

Fouille de données massives avec Hadoop

Sebastiao Correia
scorreia@talend.com



AAFD'14
29-30 avril 2014



- **Présentation de Talend**
- Définition du Big Data
- Le framework Hadoop
- 3 thématiques
 - Rapprochement des données
 - Détection de fraude
 - Clustering
- **Les futurs outils de fouille de données sur Hadoop**



- Talend propose des outils graphiques pour :
 - L'intégration de données
 - Le traitement des Big Data
 - La qualité de données
 - Le MDM
 - L'intégration d'applications (ESB)
 - La gestion des processus métier (BPM)



Le Studio de développement

- Open source
- Générateur de code
- Extensible
 - Composants DI, DQ, BD, ESB
 - Indicateurs DQ

```
Job cdh5_02simple_matching_separate_output 0.1
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
```

The screenshot shows the Talend Studio interface. At the top, a code editor window displays Java code for a job named "cdh5_02simple_matching_separate_output 0.1". The code includes sections for mapper and reducer definitions, specifically for "tGenKey_1" and "tMatchGroup_1". Below the code editor is a "Designer" view where a data processing flow is visualized. The flow starts with an "tHDFSInput_1" component (Map) reading data from HDFS. This is followed by a "tMap_1" component. The output of "tMap_1" is split into two parallel paths. The first path, labeled "TO_Calc (Main order:1)", leads to a "tAggregateRow_1" component (Map/Reduce). The second path, labeled "MajorAgencies (Main order:2)", leads to a "Uniq Agencies" component (Map). Both paths converge at a "tSortRow_1" component (Map/Reduce), which then feeds into an "tHDFSOutput_1" component (Map/Reduce) for final output.



- Présentation de Talend
- **Définition du Big Data**
- Le framework Hadoop
- 3 thématiques
 - Rapprochement des données
 - Détection de fraude
 - Clustering
- **Les futurs outils de fouille de données sur Hadoop**

Définition *en cours d'élaboration*
en même temps que les techno évoluent
<http://arxiv.org/abs/1309.5821>

- **Gartner** : 3 V ou 5 V
- **Intel** : 300 TB de données générées par semaine
- **Oracle** : extraction de **valeur** des bases de données augmentées de sources de données non structurées
- **Microsoft** : ensembles de données **complexes**
- **NIST**: dépasse les **capacités** des systèmes actuels.

Les 5V du Big Data

Volume

Vitesse

Variété

Valeur

Véracité

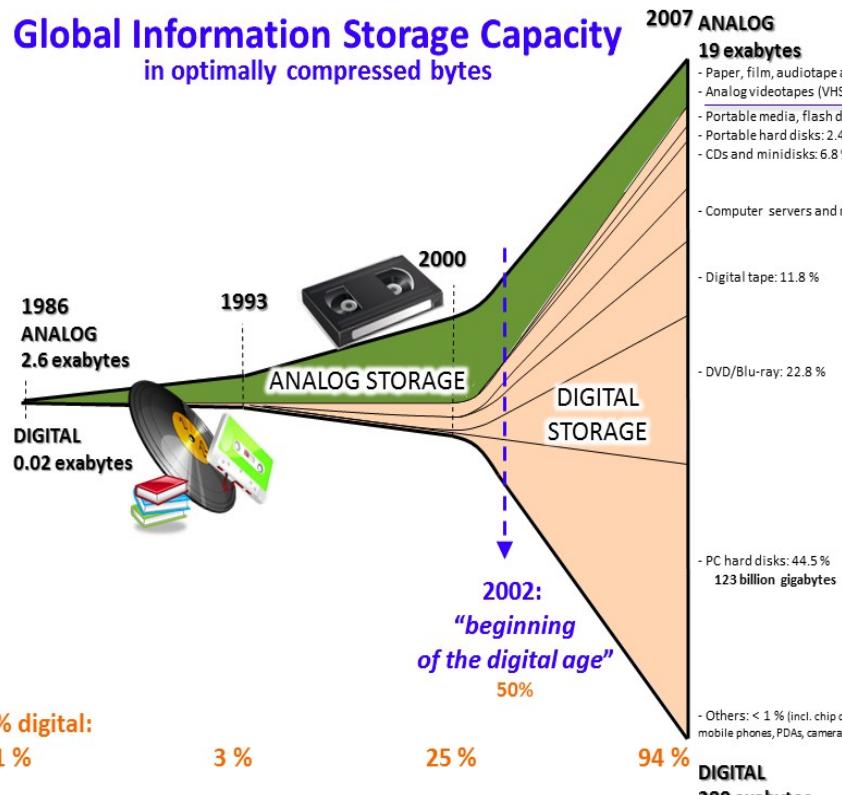
Google trend: “Big Data” associé à Hadoop, NoSQL, Google, IBM et Oracle.

Croissance exponentielle des données



En 2012, 90% des données ont été générées durant les 2 années précédentes.
Chaque jour de 2012, **2.5 Exaoctets** de données sont créés.
<http://www.martinhilbert.net/WorldInfoCapacity.html>

Global Information Storage Capacity
in optimally compressed bytes



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60–65. <http://www.martinhilbert.net/WorldInfoCapacity.html>



Quelques chiffres

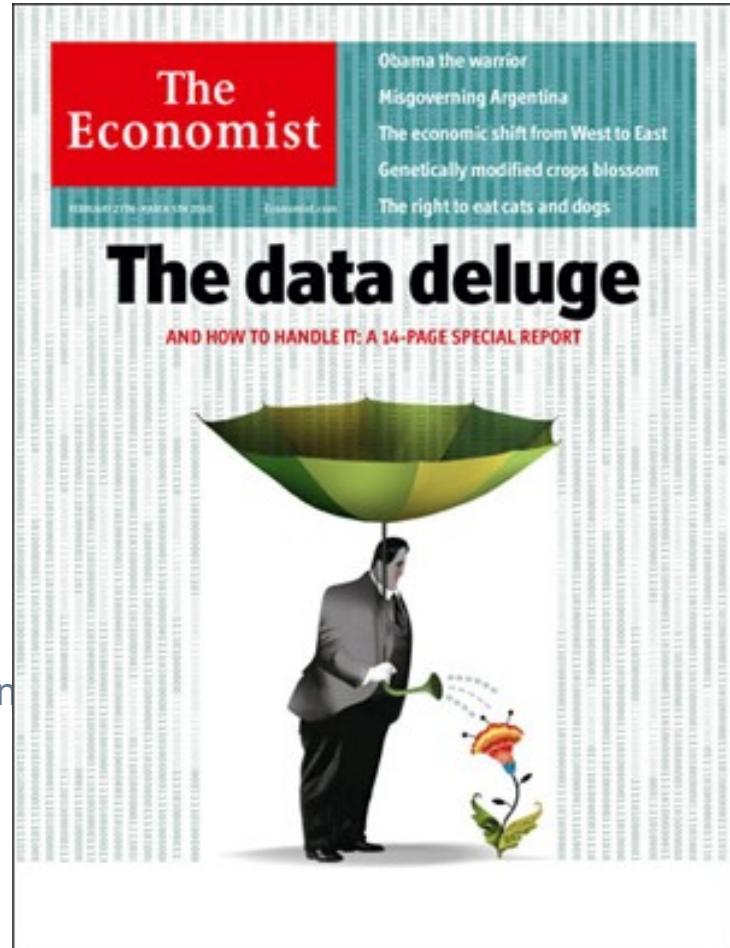
- **Par jour**

- **144.8 milliards** d'Email.
- **340 millions** tweets.
- **684 000 bits** de contenu partagé sur Facebook.

- **Par minute**

- **72 heures** (259,200 secondes) de video sont partagées sur YouTube.
- **2 millions** de recherches sur Google.
- **34 000 "likes"** des marques sur Facebook.
- **27 000** nouveaux posts sur Tumblr.
- **3 600** nouvelles photos sur Instagram.
- **571** nouveaux sites web

- **2.5 Petaoctets** dans les bases de données Wal-Mart
- **40 To** de données générées chaque secondes au LHC
- **25 Po** de données stockées et analysées au LHC chaque année
- **10 To** produits par les capteurs des avions lors d'un vol pendant 30 minutes
- **1.25 To** ce que peut contenir le cerveau humain



Plus encore sur <http://marciaconner.com/blog/data-on-big-data/>



En 2000, le stockage de 1Go coûtait moins de 1\$.

=> Augmentation des capacités de stockage.

Le Cloud a permis une généralisation du Big Data.

De nouvelles technologies sont apparues dès les années 2000 pour gérer la volumétrie et la variété des données :

- Hadoop HDFS
- Map Reduce

commodity hardware
applications **HADOOP**
Unstructured Data
scale-out **NoSQL** analytics
MPP DATABASES



- Un marché de 24 milliards de \$ en 2016
- Taux de croissance annuel de 31.7%
- Entreprises ayant un projet Big Data
 - En France : 10%
 - En Allemagne : 18%
 - Au UK : 33%
- Les technologies de pointe (Etude IDC) :
 - Bases de données objets ou graphiques : 47%
 - L'indexation de contenu : 38%
 - Les bases de données en mémoire : 37%



- Présentation de Talend
- Définition du Big Data
- **Le framework Hadoop**
- 3 thématiques
 - Rapprochement des données
 - Détection de fraude
 - Clustering
- **Les futurs outils de fouille de données sur Hadoop**



• Quelques dates

- 2003 : “The Google File System”, Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung <http://research.google.com/archive/gfs.html>
- 2004 : “MapReduce: Simplified Data Processing on Large Clusters”, Jeffrey Dean et Sanjay Ghemawat
<http://research.google.com/archive/mapreduce.html>
- 2005 : Naissance d'Hadoop chez Yahoo (HDFS et MapReduce), Doug Cutting et Mike Cafarella
- 2006 : “Bigtable: A Distributed Storage System for Structured Data”, Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber
<http://research.google.com/archive/bigtable.html>

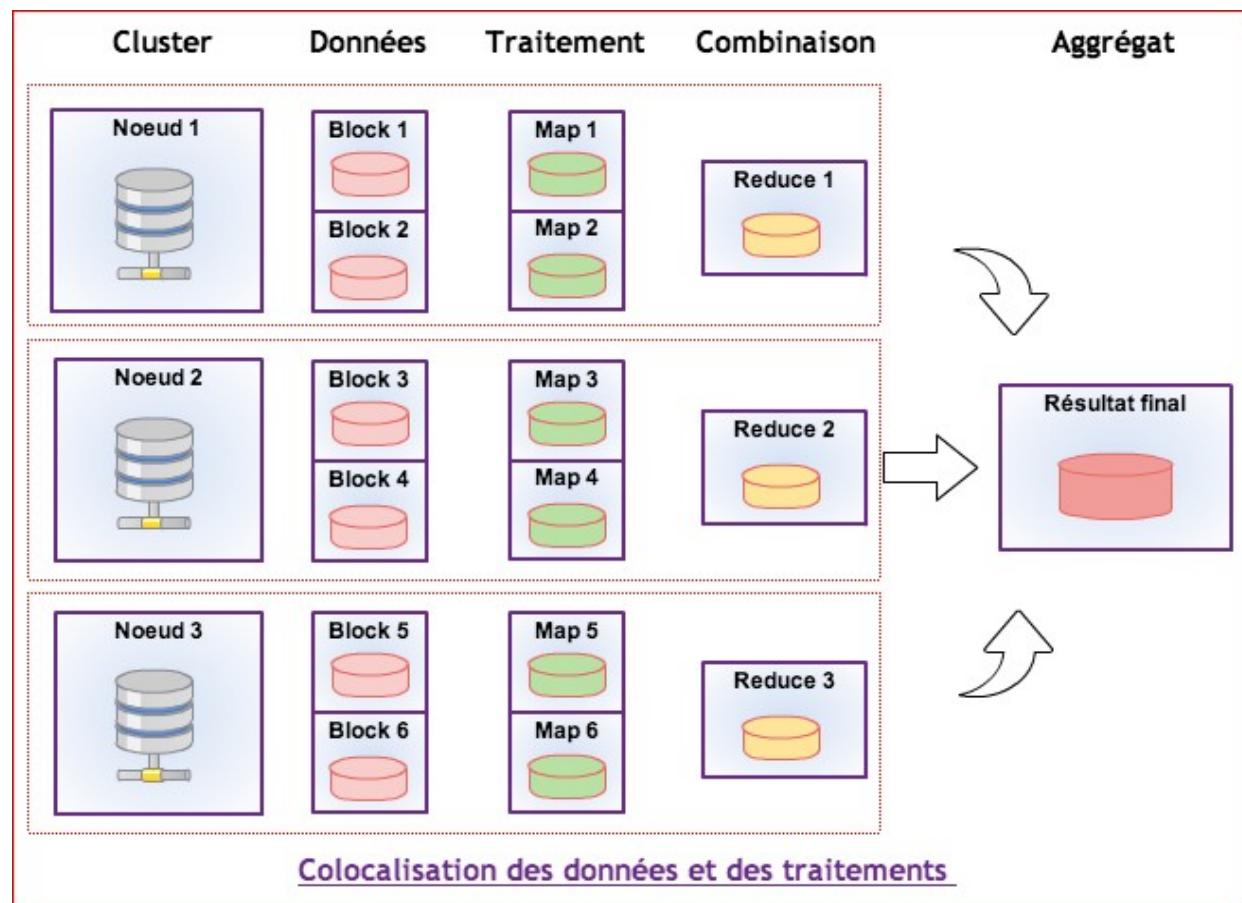


- **projet opensource (Fondation Apache) dédié au calcul distribué, fiable et scalable <http://hadoop.apache.org/>**
 - Hypothèse de départ : les machines ne sont pas fiables
 - Hadoop la haute disponibilité au niveau applicatif (redondance des données entre machines, pertes de connexions, plantages de machines,...)
- **Modules**
 - **HDFS** : Hadoop Distributed File System (inspiré de GFS)
 - **MapReduce** : système pour le traitement parallèle des gros volumes de données (inspiré de Google MapReduce)
 - En version 2 : **YARN** : système de gestion et planification des ressources du cluster



• Localité des Données

- Auparavant les données étaient déplacées dans une application pour être manipulées (SGBD, ETL, Applications...)
- Désormais, les applications (sous forme MapReduce) sont déplacées vers les données

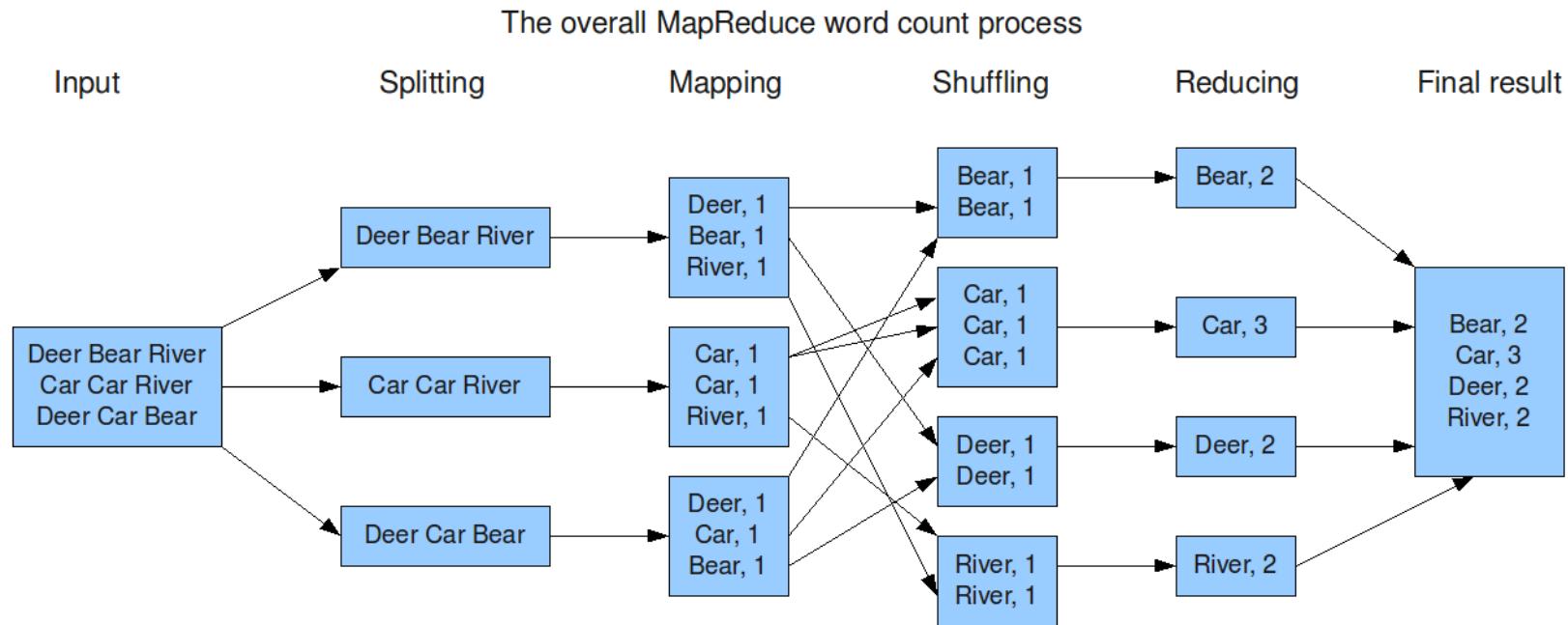




- Un programme MapReduce est composé de 2 fonctions
 - Map() divise les données pour traiter des sous-problèmes
 - Reduce() collecte et agrège les résultats des sous-problèmes
- Fonctionne avec des données sous forme de paires (clé, valeur)
 - $\text{Map}(k1, v1) \rightarrow \text{list}(k2, v2)$
 - $\text{Reduce}(k2, \text{list } (v2)) \rightarrow \text{list}(v3)$



- Exemple avec le décompte de mots





- Présentation de Talend
- Définition du Big Data
- Le framework Hadoop
- **3 thématiques**
 - Rapprochement des données
 - Détection de fraude
 - Clustering
- Les futurs outils de fouille de données sur Hadoop



- Processus permettant d'identifier les enregistrements concernant les mêmes objets

account_num	lname	fname	address1	city	M
12912208437	Ames	Raphael	9662 Red Leaf	Westminster	0
12912208437	Ames	Ralph	9662 Red Leaf	Westminster	1
14398822784	Abdulla	Larry	Wren Avenue 6968	Lemon Grove	0
14398822784	Abdulla	Larry	6968 Wren Ave.	Lemon Grove	1
27192117600	Amaro	Cathy	551 Thors Bay Road	Mill Valley	0
27192117600	Amaro	Kathy	551 Thors Bay Road	Mill Valley	1
96027305126	Alering	Robert	1451 Victory Lane	Salem	0
96027305126	Ahlering	Robert	1451 Victory Lane	Salem	1

Le rapprochement de données

- 2 enregistrements R1 = {a} et R2 = {b}

Iname	fname	address	country
Smith	John	4077 Chinquapin Ct	USA
Smith	Don	8689 St. George Court	USA

- Calcul du score
 $S = P(R1=R2) = \sum_i w_i \times p(a=b)$ avec w_i poids normalisés
 - $S = 1 \Rightarrow$ identité
 $S > T \Rightarrow$ similaires (T = seuil)

Weight on attributes:	1	1							
Normalized Weights:	0.5	0.5							
	lname	fname	address	country	GID	MASTER	score	GRP_QUALITY	ATTR_SCORES
	Smith	John	4077 Chinquapin Ct	USA	2	true		1	0.8611111343
	Smith	Don	8689 St. George Court	USA	0	false	0.8611111343	0	fname: 0.7222222685813904 lname: 1.0

Attribute similarity score

P=1	P=0.722..
-----	-----------

Score=0.5 x 1 + 0.5 x 0.722.. = 0.8611..

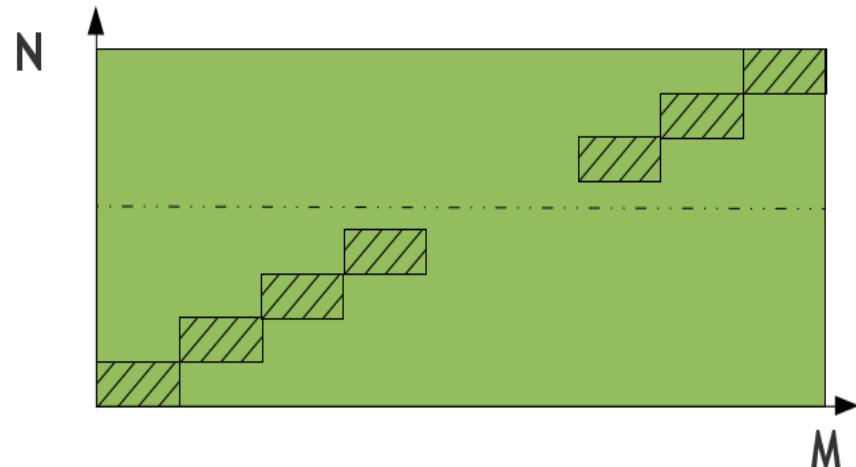


- Nécessité de comparer les enregistrements 2 à 2
- MAIS si N enregistrements à comparer avec M enregistrements, alors $N \times M$ comparaisons
 - Exemple : 1.000 nouveaux clients à comparer aux 10.000 clients référencés => 10.000.000 de comparaisons !!
 - Alors que le nombre de clients déjà référencés dans les 1000 nouveaux est au max $1000 = \min(N,M)$.
 - => 9 999 000 comparaisons inutiles



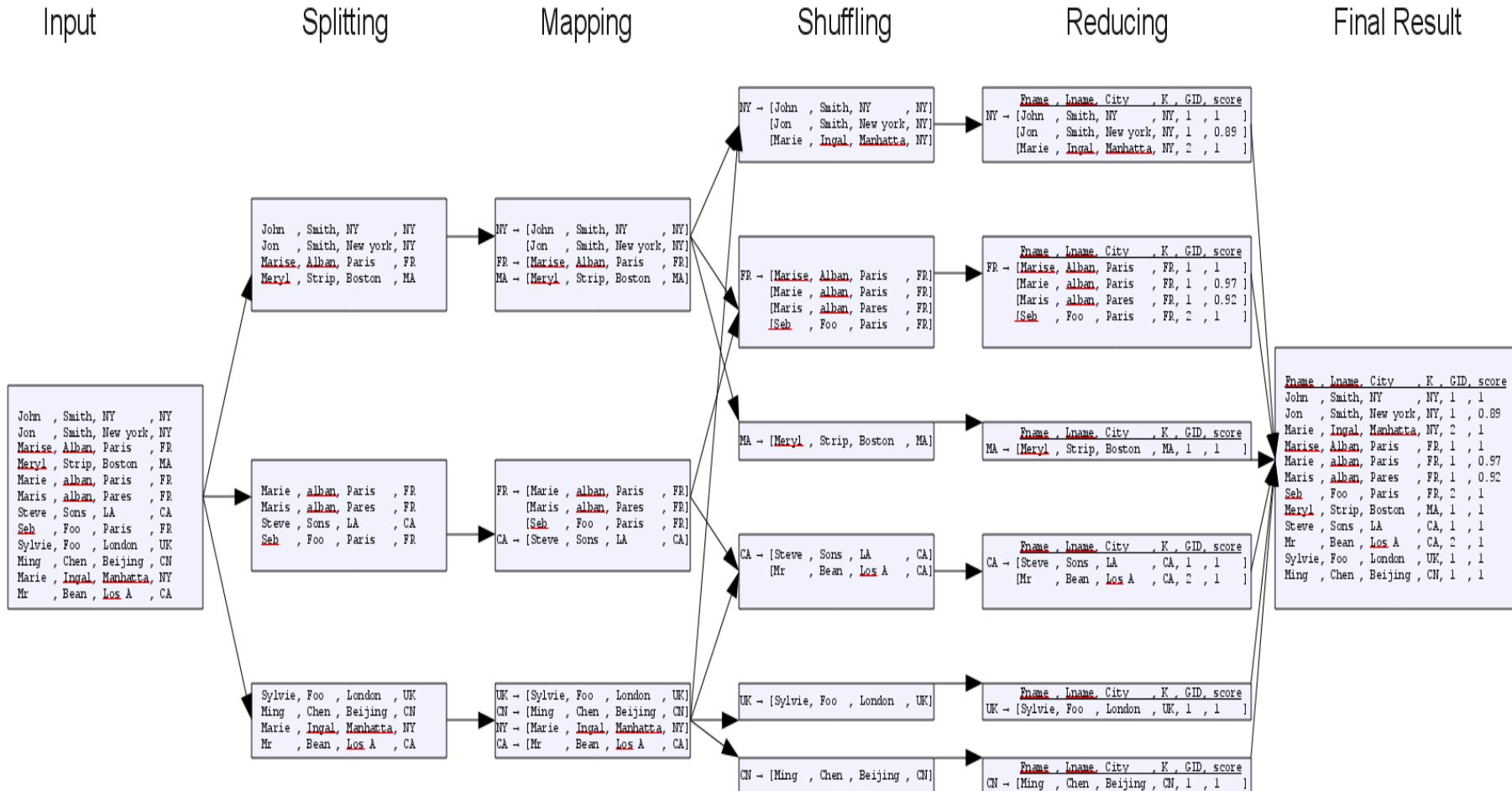
- Optimisation en réduisant le nombre de comparaisons

- Stratégie de “blocking” partitionnement des données

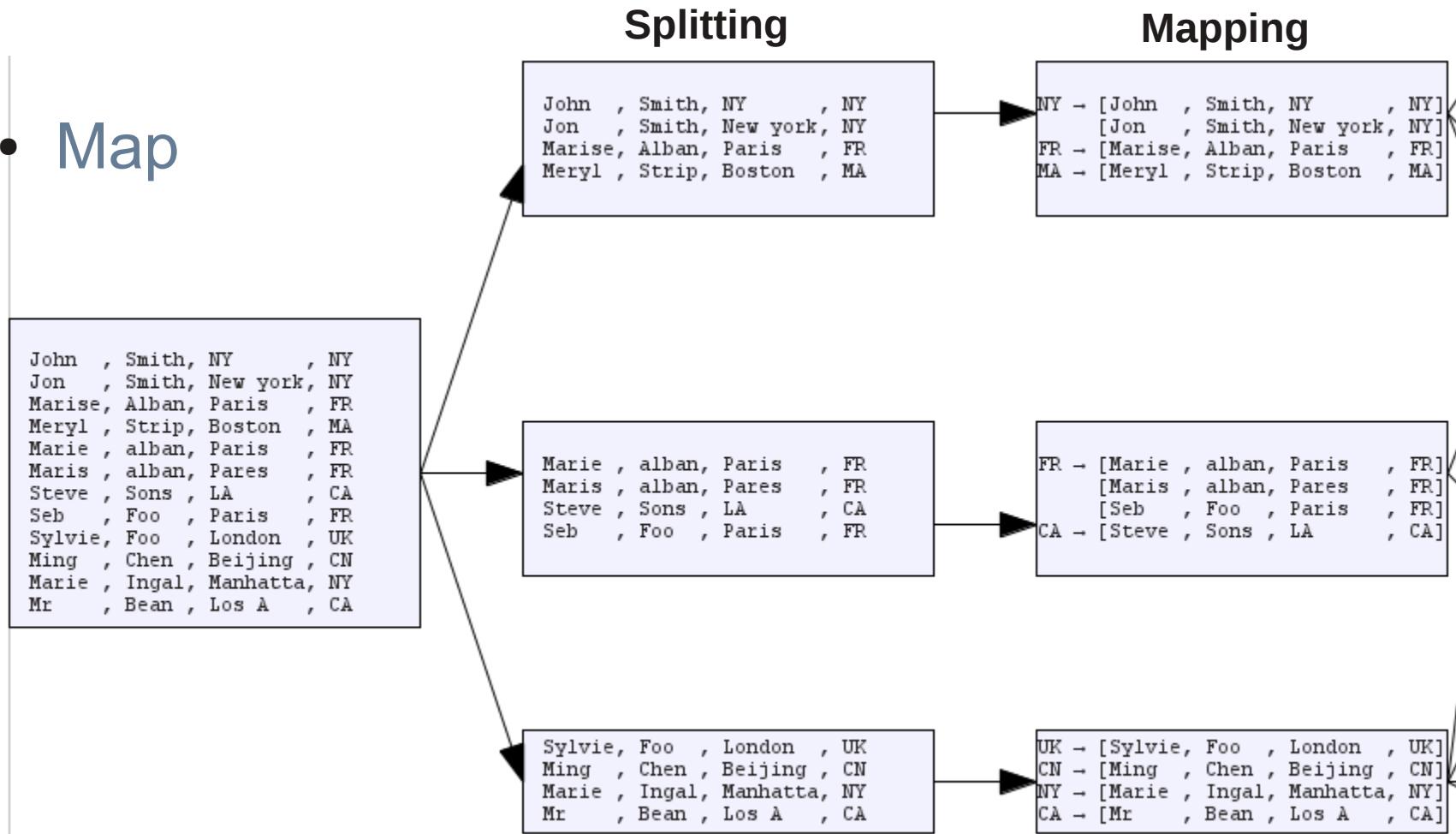


- Exemple :
100 x blocs de 10 enregistrements en entrée à comparer à 100 blocs de 100 enregistrements.
- Nb comparaisons : $100 \times (10 \times 100) = 100\ 000$
- Approche idéale pour Hadoop Map Reduce

Rapprochement avec Hadoop



- Map

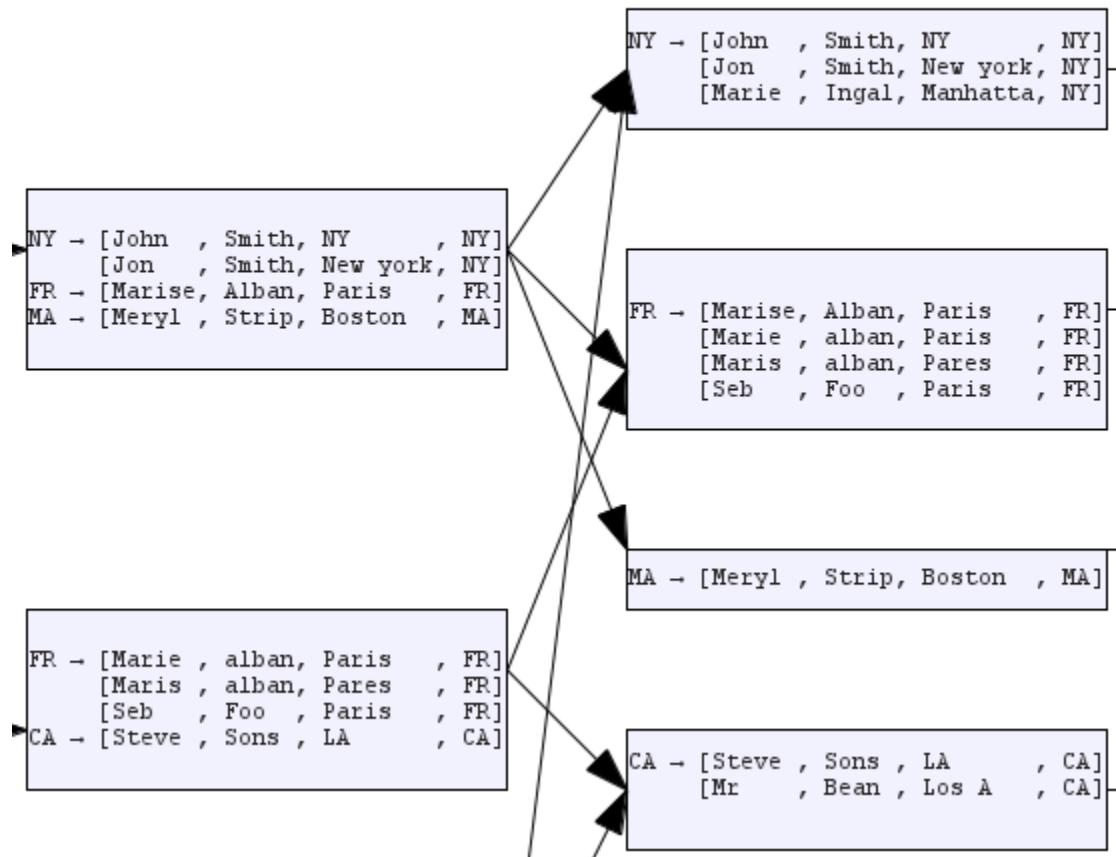




- **Shuffle**

Mapping

Shuffling

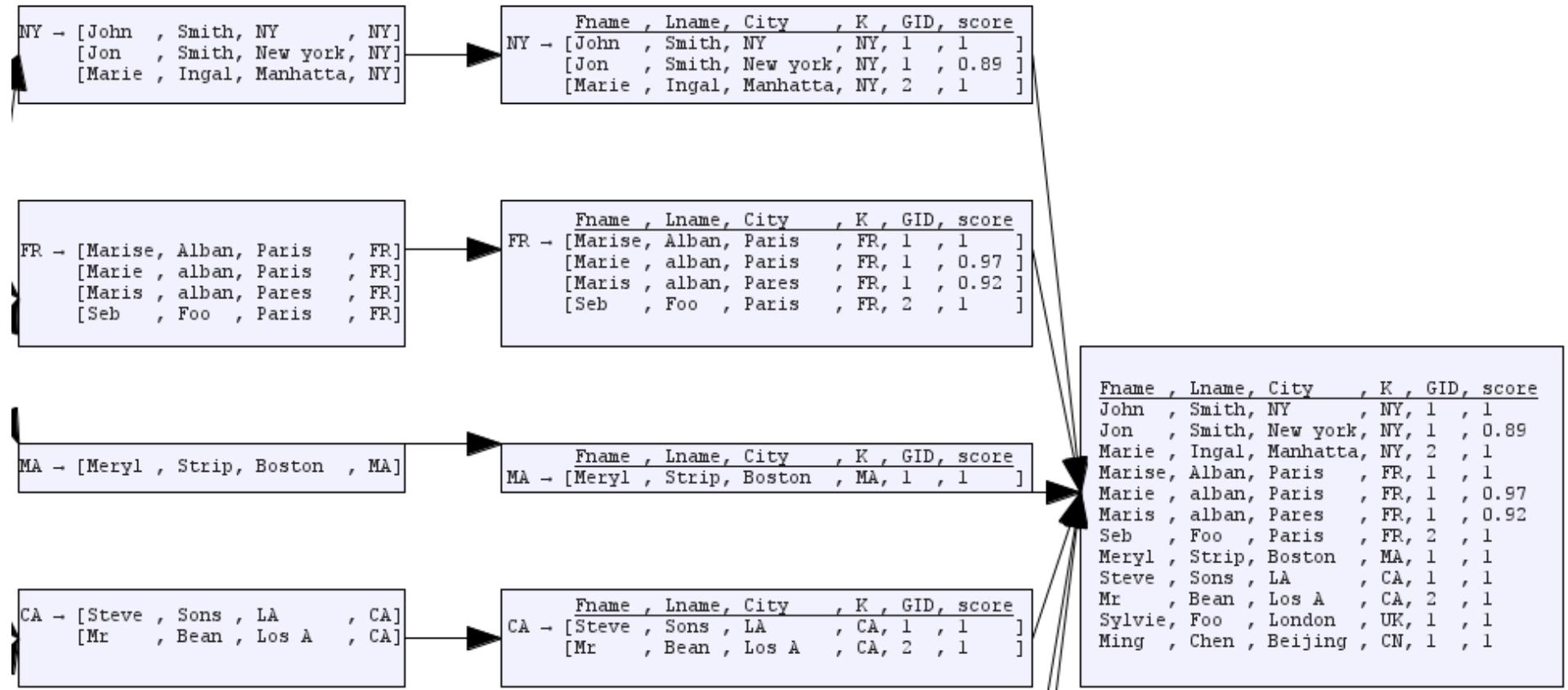


Rapprochement avec Hadoop

- Reduce Shuffling

Reducing

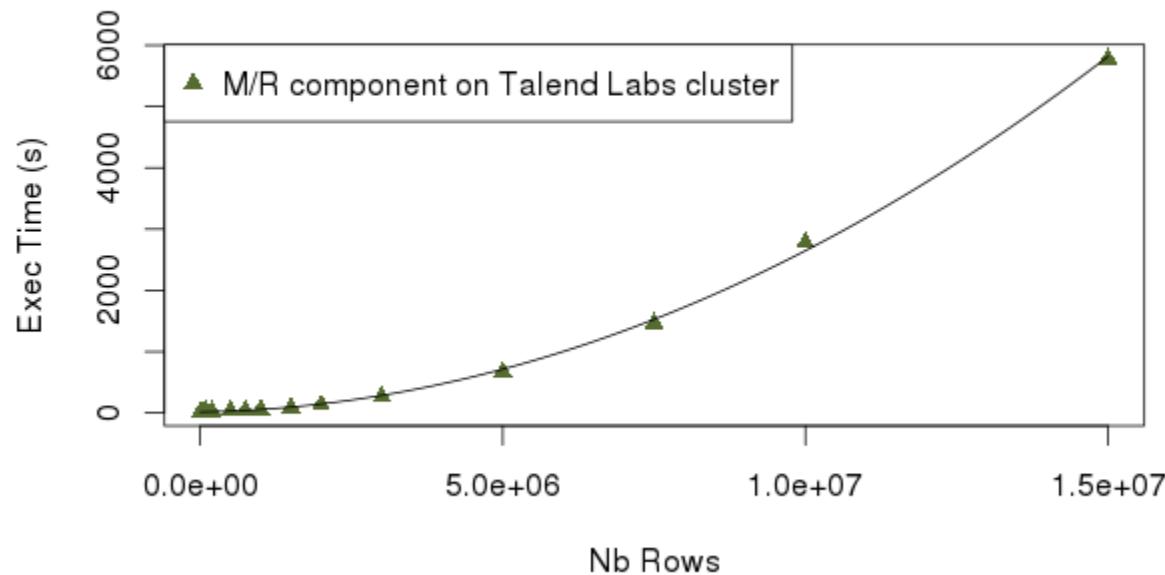
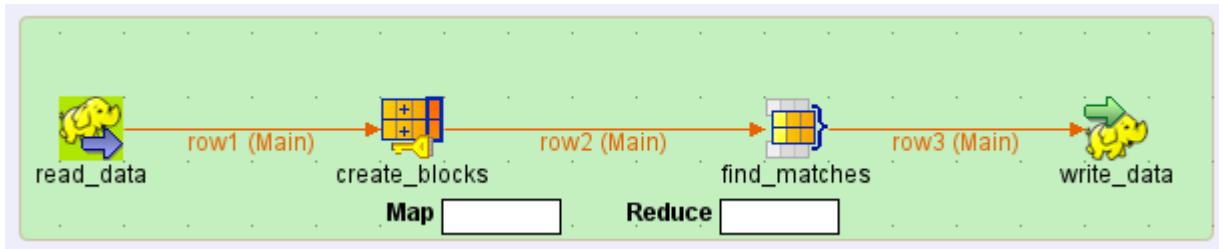
Final Result



Rapprochement avec Talend



- Cluster 9 noeuds (Cloudera CDH 4.5 avec Yarn)



Modèle quadratique dépend de la stratégie de “blocking”
 $t \sim 16 + 1,6 \times 10^{-5} N + 2,5 \times 10^{-11} N^2$



- Présentation de Talend
- Définition du Big Data
- Le framework Hadoop
- **3 thématiques**
 - Rapprochement des données
 - **Détection de fraude**
 - Clustering
- Les futurs outils de fouille de données sur Hadoop



• Loi de Benford

- Loi du 1er chiffre $d \in \{1,2,\dots,9\}$
- Quelle est la répartition de ce 1er chiffre ?

Patrimoine moyen en €	1er chiffre
2,408,869	2
1,925,421	1
2,319,719	2
2,984,232	2
2,325,543	2
2,593,587	2
2,036,967	2
2,392,431	2
2,508,619	2
2,370,514	2
2,393,076	2
3,236,227	3
2,374,940	2
2,201,594	2
2,321,516	2
2,171,647	2



• Loi de Benford

- Loi du 1er chiffre $d \in \{1,2,\dots,9\}$
- Quelle est la répartition de ce 1er chiffre ?

Digit	Fréquence
1	11.11%
2	11.11%
3	11.11%
4	11.11%
5	11.11%
6	11.11%
7	11.11%
8	11.11%
9	11.11%

?

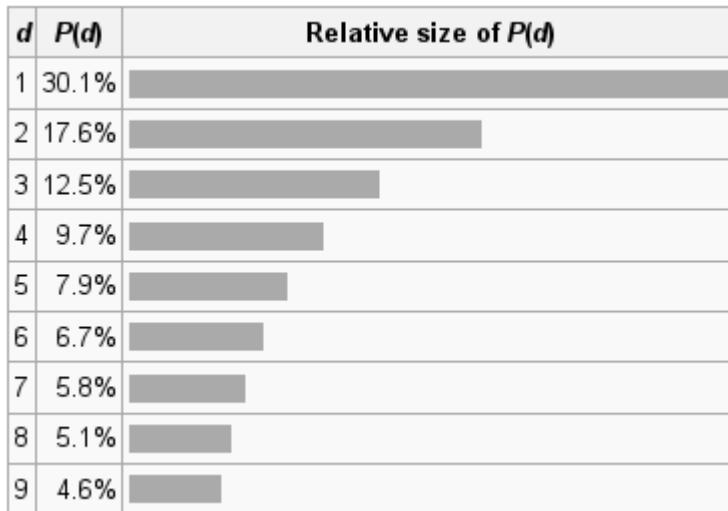
Patrimoine moyen en €	1er chiffre
2,408,869	2
1,925,421	1
2,319,719	2
2,984,232	2
2,325,543	2
2,593,587	2
2,036,967	2
2,392,431	2
2,508,619	2
2,370,514	2
2,393,076	2
3,236,227	3
2,374,940	2
2,201,594	2
2,321,516	2
2,171,647	2



• Loi de Benford

- Loi du 1er chiffre $d \in \{1,2,\dots,9\}$
- Quelle est la répartition de ce 1er chiffre ?

$$P(d) = \log_{10}(d + 1) - \log_{10}(d) = \log_{10} \left(1 + \frac{1}{d} \right).$$



Patrimoine moyen en €	1er chiffre
2,408,869	2
1,925,421	1
2,319,719	2
2,984,232	2
2,325,543	2
2,593,587	2
2,036,967	2
2,392,431	2
2,508,619	2
2,370,514	2
2,393,076	2
3,236,227	3
2,374,940	2
2,201,594	2
2,321,516	2
2,171,647	2



- Intérêt pour la détection de fraude ?
 - Les nombres falsifiés suivent souvent une distribution uniforme
 - Très simple à mettre en oeuvre : comparer la distribution du 1er chiffre avec la loi de Benford
 - Utilisé sur des données de finance, comptabilité, socio-économiques, ...
 - Loi de Benford a un statut légal aux US
 - Mise en évidence de fraude aux élections iraniennes en 2009
 - Accord avec les données du génome, ou les publications scientifiques.
 - Série télévisée Numb3rs (S2 Ep 15)
- Conditions d'application
 - Avoir plusieurs ordres de grandeur (au moins 3)

Détection de fraude – Exemple 1

- 208023 lignes
- nb étudiants/commune ou dépt ou pays



data.gouv.fr

Plateforme ouverte

Comment ça marche ? Producteurs Licence Ouverte Métriques Etalab

Rechercher Où Thématiques +

Effectifs d'étudiants inscrits dans les établissements et les formations de l'enseignement supérieur

Ce jeu de données provient d'un service public certifié
Publié le 14 septembre 2013 par Ministère de l'Enseignement Supérieur et de la Recherche

Ce jeu de données présente les effectifs d'étudiants inscrits dans les établissements et les formations de l'enseignement supérieur, recensés pour les années 2001-2002 à 2011-12 dans les systèmes d'information et enquêtes du ministère de l'Enseignement supérieur et de la Recherche, du ministère de l'Éducation nationale, des ministères en charge de l'Agriculture, de la Pêche, de la Culture, de la Santé et des Sports. Il décline les informations à tous les niveaux géographiques, de la commune jusqu'au national.

Ressources

[CSV](#) Effectifs d'étudiants inscrits dans les établissements et les formations de l'enseignem...
Fichier de données pour les rentrées 2001 à 2011

[CSV](#) Descriptions des modalités des variables
Description des modalités des variables du fichier de données

[RTF](#) Descriptif du jeu de données
Descriptif du jeu de données

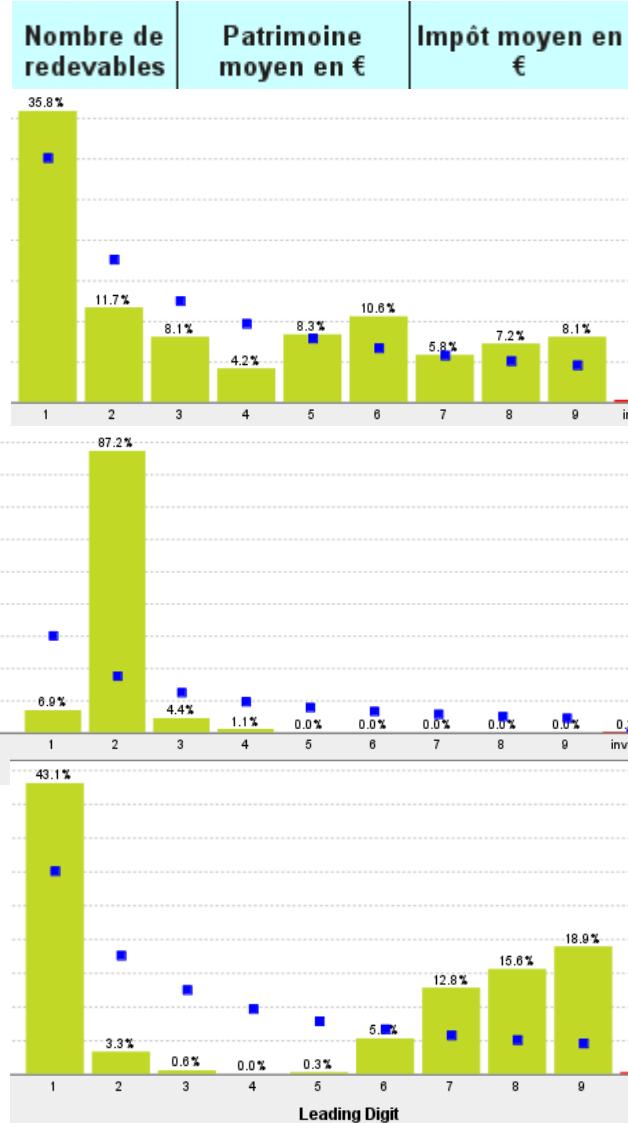
Détection de fraude – Exemple 2

The screenshot shows the data.gouv.fr platform interface. At the top, there's a logo with the French tricolor and the text "Liberté • Égalité • Fraternité" followed by "RÉPUBLIQUE FRANÇAISE". Below it, the URL "data.gouv.fr" is displayed. A banner at the top right says "Plateforme ouverte à tous". The main navigation bar includes links for "Comment ça marche?", "Producteurs", "Licence Ouverte", "Métriques", and "Etablis". Below the navigation is a search bar with a magnifying glass icon, a location search bar with a location pin icon, a dropdown menu for "Thématiques", and a blue "+" button. The main content area has a blue header with the title "Impôt de solidarité sur la fortune". Below the header, it says "Ce jeu de données provient d'un service public certifié" and "Publié le 26 novembre 2013 par Ministère de l'Economie et des Finances". The main text explains that for each commune with over 20,000 inhabitants paying the ISF, the number of liable persons, average wealth, and average contribution can be known. The "Ressources" section provides links to "L'ISF 2011" and "L'ISF 2010" in XLS format.

Attention aux conditions:

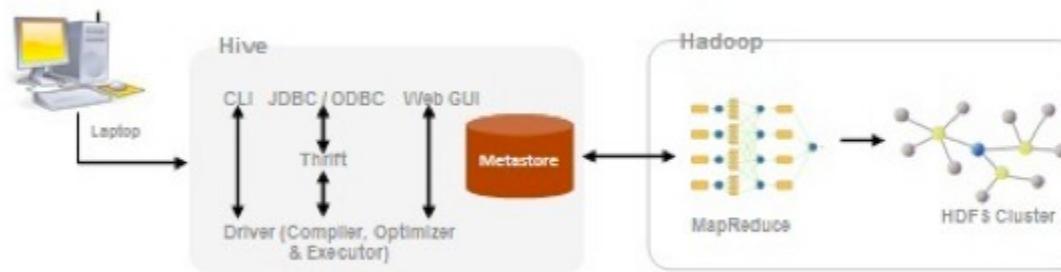
- nombre de lignes = 360
- Ordres de grandeurs pas toujours respectés
- Biais dans les données

Structure du fichier





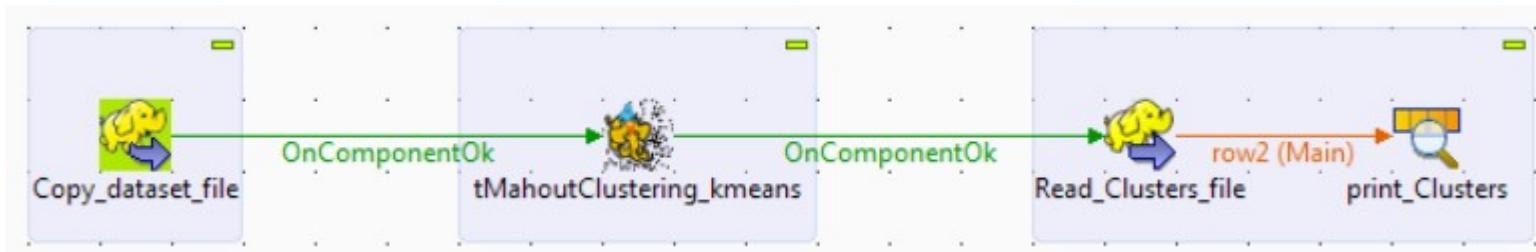
- Utilisation de Hive
- Hive projet Apache initié par Facebook
- Langage de requêtage de type SQL
 - Traduit les requêtes HiveQL en jobs map/reduce Hadoop.



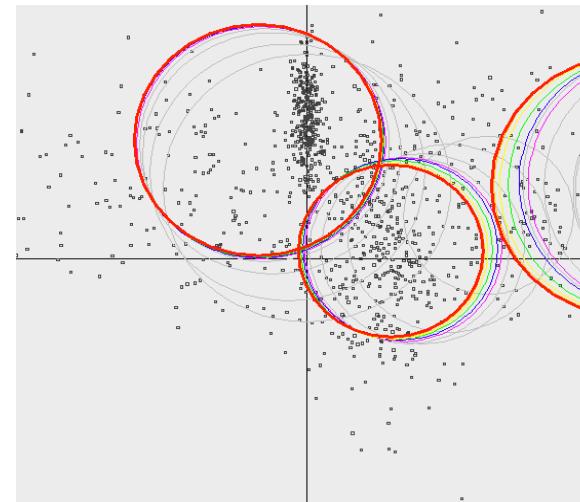
```
SELECT substr(col,1,1), COUNT(*) FROM t  
GROUP BY substr(col,1,1)
```



- Présentation de Talend
- Définition du Big Data
- Le framework Hadoop
- **3 thématiques**
 - Rapprochement des données
 - Détection de fraude
 - **Clustering**
- Les futurs outils de fouille de données sur Hadoop



- Algorithmes disponibles (basés sur Mahout)
 - Canopy (souvent utilisé pour initialiser les clusters du k-means)
 - K-means
 - Fuzzy k-means
 - Dirichlet
- Et plusieurs distances
 - Euclidienne
 - Manhattan
 - Chebyshev
 - Cosinus





- Mahout k-means en ligne de commande

```
bin/mahout kmeans \
    -i <input vectors directory> \
    -c <input clusters directory> \
    -o <output working directory> \
    -k <optional number of initial clusters to sample from input vectors> \
    -dm <DistanceMeasure> \
    -x <maximum number of iterations> \
    -cd <optional convergence delta. Default is 0.5> \
    -ow <overwrite output directory if present>
    -cl <run input vector clustering after computing Canopies>
    -xm <execution method: sequential or mapreduce>
```

- Chaque itération de l'algorithme génère un job map-reduce



- Initialisation des clusters du k-means en exécutant l'algorithme de Canopy
- Parallélisation de l'algorithme de canopy en 3 étapes
 - Chaque Mapper calcule les centroïdes des canopies sur son jeu de données
 - Les Reducers groupent les centroïdes des canopies pour former un ensemble de centroides finaux
 - Chaque point (donnée) est rattaché à son canopy final

Référence (lecture vidéo Google) : <http://is.gd/roeTXK>

Cluster Computing for Web-Scale Data Processing (2008)
Aaron Kimball



- Présentation de Talend
- Définition du Big Data
- Le framework Hadoop
- 3 thématiques
 - Rapprochement des données
 - Détection de fraude
 - Clustering
- **Les futurs outils de fouille de données sur Hadoop**

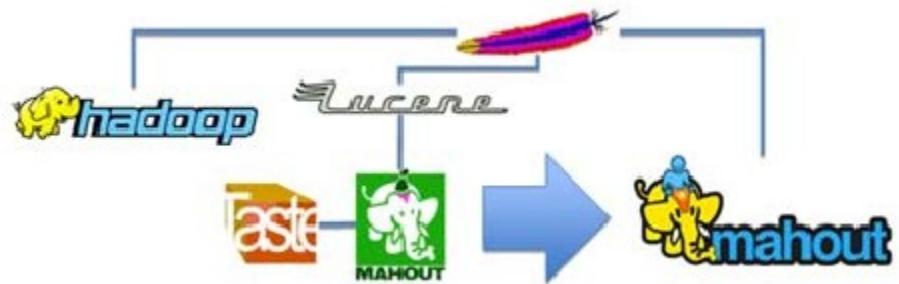


- 
- Librairie d'algorithmes d'apprentissage automatique
- 3 familles d'algorithmes reposant sur Hadoop
 - Clustering
 - Classification
 - Filtrage collaboratif (Recommandation)
- Contient d'autres algorithmes non distribués



- ## Mahout

- fondé par Isabel Drost, Grant Ingersoll, Karl Witten
- En 2008 comme sous projet de Lucene (moteur de recherche) + Taste (Sean Owen)
- Devient un projet Apache à part entière en 2010

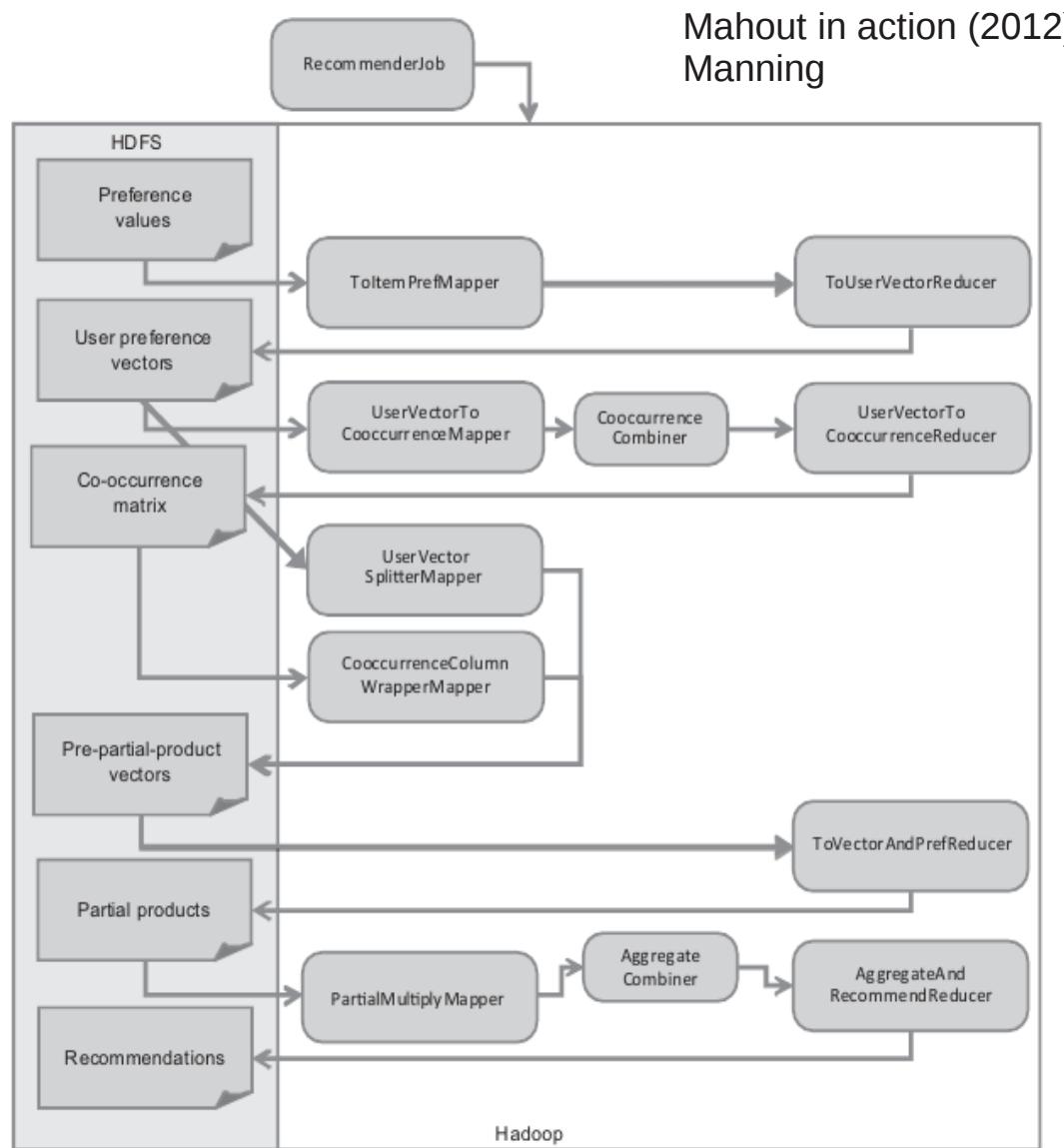




- Classification avec Mahout
 - Classification naïve bayesienne
 - Modèle de Markov caché (HMM)
 - Régression logistique
 - Forêts d'arbres décisionnels (random forests)
- La classification avec Mahout devient intéressante au-delà de 1 à 10 millions de lignes
 - Là où les autres approches ne sont plus scalables.

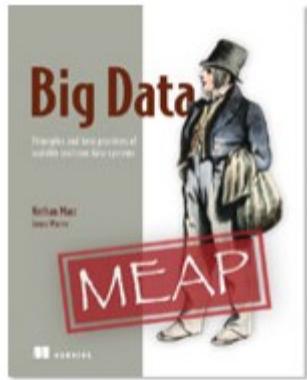


- Recommandation (filtrage collaboratif)
 - filtrage collaboratif utilisateurs
 - le filtrage collaboratif objets
- Plusieurs lectures/écritures disque





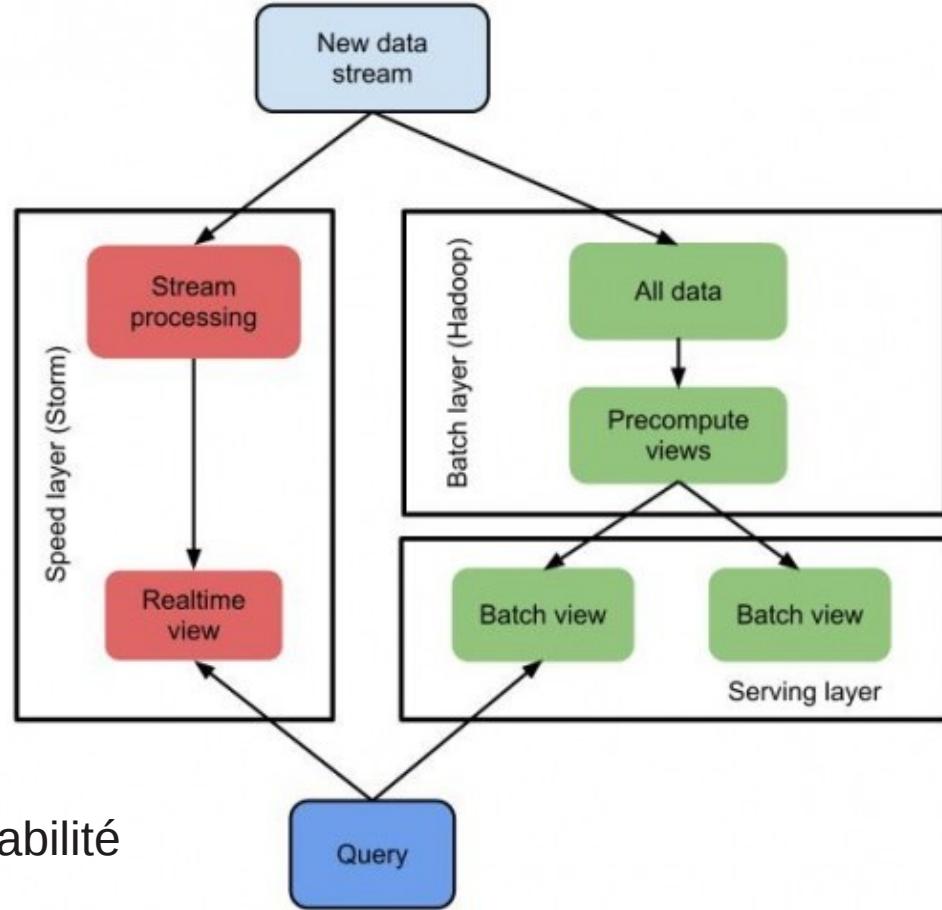
- Mahout a été pensé pour fonctionner avec Map Reduce v1.
 - Traitement batch des données
 - les algorithmes d'apprentissage sont coûteux en IO
 - La conversion des calculs matriciels en programmes MapReduce n'est pas si efficace
 - A la limite, Mahout serait bon pour l'apprentissage, mais la recommandation, les prédictions ou la classification doivent être faite en "temps réel"



Nathan Marz de Twitter définit une architecture générique pour être robuste face aux

- erreurs humaines
- problèmes matériels

Permettant des requêtes ad-hoc, une scalabilité en ajoutant des machines





- Amélioration de la scalabilité
 - Hadoop1: taille max du cluster ~ 5000 noeuds
 - Nombre max de tâches ~ 40 000
- Haute disponibilité
- Meilleure gestion des ressources du cluster
 - Mauvaise répartition des tâches Mappers et Reducer
- Support d'autres modèles de programmation que MapReduce (parcours de graphe, MPI)
 - tout n'est pas adapté à MapReduce
- YARN : système de gestion d'applications distribuées



Single Use System

Batch Apps

HADOOP 1.0

MapReduce
(cluster resource management & data processing)

HDFS
(redundant, reliable storage)

Multi Use Data Platform

Batch, Interactive, Online, Streaming, ...

HADOOP 2.0

MapReduce
(batch)

Tez
(interactive)

Others
(varied)

YARN

(operating system: cluster resource management)

HDFS2

(redundant, reliable storage)

Data Processing Engines Run Natively IN Hadoop

BATCH
MapReduce

INTERACTIVE
Tez

ONLINE
HBase

STREAMING
Storm

GRAPH
Giraph

Microsoft
REEF

Sas
LASR, HPA

OTHERS

YARN

(operating system: cluster resource management)

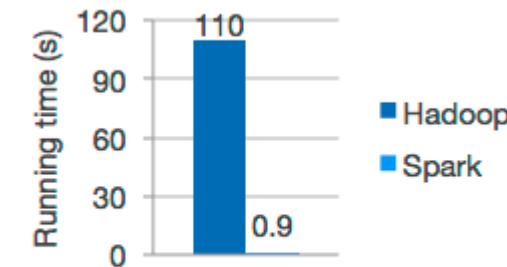
HDFS2

(redundant, reliable storage)



• Spark (projet Apache)

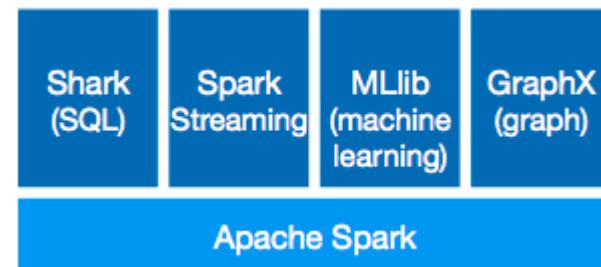
- Tourne sur un cluster Hadoop 2
- 100x plus rapide que Hadoop (en mémoire)



Logistic regression in Hadoop and Spark

• Mllib

- K-means
- Régression linéaire
- Régression logistique
- Classification naïve bayesienne
- Descente de gradient stochastique

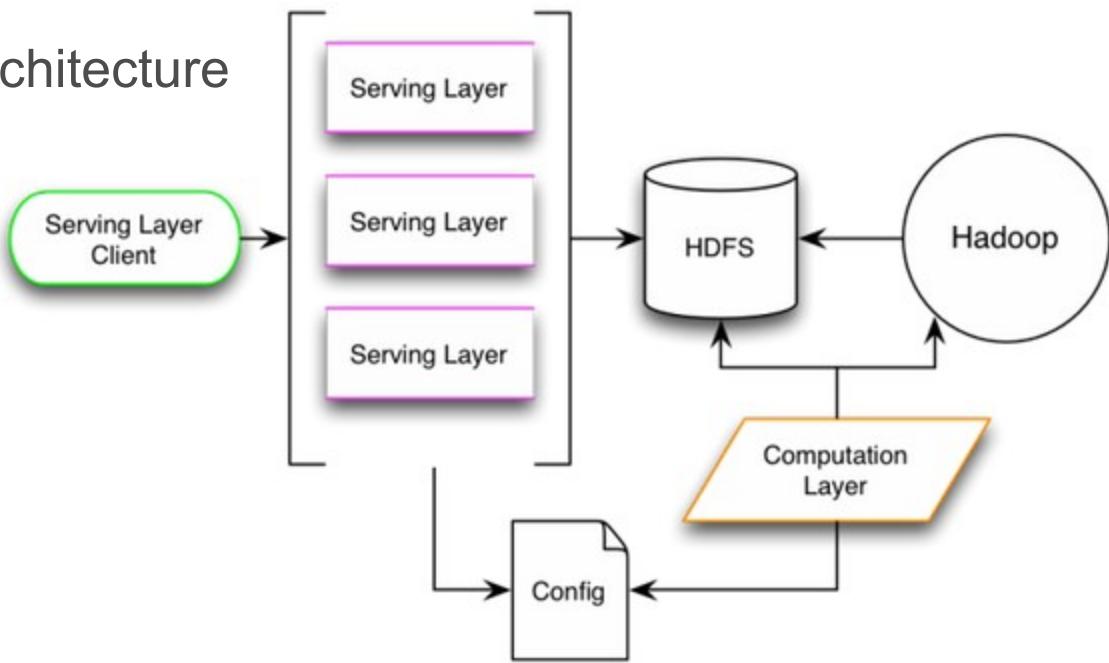




- Mahout va être réécrit pour supporter
 - Spark
 - Ainsi que H2O <http://0xdata.com/h2o-2/>
 - Moteur opensource de machine learning et math.
 - Travaille principalement en mémoire (distributed in-memory Key/Value store)
 - Swap sur HDFS si besoin

- Nouveau projet Oryx de Cloudera

- <http://is.gd/nSk14S>
- S'appuyant sur l'architecture lambda



The Oryx architecture



- ## Futurs travaux

- sur le rapprochement
 - Mesure de la qualité de l'algorithme (F-score)
 - Comparaison avec algorithme MFB du LIPN
 - Version distribuée de l'algorithme MFB
- Sur la détection de fraude
 - Alerter l'utilisateur si déviation du modèle trop élevée
- Sur l'apprentissage
 - Développer des composants pour la classification et la recommandation
 - Supporter les nouveaux frameworks Big Data

Merci !

