

# Introduction à MapReduce/Hadoop et Spark

Certificat Big Data

Ludovic Denoyer et Sylvain Lamprier

UPMC

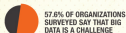


WWW.WFPRO.COM

## BIG DATA

Big Data is data that is too large, complex and dynamic for any conventional data tools to capture, store, manage and analyze.

The right use of Big Data allows analysts to spot trends and gives niche insights that help create value and innovation much faster than conventional methods.



The "Three V's", i.e. the Volume, Variety and Velocity of the data coming in is what creates the challenge.

### VOLUME



### VARIETY

**PEOPLE TO PEOPLE**  
 NETWORKS, VIRTUAL COMMUNITIES, SOCIAL NETWORKS, WEB LOGS...

**PEOPLE TO MACHINE**  
 ARCHIVES, MEDICAL DEVICES, DIGITAL TV, E-COMMERCE, SMART CARDS, BANK CARDS, COMPUTERS, MOBILES...

**MACHINE TO MACHINE**  
 SENSORS, GPS DEVICES, BAR CODE SCANNERS, SURVEILLANCE CAMERAS, SCIENTIFIC RESEARCH...

### VELOCITY

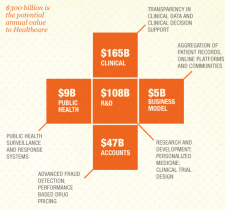
**2.9 MILLION** EMAILS SENT EVERY SECOND

**20 HOURS** OF VIDEO UPLOADED EVERY MIN

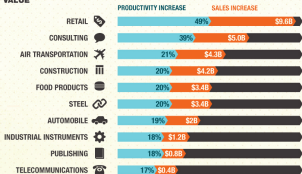
**50 MILLION** TWEETS PER DAY

### CASE STUDY - Healthcare

\$300 billion is the potential annual value to Healthcare



### VALUE



40% PROJECTED GROWTH IN GLOBAL DATA CREATED PER YEAR



5% PROJECTED GROWTH IN GLOBAL IT SPENDING PER YEAR

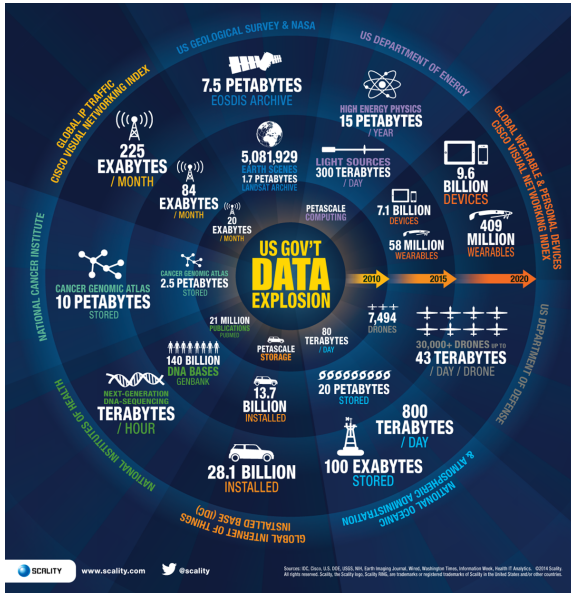
The estimated size of the digital universe in 2011 was 1.8 zettabytes. It is predicted that between 2009 and 2020, this will grow 44 fold to 35 zettabytes per year. A well defined data management strategy is essential to successfully utilize Big Data.

Source: Mapping the Frontiers of Big Data, Morgan Stanley, Big Data: The Next Frontier for Innovation, Disruption and Opportunity, McKinsey Global Institute, Research, Analytics and Insights, 2012; Mapping the Business Impact of Big Data, study by University of Texas, Austin, 2012 Department of Culture

DO BUSINESS BETTER

WFPRO HELPS YOUR COMPANY EMPLOYEES AND CUSTOMERS CONSUME INFORMATION AND DATA EFFICIENTLY





# Data driven science: le 4e paradigme (Jim Gray - Prix Turing)

## SNR 2013

Extrait: "A l'heure actuelle, la science vit une révolution qui conduit à nouveau paradigme selon lequel 'la science est dans les données', autrement dit la connaissance émerge du traitement des données [...]

**Le traitement de données et la gestion de connaissances représentent ainsi le quatrième pilier de la science après la théorie, l'expérimentation et la simulation.** L'extraction de connaissances à partir de grands volumes de données (en particulier quand le nombre de données est bien plus grand que la taille de l'échantillon) , l'apprentissage statistique, l'agrégation de données hétérogènes, la visualisation et la navigation dans de grands espaces de données et de connaissances sont autant d'instruments qui permettent d'observer des phénomènes, de valider des hypothèses, d'élaborer de nouveaux modèles ou de prendre des décisions en situation critique"

.....

68% des entreprises qui ont systématiquement recours à une analyse de données dans leurs prises de décision voient leurs bénéfices augmenter

\* *selon une étude menée par the Economist Intelligence Unit (2014)*

.....

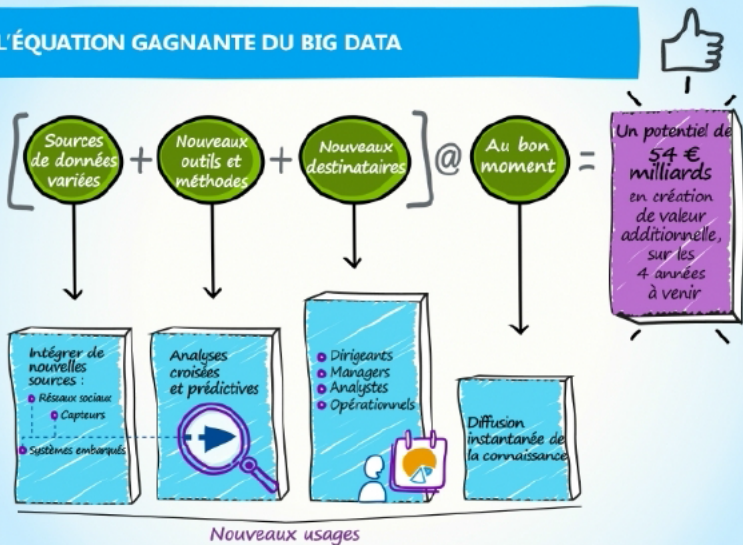
Pour qui réussit à optimiser son usage, la donnée devient **information**, puis, bien partagée au sein de l'entreprise, elle se transforme en **connaissance** et constitue son **savoir**. Elle peut être une source de services et d'innovations, notamment lorsqu'on la croise avec d'autres données et qu'elle provient de sources diverses.

\* *Enjeux Business des données - CIGREF 2014*

La donnée est donc l'un des principaux actifs immatériels de nos organisations, et pour autant, n'est pas encore gérée avec la même rigueur ni les mêmes moyens que les autres ressources, capital et ressources humaines notamment. Dans un contexte où elle est devenue critique pour l'activité de l'entreprise, la mise en place d'une gestion structurée et industrielle de la donnée est impérative.

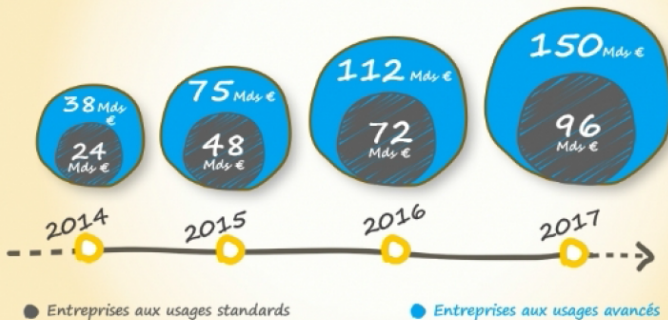
\* *Enjeux Business des données - CIGREF 2014*

## L'ÉQUATION GAGNANTE DU BIG DATA





## ÉVOLUTION DU CUMUL DE VALEURS DE LA DONNÉE (en milliards d'euros)

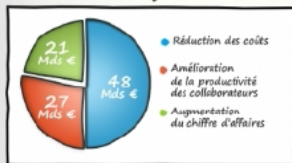


## COMPÉTITIVITÉ DES ENTREPRISES ET DONNÉES

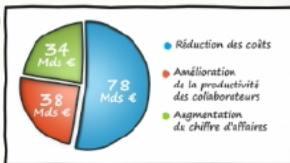


Une meilleure exploitation des données pourrait améliorer la compétitivité des entreprises à plusieurs niveaux

Entreprises aux usages avancés



Entreprises aux usages standards



## Calcul distribué

Un des enjeux concerne le traitement de grandes quantités de données. Ce traitement ne peut être réalisé avec les paradigmes classiques de traitement de données et nécessite l'utilisation de plateformes distribuées de calcul

- Introduction à Hadoop
- Introduction à Spark
- Prise en main des deux plateformes

## Contexte

- Très grand Volume de Données
  - Google = 20 milliards de pages Web = 400 TeraOctets, 30-35 MB/sec (lecture sur disque)  $\Rightarrow$  4 mois de lecture.
- Données Distribuées

## Problématique

Comment effectuer des calculs sur ces grandes masses de données ?

## Problématique

Comment effectuer des calculs sur ces grandes masses de données ?

## Pourquoi faire ?

- Indexation
  - Moteurs de Recherche
  - Indexation d'images
- Reporting:
  - Caractériser les utilisateurs d'un réseau social
  - Détecter la fraude à la CB
  - Statistiques sur des logs de connexions (publicité)
- Analyse:
  - Détection de Communautés
  - Analyse du Churn

## Problématique

Comment effectuer des calculs sur ces grandes masses de données ?

## Map-Reduce

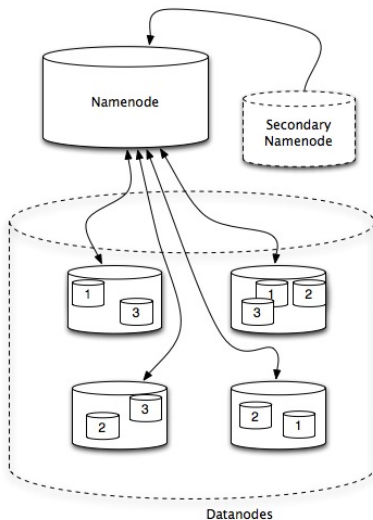
- Map-Reduce a été introduit par Google en 2004
- Map-Reduce est un:
  - Un modèle de programmation,
  - avec un schéma très contraint,
  - qui permet:
    - parallélisation automatique,
    - de l'équilibrage de charge,
    - des optimisations sur les transferts disques et réseaux,
    - de la tolérance aux pannes

- The Yahoo ! Search Webmap is a Hadoop application that runs on a more than 10,000 core Linux cluster and produces data that is now used in every Yahoo ! Web search query.
- Google : the size of one phase of the computation [of the index] dropped from approximately 3800 line of C++ code to approximately 700 lines when expressed using MapReduce.
- Facebook has multiple Hadoop clusters deployed now - with the biggest having about 2500 cpu cores and 1 PetaByte of disk space. We are loading over 250 gigabytes of compressed data (over 2 terabytes uncompressed) into the Hadoop file system every day and have hundreds of jobs running each day against these data sets.
- Hadoop est aussi utilisée par Twitter, Amazon, Rackspace, LinkedIn, IBM, Veoh, Last.fm (en Bash ! !), Microsoft...

- Apache Hadoop
  - Framework distribué
  - Utilisé par de très nombreuses entreprises
  - Traitements parallèles sur des clusters de machines
  - ⇒ Amener le code aux données
- Système de fichiers HDFS
  - Système de fichiers virtuel de Hadoop
  - Conçu pour stocker de très gros volumes de données sur un grand nombre de machines
  - Permet l'abstraction de l'architecture physique de stockage
  - Réplication des données



# Hadoop Distributed File System



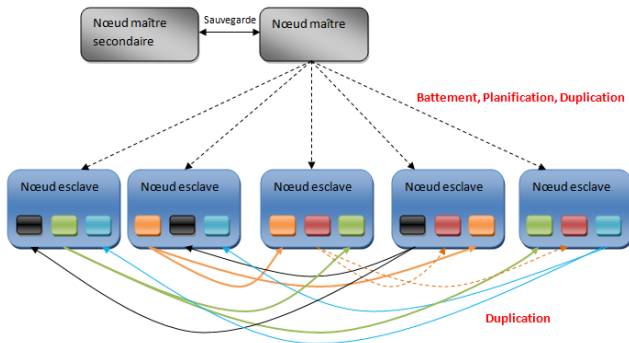
- Architecture de machines HDFS
  - NameNode :

Gère l'espace de noms, l'arborescence du système de fichiers et les métadonnées des fichiers et des répertoires
  - SecondaryNameNode :

Gère l'historique des modifications dans le système de fichiers  
Permet la continuité du fonctionnement du cluster en cas de panne du NameNode principal
  - DataNode :

Stocke et restitue les blocs de données.

# Hadoop Distributed File System



- De nombreux outils basés sur Hadoop
  - MapReduce : Outil de mise en oeuvre du paradigme de programmation parallèle du même nom
  - HBase : Base de données distribuée disposant d'un stockage structuré pour les grandes tables
  - Hive : Logiciel d'analyse de données (initialement développé par Facebook) permettant d'utiliser Hadoop avec une syntaxe proche du SQL
  - Pig : Logiciel d'analyse de données (initialement développé par Yahoo!) comparable à Hive mais utilisant le langage Pig Latin
  - Spark : Framework de traitement de données distribué avec mémoire partagée
- Plateforme d'apprentissage Mahout

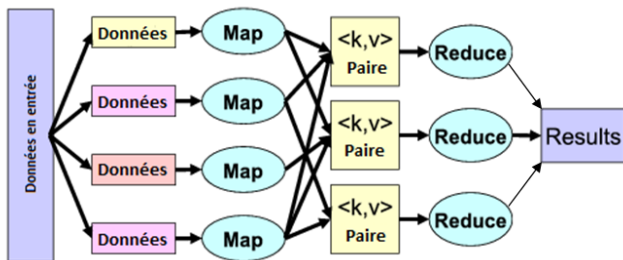
# Qui utilise Hadoop?



... et des centaines d'entreprises et universités à travers le monde.

- Exécution d'un problème de manière distribuée
  - ⇒ Découpage en sous-problèmes
  - ⇒ Execution des sous-problèmes sur les différentes machines du cluster
    - Stratégie algorithmique dite du *Divide and Conquer*
- Map Reduce
  - Paradigme de programmation parallèle visant à généraliser les approches existantes pour produire une approche unique applicable à tous les problèmes.
  - Origine du nom : langages fonctionnels
  - Calcul distribué : "MapReduce: Simplified Data Processing on Large Clusters" [Google,2004]

# Map Reduce



- Deux étapes principales :
  - Map: Emission de paires <clé,valeur> pour chaque donnée d'entrée lue
  - Reduce: Regroupement des valeurs de clé identique et application d'un traitement sur ces valeurs de clé commune

- Ecrire un programme Map Reduce:
  - 1 Choisir une manière de découper les données afin que Map soit parallélisable
  - 2 Choisir la clé à utiliser pour notre problème
  - 3 Écrire le programme pour l'opération Map
  - 4 Écrire le programme pour l'opération Reduce



- Exemple classique : le Comptage de mots
  - Fichiers d'entrée textuels
  - On veut connaître le nombre d'occurrences de chacun des mots dans ces fichiers
- Il faut décider :
  - De la manière dont on découpe les textes
  - Des couples <clé,valeur> à émettre lors du Map appliqué à chaque morceau de texte
  - Du traitement à opérer lors du regroupement des clés communes (Reduce)

Fichier d'entrée :

```
Celui qui croyait au ciel  
Celui qui n'y croyait pas  
[...]  
Fou qui fait le délicat  
Fou qui songe à ses querelles
```

(Louis Aragon, *La rose et le Réséda*, 1943, fragment)

- Pour simplifier, on retire tout symbole de ponctuation et caractères spéciaux. On passe l'intégralité du texte en minuscules.

Découpage des données d'entrée: par exemple par ligne

```
celui qui croyait au ciel
```

```
celui qui ny croyait pas
```

```
fou qui fait le delicat
```

```
fou qui songe a ses querelles
```

- Ici, 4 unités de traitement après découpage

# Map Reduce: WordCount

- Opération map :
  - Séparation de l'unité en mots (selon les espaces)
  - Emission d'une paire <mot,1> pour chaque mot

celui qui croyait au ciel → (celui;1) (qui;1) (croyait;1) (au;1) (ciel;1)

celui qui ny croyait pas → (celui;1) (qui;1) (ny;1) (croyait;1) (pas;1)

fou qui fait le delicat → (fou;1) (qui;1) (fait;1) (le;1) (delicat;1)

fou qui songe a ses querelles → (fou;1) (qui;1) (songe;1) (a;1) (ses;1) (querelles;1)

# Map Reduce: WordCount

- Après le map : regroupement (ou shuffle) des clés communes
  - Effectué par un tri distribué
  - Pris en charge de manière automatique par Hadoop

(celui;1) (celui;1)

(qui;1) (qui;1) (qui;1) (qui;1)

(croyait;1) (croyait;1)

(au;1) (ny;1)

(ciel;1) (pas;1)

(fou;1) (fou;1)

(fait;1) (le;1)

(delicat;1) (songe;1)

(a;1) (ses;1)

(querelles;1)

# Map Reduce: WordCount

- Opération Reduce :
  - Sommation des valeurs de toutes les paires de clé commune
  - Ecriture dans un (ou des) fichier(s) resultats

```
qui: 4
celui: 2
croyait: 2
fou: 2
au: 1
ciel: 1
ny: 1
pas: 1
fait: 1
[...]
```

```
Map(String input_key, String input_values) :  
  foreach word w in input_values:  
    EmitIntermediate( w, "`1"');
```

```
Reduce (String key, Iterator intermediate_values):  
  int result=0;  
  foreach v in intermediate_values:  
    result += ParseInt( v );  
  Emit( key, String( result ));
```

- Plusieurs implémentations existent.
- Dans différents langages (C++, C#, Erlang, Java, Python, Ruby, R, ..)
- La plus connue est Hadoop (de la fondation Apache)



```
public static class TokenizerMapper extends Mapper<
{
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context)
        throws IOException, InterruptedException
    {
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens())
        {
            word.set(itr.nextToken());
            context.write(word, one);
        }
    }
}
```

- Composants d'un processus Map Reduce:
  - 1 Split: Divise les données d'entrée en flux parallèles à fournir aux noeuds de calcul.
  - 2 Read: Lit les flux de données en les découpant en unités à traiter. Par défaut à partir d'un fichier texte: unité = ligne.
  - 3 Map: Applique sur chaque unité envoyé par le Reader un traitement de transformation dont le résultat est stocké dans des paires <clé,valeur>.
  - 4 Combine: Composant facultatif qui applique un traitement de reduction anticipé à des fins d'optimisation. Cette étape ne doit pas perturber la logique de traitement.
  - 5 ...

- Composants d'un processus Map Reduce:
  - 1 ...
  - 2 Group: Regroupement (ou shuffle) des paires de clés communes. Réalisé par un tri distribué sur les différents noeuds du cluster.
  - 3 Partition: Distribue les groupes de paires sur les différents noeuds de calcul pour préparer l'opération de reduction. Généralement effectué par simple hashage et découpage en morceaux de données de tailles égales (dépend du nombre de noeuds Reduce).
  - 4 Reduce: Applique un traitement de réduction à chaque liste de valeurs regroupées.
  - 5 Write: Écrit le resultat du traitement dans le(s) fichier(s) résultats(s). On obtient autant de fichiers resultats que l'on a de noeuds de reduction.

# Map Reduce

