Zeste de savoir

À la découverte de l'univers du Big Data

12 août 2019

Table des matières

S'il y a un concept qui est devenu très à la mode ces dernières années c'est bien le *Big Data*, traduit officiellement en France par **mégadonnées**. Si l'on s'en tient aux termes composants le mot *Big Data*, on pourrait penser à tort que le concept ne s'applique que sur de gros volumes de données. Aujourd'hui on parle d'univers du *Big Data* car il s'agit d'un écosystème d'outils logiciels, qui, mis tous ensemble, répondent à un besoin.

Qu'est-ce réellement le *Big Data* ? Quelles sont ses fondations, ses acteurs et ses champs d'application ? Comment l'écosystème Big Data répond à des problématiques rencontrées aujourd'hui ? Beaucoup de questions auxquelles nous allons apporter des réponses dans la suite de l'article qui se veut volontairement introductif. Ce thème fera l'objet d'une suite d'articles pour détailler un peu plus en profondeur certains aspects de l'écosystème.

1. Big Data, un écosystème de taille

Le stockage et le traitement de l'information ont toujours été un vrai défi dans l'informatique. Au fil du temps les informaticiens ont appris à gérer des fichiers, des répertoires, des bases de données, etc. Le besoin est de plus en plus grandissant, les données ne cessent d'augmenter, les formats de stockage se diversifient, on a de plus en plus besoin de traiter l'information rapidement. C'est donc pour répondre à ces besoins qu'est né le *Big Data*.

> Big data is **high-volume**, **high-velocity** and **high-variety** information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making. Source : Gartner

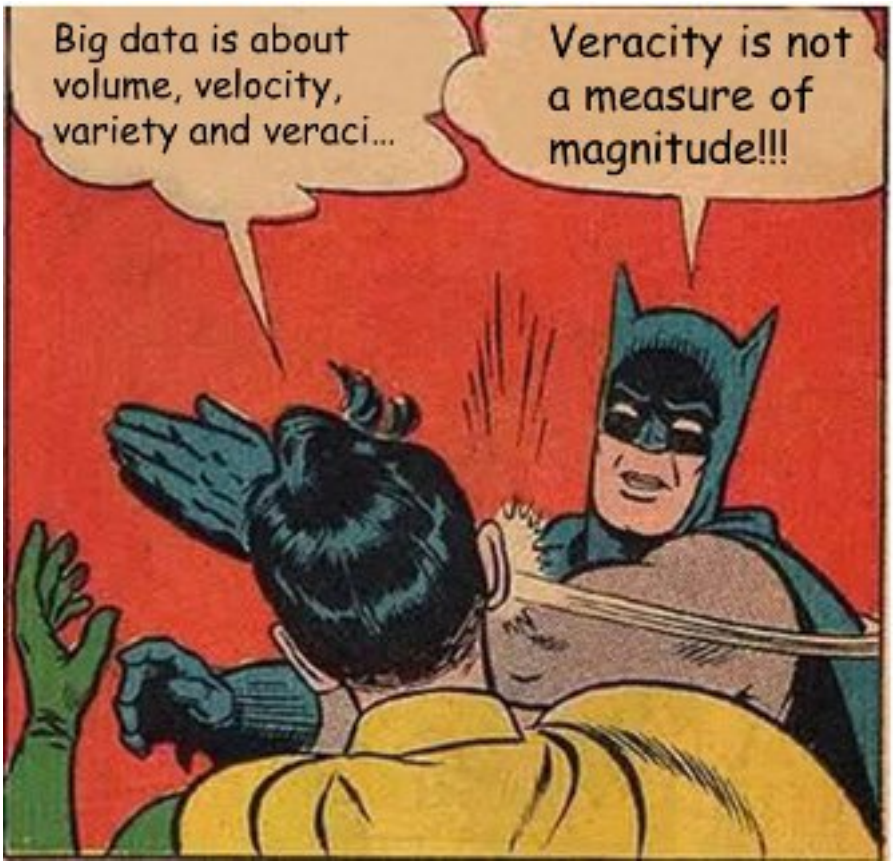

FIGURE 1. – .

Autrement dit, selon Gartner, le *Big Data* est basé sur la règle des trois **V**.

- **Volume** : la gestion de gros volumes de données (on parle ici de milliards d'enregistrements)
- **Vitesse** : le traitement des données quelque soit la vitesse à laquelle elles arrivent
- **Variété** : On passe d'un format structuré (tables, colonnes) à des formats non structurés (documents, vidéos)

Vous allez certainement me dire que les SGBDclassiques savent très bien y faire, et je vous répondrai que non. Avec des SGBDtraditionnels comme MySQL, Oracle, ou encore Teradata, lorsque vous avez de gros flux à traiter, certaines limitations vous sont imposées (système de fichier local, propriétés ACID , rigidité du modèle), ce qui rend les traitements plus lourds.

En mars dernier, le directeur des systèmes d'information de la société AMD annonçait :

> Avec notre plateforme Oracle, nous avons lutté du point de vue performance et fiabilité
> Source : Jake Dominguez

Pour traiter environ 276 téraoctets de données, le SGBDOracle était devenu insuffisant chez AMD, là où la suite Hadoop a séduit par ses performances, sa stabilité, sa scalabilité et son écosystème quasi *opensource*. Un écosystème qui regorge d'outils clés et de concepts tout aussi intéressants les uns que les autres.

1.1. Hadoop, au cœur du Big Data

Conçu à l'origine par Facebook et Yahoo, *Hadoop*, est un *framework opensource* développé aujourd'hui par la fondation Apache. Programmé en langage **Java**, il est conçu pour distribuer de manière efficace d'énormes quantités de données et de traitements sur plusieurs ordinateurs.

Hadoop arrive avec un certain nombre de concepts qu'il faut connaitre pour savoir ce que l'on fait.

1.1.1. Map-Reduce pour le traitement en parallèle

Map-Reduce est une architecture qui facilite la distribution et la répartition des traitements de données sur plusieurs nœuds d'un *cluster*[1]. Il est, comme son nom l'indique composé de deux étapes majeures :

— *Map* : C'est l'étape pendant laquelle les données à traiter ainsi que les traitements à effectuer sont répartis sur les nœuds.
— *Reduce* : C'est l'étape pendant laquelle chaque nœud remonte le résultat des traitements pour qu'ils soient consolidés.

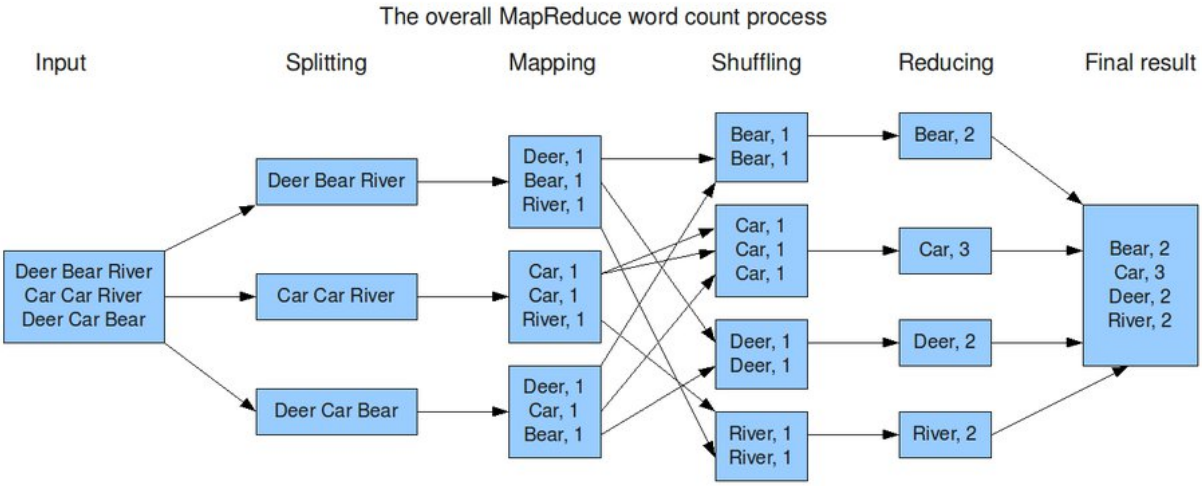

FIGURE 1. – Déroulement de l'algorithme Map-Reduce

Dans l'exemple illustré par l'image ci-dessus, on voit que pour calculer le nombre de mots contenus dans le texte saisi, le programme va appliquer l'algorithme de *Map-Reduce*. Il répartit plus ou moins équitablement l'ensemble des données sur chaque nœud, les nœuds vont effectuer

de manière indépendante le traitement (nombre de mots) et l'étape de *reduce* va se charger de consolider les résultats.

On profite donc au maximum de la puissance de calcul de chaque nœud. Ce qui permet de répondre aux trois critères de *V* du *Big Data*.

1.1.2. Le système de fichiers HDFS

HDFS est un système de fichier au même titre que FAT, NTFS, ext4, etc. Il fait partie de la famille des systèmes de fichier dit distribués (sur les nœuds d'un cluster) et/ou réseaux, car il permet le partage des fichiers entre différentes machines.

Il est très adapté à la réplication de fichiers de grande taille car il utilise des tailles de blocs plus élevées que les systèmes de fichiers classiques (64Mo par défaut et paramétrable).

1.1.3. Le stockage avec HBase

Très inspiré de Google *BigTable*, *HBase* est une base de donnée conçue d'après le modèle NoSQL , et intégré au sein d'Hadoop. Les systèmes NoSQL à l'inverse des bases de données relationnelles classiques (Mysql, Oracle, PostgreSQL, etc.), ont la particularité de rendre la gestion de la donnée plus flexible et plus scalable grâce à son système de stockage sous forme de clé/valeurs.

HBase utilise le système HDFS, et il n'y a aucune garantie qu'il fonctionne correctement avec un autre système de fichier.

1.1.4. Les packages incontournables

Même s'il est possible de se servir uniquement d'Hadoop pour faire du *Big Data*, ça reste tout de même fastidieux. Hadoop n'est qu'un ensemble de package *jar*, livré sans interface homme-machine, où la configuration réseau doit être faite en modifiant les fichiers manuellement. Impossible de faire de la supervision (*start*, *stop*, *restart*) des *jobs* en cours d'exécution sur les *clusters*, et la manipulation des données dans HBase via son *shell* n'est pas triviale.

C'est pour toutes ces raisons que de nombreux packages, développés essentiellement en Java, ont vu le jour et constituent aujourd'hui les incontournables de la suite *Big Data*.

- Ambari , Hue , Whirr : pour la configuration et le monitoring des clusters Hadoop à l'aide d'une interface homme-machine.
- Oozie , Zookeeper : pour la coordination des tâches
- Mahout , MLlib/MLBase : qui permettent d'implémenter des algorithmes de *machine learning*
- Impala , Drill , Shark : pour exécuter du SQL interactif
- Hive , Pig , Spark , Cascading : des surcouches de HBase
- Sqoop , Flume : pour le transfert de données
- Storm : pour la gestion des données en temps réel
- Knox : qui permet de sécuriser les environnements Hadoop

Cette liste n'est pas exhaustive, mais elle a le mérite de présenter un bon nombre de packages qui vous seront très certainement utiles dans la gestion de votre périmètre *Big Data*.

2. Le Big Data aujourd'hui

2.1. Les cas d'usages du Big Data

Si les géants du web (à cause du volume de données traitées) s'étaient déjà confrontés au problème, ce n'était pas encore le cas des entreprises dites classiques. Le besoin commence à se faire ressentir pour des cas assez triviaux.

— **L'analyse de sentiment sur un nouveau produit** : c'est le cas des entreprises à forte concurrence et/ou qui sortent de nouveaux produits régulièrement. C'est très important pour un service marketing de savoir comment a été reçue sur les réseaux sociaux et en temps réels l'arrivée d'un nouveau produit. Ce sont des problématiques solvables aujourd'hui grâce au groupe Hadoop, Mahout et Storm .
— **L'analyse de l'audience d'un site** : Sur un site comme Zeste de Savoir, bien exploitée, l'analyse de l'audience en temps réel permet d'en apprendre un peu plus sur le comportement d'un membre du site et lui faire des suggestions de tutoriels ou d'articles en rapport avec son profil. Là encore, Mahout (et ses algorithmes) est indétrônable.
— **La traçabilité des colis** : pour les entreprises de logistique, c'est un véritable défi de gérer la problématique de suivi d'un colis en temps réel. Avec du *Big Data* ça deviendrait plus simple d'affiner la traçabilité (suivi GPS) et de pouvoir trouver de meilleurs itinéraires en temps réel.

Plus généralement, aujourd'hui en France les entreprises de télécommunications, les chaînes de télévision, la logistique et même les agences de statistiques se mettent au Big Data. Seuls les traders ne sautent pas le pas, à cause de la forte variabilité de leurs données.

2.2. Les acteurs du Big Data

Aujourd'hui, rares sont ceux qui installent tous ces outils à la main. Avec le temps, des distributions qui *packagent* Hadoop et ses packages sont nées. On distingue trois grands et très connus :

— Mapr (2011)
— Hortonworks (2011)
— Cloudera (2008)

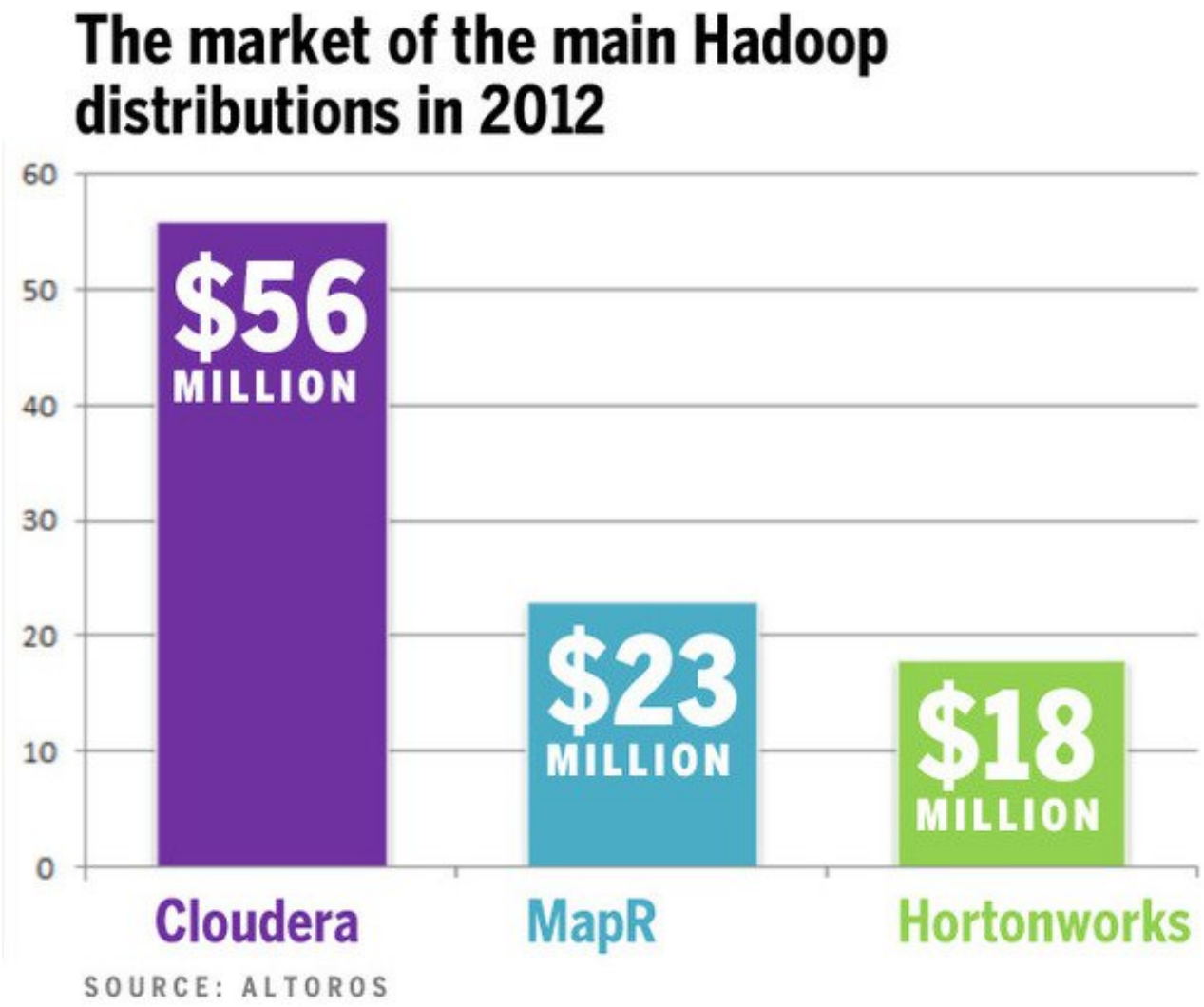

FIGURE 2. – Part de Marché des Acteurs du Big Data

Ce sont des distributions qui ont fait des choix d'architecture et intègrent chacune un ensemble de packages *Hadoop*. Les diverses distributions vous assurent l'intégration (et donc la cohérence) des paquets *Hadoop*. Etant donné que les versions des paquets évoluent rapidement, la compatibilité avec le reste de l'écosystème n'est pas toujours garantie entre chaque mis à jour d'un paquet. De plus *Hadoop* de base, ne gère pas les problématiques de réplication/sauvegarde des données, alors qu'on peut retrouver ces mécanismes dans la distribution Mapr par exemple.

Toutes les distributions ont chacune leurs spécificités et s'adressent à des publics différents. Si vous devez faire un choix de distribution, vérifiez d'abord qu'elle a tout ce qu'il vous faut en prérequis y compris avec les bonnes versions des outils. Si on prend par exemple le cas de la distribution Hortonworks, elle a fait le choix d'inclure le maximum de *packages* de l'écosystème *Hadoop*, ce qui ralentit souvent les mises à jour de la distribution elle même.

Le choix de la distribution est donc très important au départ, car elle oriente votre architecture ainsi que vos choix conceptuels.

À la lecture de cet article, nous savons désormais ce qui se cache sous le terme *Big Data*, ses points clés ainsi que son écosystème. À travers les cas d'usage, nous pouvons confirmer que le

Big Data a su répondre à un besoin et certains l'ont compris il y'a très longtemps. Cependant, les distributions Hadoop sont encore considérées comme lourdes, et on ne les retrouve guère dans l'univers de la mobilité. C'est peut-être là sa future orientation.

3. Pour en savoir plus

— Hadoop official Web Site
— Comparing the top Hadoop distributions

1. Encore appelé grappe de données, il désigne un point de regroupement de plusieurs serveurs que l'on appelle des nœuds.

Liste des abréviations

HDFS Hadoop Distributed File System. 4

NoSQL Not Only SQL. 4

SGBD Système de Gestion de Base de Données. 2