vous en êtes satisfait.

OK

particulièrement ces 3 dernières.

Tug nous fait ensuite réfléchir à la question suivante : **faut-il migrer une application ou tout réécrire ?**

Il n'y a pas de réponse type. Les 2 scénarios doivent être envisagés et chiffrés.

Une migration sera privilégiée pour une application bien architecturée et comportant peu de tables. Une réécriture donne l'opportunité de changer d'architecture (ex : API REST API).

Pour nous aider à prendre une décision, Tug nous montre quelques outils sur lesquels nous appuyer :



## Bilan

En conclusion, cette 8ième édition de NoSQL matters m'aura permis de rencontrer et d'échanger avec des experts du NoSQL et du BigData. A ma grande surprise, je n'y ai croisé aucun DBA. A croire que le monde du NoSQL n'intéresse majoritairement que les développeurs et quelques sys admin ?

De cette journée, mon seul regret est ne pas avoir pu participer à la session pratique qui se déroulait la veille dans les locaux de Zenika. L'année prochaine, qui sait ?

J'aime 0

Tweeter

Ce contenu a été publié dans [Conférence](#) par [Antoine](#), et marqué avec [Elasticsearch](#), [NoSQL](#). Mettez-le en favori avec son [permalien \[https://javaetmoi.com/2015/03/nosql-matters-paris-2015/\]](https://javaetmoi.com/2015/03/nosql-matters-paris-2015/) .

Ce site utilise Akismet pour réduire les indésirables. [En savoir plus sur comment les données de vos commentaires sont utilisées.](#)

Nous utilisons des cookies pour vous garantir la meilleure expérience sur notre site web. Si vous continuez à utiliser ce site, nous supposons que vous en êtes satisfait.

OK