

# Big Data

**Enseignant : Mohamed MANAA**

*Année Universitaire 2022-2023*

## Plan

- I. Qu'est-ce que Big Data ?
- II. Cas d'utilisation du Big Data
- III. Apache Hadoop
- IV. HDFS
- V. MapReduce
- VI. YARN
- VII. SPARK
- VIII. Hadoop Query Languages



# Qu'est-ce que Big Data ?

**BigData est conséquence de la maturité de plusieurs disciplines**

**GRID Computing** : Calcul parallèle et distribué, HPC (High Performance Computer), capacité de calcul haute performance

**Cloud Computing** : Capacité de stockage infini, réparti et sécurisé, fragmentation/réPLICATION

**Internet of Things (IoT)** : Ubiquitous Computing (informatique ambiante)

Multitudes de devices connectés (plages IPV6 suffisantes)

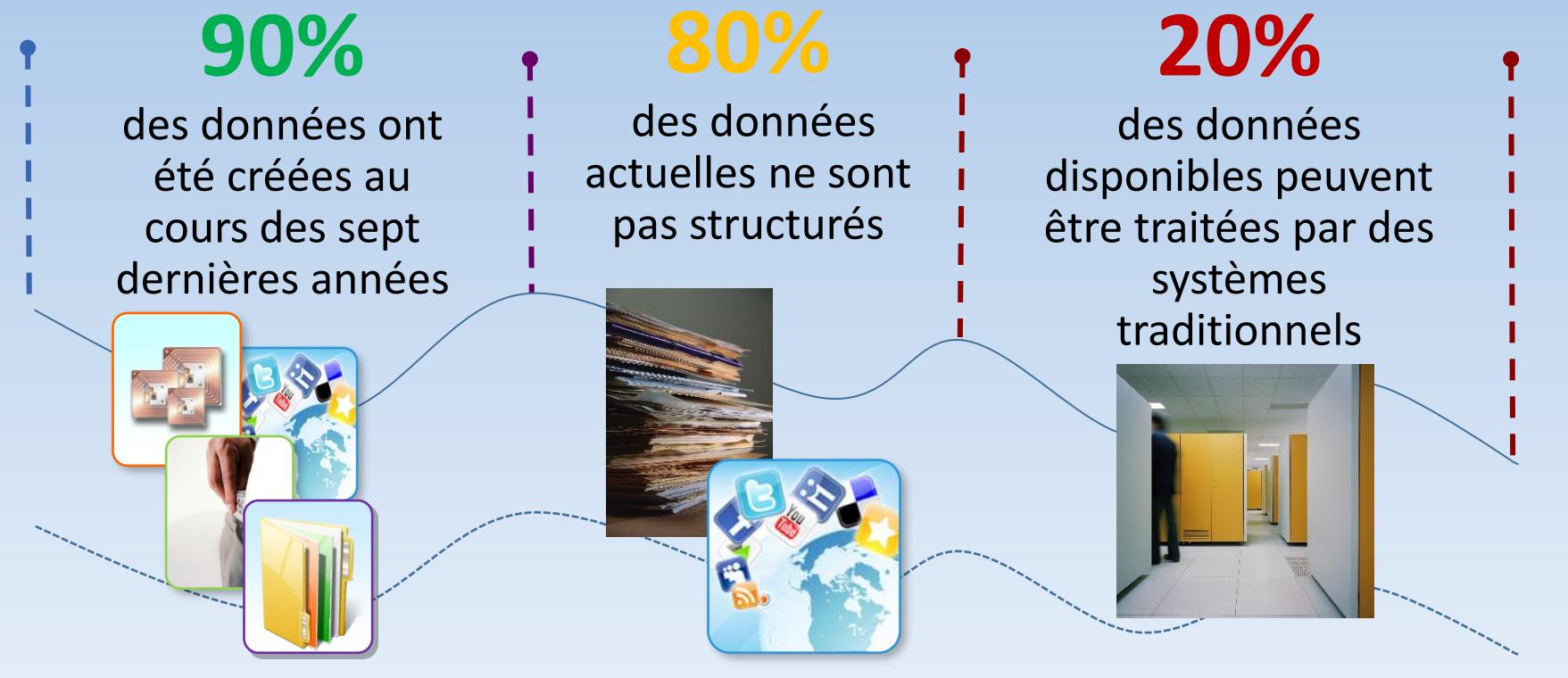
Exemples : les web services façades de tout objet pingable (caméra, capteur, etc.), La voiture comme ordinateur ambulant, la télé-maintenance proactive, la traçabilité (RFID), le tracking par GPS, etc.

**Web 3.0 (Social, Sémantique)** : SNA (Social Network Analysis)

**Data Management** : SQL, noSQL, DWH (datawarehousing), BI (Business Intelligence)

**NLP** (Natural Language Processing)

# Qu'est-ce que Big Data ?



**1 in 2**

des chefs d'entreprise n'ont pas accès aux données dont ils ont besoin

**83%**

des responsables des technologies de l'information et de la communication citent BI et analyses des données comme part de leurs plan prévisionnel

**5.4X**

plus probable que les plus performants utilisent business analytics

# Qu'est-ce que Big Data ?

Un monde  
interconnecté  
et instrumenté



# Qu'est-ce que Big Data ?

## Quelques Statistiques : 1 minute / 2021

- ✓ 168 millions d'emails envoyés
- ✓ 510 000 commentaires sur Facebook
- ✓ 98 000 tweets sur Twitter
- ✓ 25 000 nouveaux messages sur Instagram
- ✓ 694 445 requêtes de recherche sur Google
- ✓ 25 heures de vidéos postées sur YouTube
- ✓ 20,8 millions de messages sur WhatsApp

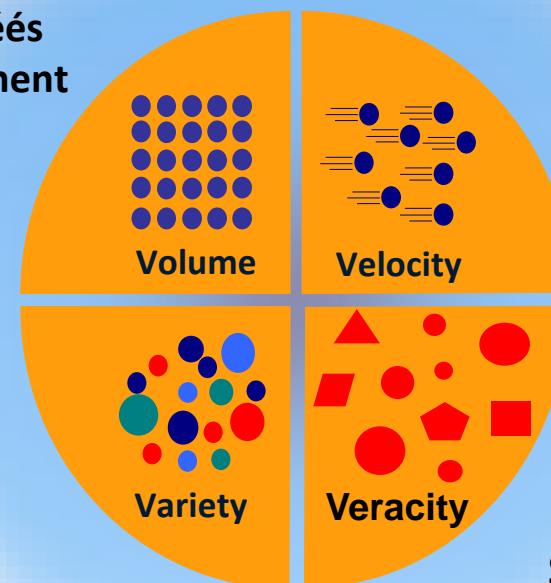


# Qu'est-ce que Big Data ?

Avec le Big Data, nous sommes entrés dans une nouvelle ère d'analyse

**12+** terabytes

de Tweets créés  
quotidiennement



**100**

différents types  
de données

**5+** million

transactions commerciales  
par seconde

Seulement **1 sur 3**

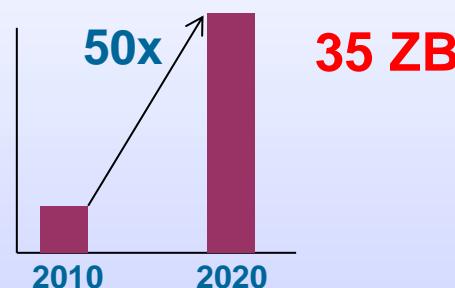
décideurs fait confiance à ses  
informations

# Qu'est-ce que Big Data ?

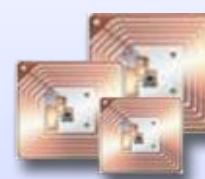
## Caractéristiques du Big Data

$V^4 = \text{Volume Velocity Variety Veracity}$

Coûts de traitement efficace du **Volume** croissant



Répondre à la **Vitesse** croissante **Velocity**



**30 Milliards**  
Capteurs RFID et comptage

Analyser collectivement  
l'élargissement de la  
**Variété Variety**



**80%** des données mondiales ne sont pas structurés



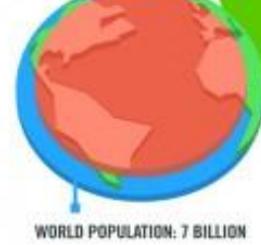
Établissement de la **Véracité** des grandes sources de données  
**Veracity**

**1 à 3** des chefs d'entreprise ne font pas confiance à l'information qu'ils utilisent pour prendre des décisions

**40 ZETTABYTES**

[ 43 TRILLION GIGABYTES ]  
of data will be created by  
2020, an increase of 300  
times from 2005

**6 BILLION PEOPLE**  
have cell phones



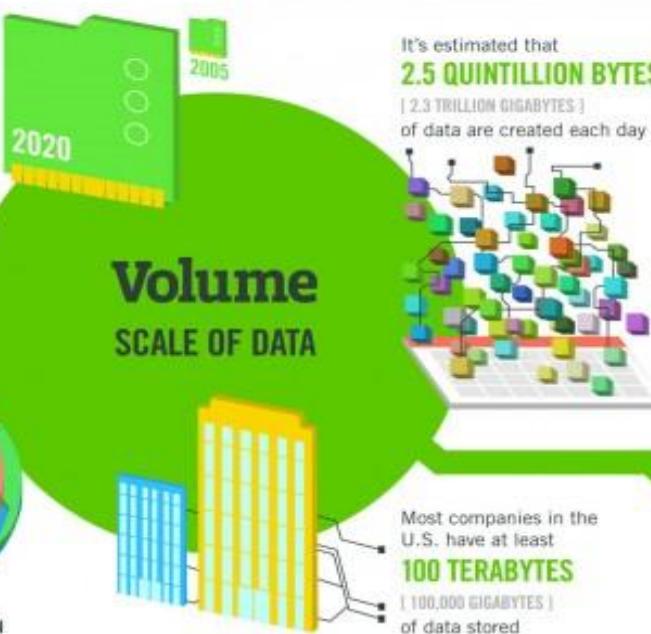
The New York Stock Exchange captures  
captures

**1 TB OF TRADE INFORMATION**

during each trading session



## Volume SCALE OF DATA

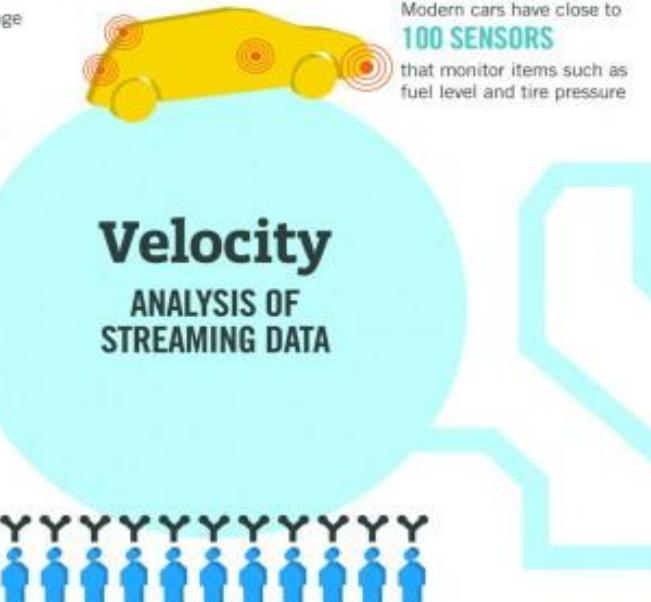


## Velocity ANALYSIS OF STREAMING DATA

By 2016, it is projected there will be

**18.9 BILLION NETWORK CONNECTIONS**

– almost 2.5 connections per person on earth



# The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume**, **Velocity**, **Variety** and **Veracity**.

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015 **4.4 MILLION IT JOBS** will be created globally to support big data, with 1.9 million in the United States.



As of 2011, the global size of data in healthcare was estimated to be

**150 EXABYTES**  
[ 161 BILLION GIGABYTES ]



**30 BILLION PIECES OF CONTENT**

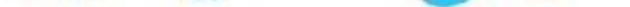
are shared on Facebook every month



## Variety DIFFERENT FORMS OF DATA

**400 MILLION TWEETS**

are sent per day by about 200 million monthly active users



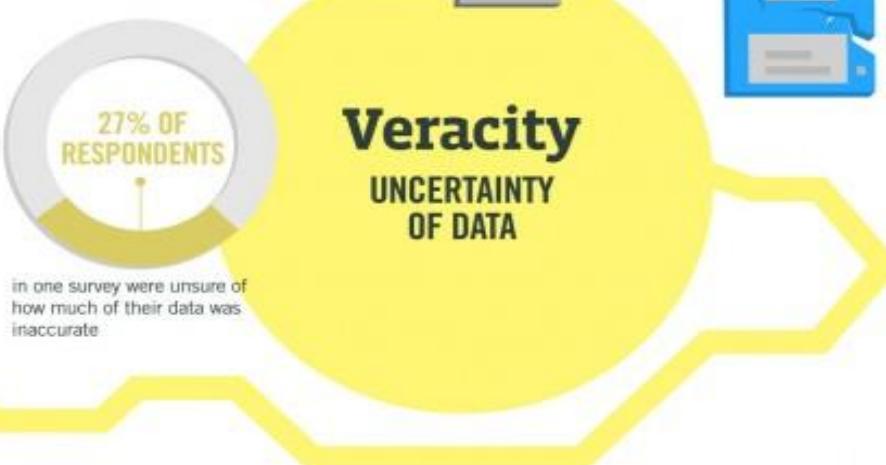
**1 IN 3 BUSINESS LEADERS**

don't trust the information they use to make decisions



**27% OF RESPONDENTS**

in one survey were unsure of how much of their data was inaccurate



IBM

# Cas d'utilisation du Big Data

# Cas d'utilisation du Big Data

## Les 5 principaux cas d'utilisation de données clés



### Exploration du Big Data

Trouver, visualiser, comprendre toutes les grandes données pour améliorer la prise de décision



### Vue 360 ° améliorée du client

Étendre les vues des clients existantes en intégrant des sources de données internes et externes supplémentaires



### Extension de sécurité / intelligence

Risque plus faible, détection de la fraude et suivi de la cybersécurité en temps réel



### Analyse d'opérations

Analyser une variété de données machine pour améliorer les résultats commerciaux



### Augmentation du stockage de données

Intégrer big data et data warehouse pour accroître l'efficacité opérationnelle

## **Cas d'utilisation du Big Data**

### **Exploration du Big Data**



Le centre d'appels du premier fournisseur d'assurance maladie permet à 14 000 agents d'avoir une vue unique des données client et produit

#### Besoin

- L'accès inefficace à d'énormes volumes de données cloisonnées sur les clients et les produits a réduit la productivité des agents et augmenté le temps moyen de traitement des appels. Les agents avaient besoin d'un accès plus rapide aux informations

#### Avantages

- ✓ Amélioration de la productivité de 14 000 agents, économisant en moyenne 3 secondes sur le temps de traitement des appels et des millions de dollars par an
- ✓ A aidé à garantir une disponibilité de 99,999 % à chaque emplacement, offrant une vitesse de requête par seconde impressionnante
- ✓ Amélioration des performances des applications pour prendre en charge les opérations quotidiennes et les utilisateurs professionnels sur 180 sites



Le fabricant mondial de l'aérospatiale permet à son personnel d'accéder aux informations critiques

## Besoin

- Améliorez l'efficacité opérationnelle en fournissant une capacité unifiée de recherche, de découverte et de navigation pour fournir un accès rapide aux informations pertinentes dans toute l'entreprise

## Avantages

- ✓ Mise en service de 50 avions supplémentaires dans le monde au cours de la première année sans augmentation de personnel
- ✓ Économie de 36 millions de dollars par an grâce au soutien du programme d'avions en vol 24h/24 et 7j/7
- ✓ Visibilité de la chaîne d'approvisionnement pour réduire le temps de cycle, économisant des millions de dollars sur les livraisons de pièces critiques

Les techniciens font la course contre la montre pour faire décoller les avions et les remettre en état de marche.

## Cas d'utilisation du Big Data

**Vue 360 ° améliorée du client**

# Cas d'utilisation du Big Data

## Vue améliorée à 360° du client : besoins



Étendre les vues clients existantes (MDM, CRM, etc.) en incorporant des sources d'informations internes et externes supplémentaires



Besoin d'une compréhension plus approfondie du sentiment des clients provenant de sources internes et externes

Désir d'augmenter la fidélité et la satisfaction des clients en comprenant quelles actions significatives sont nécessaires

Difficile de fournir les bonnes informations aux bonnes personnes pour fournir aux clients ce dont ils ont besoin pour résoudre les problèmes

# Cas d'utilisation du Big Data



Vue unifiée des informations du client

## **Cas d'utilisation du Big Data**

### **Security/Intelligence Extension**

# Cas d'utilisation du Big Data

## Extension de sécurité / intelligence



Security/Intelligence Extension améliore les solutions de sécurité traditionnelles en analysant tous les types et toutes les sources de données



Intelligence améliorée et aperçu de la surveillance

**Analysez les données en mouvement et au repos pour :**

- Trouver des associations
- Découvrez des modèles et des faits
- Maintenir l'actualité des informations



Prédiction et atténuation des cyberattaques en temps réel

**Analysez le trafic réseau pour :**

- Découvrez rapidement les nouvelles menaces
- Déetecter les menaces complexes connues
- Agissez en temps réel



Prédiction du crime et protection

Réduire le taux de désabonnement des clients

**Analyser les données Telco et sociales pour**

- Recueillir des preuves criminelles
- Prévenir les activités criminelles
- Appréhender les criminels de manière proactive
- Fidélisation de la clientèle



## Asian Telco réduit les coûts de facturation et améliore la satisfaction client.

Capacités :

Calcul de flux

Accélérateurs analytiques

Médiation et analyse en temps réel de  
**6 milliards de CDR par jour**

Temps de traitement des données réduit de  
**12h à 1sec**

**Coût du matériel réduit à 1/8ème**

Résoudre de manière proactive les problèmes (par exemple, les appels interrompus) ayant une incidence sur la satisfaction des clients.



Asian Government  
Agency

## National Intelligence Platform

Capacités :

Calcul de flux

Analyser tout le trafic Internet (réseaux sociaux, e-mail, etc.)

Suivre les personnes d'intérêt (trafiquants de drogue/sexuels, terroristes, réfugiés/immigrants illégaux) et les activités civiles/frontalières



## **Cas d'utilisation du Big Data**

### **Analyse des opérations**

# Cas d'utilisation du Big Data

## Analyse des opérations : besoins



**Analyser une variété de données machine pour améliorer les résultats commerciaux**

### Défis commerciaux :

- Complexité et croissance rapide des données machine
- Difficile de capturer une petite fraction de la machine pour une meilleure décision
- Incapacité à analyser les données de la machine et à les combiner avec les données de l'entreprise pour une analyse complète



1	11.1.2.4	Oct 21 06:33:45	hex('1043B82D007BA4D16')	syslog	%L2-BDF	Cisco - Nur
2	11.1.2.4	Oct 21 06:33:45	hex('1043B82D007BA4D16')	syslog	%L2-BDF	AnyVendor
3	11.1.2.4	Oct 21 06:33:45	hex('1043B82D007BA4D16')	syslog	%L2-BDF	PNOC - Intb
4	11.1.2.4	Oct 21 06:33:45	hex('1043B82D007BA4D16')	syslog	%L2-BDF	PNOC - Intb
5	11.1.2.4	Oct 21 06:33:45	hex('1043B82D007BA4D16')	syslog	%L2-BDF	AnyVendor
6	11.1.2.4	Oct 21 06:33:45	hex('1043B82D007BA4D16')	syslog	%L2-BDF	Cisco - Nur

### Benefits:

- Bénéficiez d'une visibilité en temps réel sur les opérations, l'expérience client, les transactions et le comportement
- Planifier de manière proactive pour augmenter l'efficacité opérationnelle

- Identifier et enquêter sur les anomalies
- Surveiller l'infrastructure de bout en bout pour éviter de manière proactive la dégradation ou les pannes de service

### Capacités :

Hadoop et calcul de flux

### Gestion intelligente de l'infrastructure :

analyse des journaux, prévision des factures d'énergie, optimisation de la consommation d'énergie, détection de l'utilisation anormale d'énergie, gestion de l'énergie en fonction de la présence

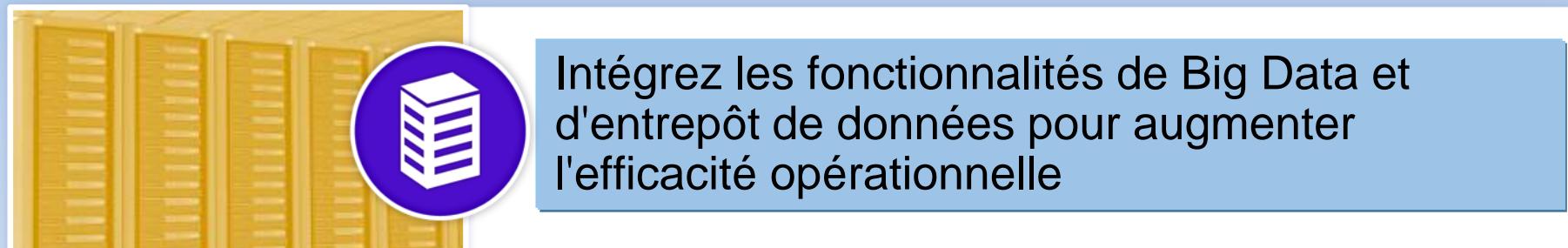
**Optimisation de la consommation énergétique** du bâtiment avec une surveillance centralisée ; Maintenance préventive et corrective automatisée

## **Cas d'utilisation du Big Data**

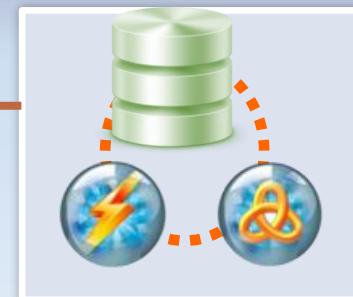
**Augmentation de l'entrepôt de données**

# Cas d'utilisation du Big Data

## Augmentation de l'entrepôt de données : besoins



Intégrez les fonctionnalités de Big Data et d'entrepôt de données pour augmenter l'efficacité opérationnelle



### Besoin d'exploiter une variété de données

- Sources de données structurées, non structurées et en continu requises pour une analyse approfondie
- Exigences de faible latence (heures, pas semaines ou mois)
- Accès de requête requis aux données

### Étendre l'infrastructure de l'entrepôt

- Optimisation des coûts de stockage, de maintenance et de licence en migrant les données rarement utilisées vers Hadoop
- Coûts de stockage réduits grâce au traitement intelligent des données en continu
- Amélioration des performances de l'entrepôt en déterminant les données à y intégrer

# Apache Hadoop

# Apache Hadoop

- Pourquoi? Quand? Où?
- Hadoop Basics
  - ✓ Comparaison avec le SGBDR
- Hadoop architecture
  - ✓ MapReduce
  - ✓ HDFS
  - ✓ Hadoop Common
  - ✓ Ecosystème de projets connexes
  - ✓ Pig, Hive, Jaql
  - ✓ Autres projets
- Distributions Hadoop

# Apache Hadoop

## Améliorations matérielles au cours des années ...

- Vitesses du CPU:
  - ✓ 1990 - 44 MIPS à 40 MHz
  - ✓ 2010 - 147 600 MIPS à 3,3 GHz
- La mémoire RAM
  - ✓ 1990 - Mémoire conventionnelle 640K (mémoire étendue 256K recommandée)
  - ✓ 2010 - 8-32GB (et plus)
- Capacité du disque
  - ✓ 1990 - 20 Mo
  - ✓ 2010 - 1To
- Latence du disque (vitesse de lecture et d'écriture) - pas beaucoup d'amélioration au cours des 7 à 10 dernières années, actuellement d'environ 70 à 80 Mo / s

# Apache Hadoop

## Améliorations matérielles au cours des années ...

**Combien de temps faudra-t-il pour lire 1 To de données?**

**1TB (à 80Mb / sec):**

1 disque - 3,4 heures

10 disques - 20 min

100 disques - 2 min

1000 disques - 12 sec

# Apache Hadoop

**Le traitement de données en parallèle est la réponse!**

- Défis

- ✓ Hétérogénéité
- ✓ Ouverture
- ✓ Sécurité
- ✓ Évolutivité
- ✓ Concurrence
- ✓ Tolérance aux pannes
- ✓ Transparence



# Apache Hadoop

## Qu'est-ce que Hadoop?

- ↳ Structure logicielle open source Apache pour un calcul réparti, évolutif et distribué d'une quantité massive de données
  - ⇒ Masque les détails et les complexités du système pour l'utilisateur
  - ⇒ Développé en Java
- ↳ Composé de 3 sous-projets:
  - ⇒ MapReduce
  - ⇒ Hadoop Distributed File System «HDFS» : Système de fichiers distribués Hadoop
  - ⇒ Hadoop Common
- ↳ Soutenu par plusieurs projets liés à Hadoop
  - ⇒ Hbase, Zookeeper, Avro, Etc.
- ↳ Destiné à un matériel de produits hétérogènes



# Apache Hadoop

## Les principes de conception de Hadoop

- ✓ Nouvelle façon de stocker et de traiter les données:
  - Laissez le système gérer la plupart des problèmes automatiquement:
    - Les échecs
    - Évolutivité
    - Réduire les communications
    - Distribuer les données et le pouvoir de traitement à l'endroit où les données sont
    - Faire en sorte que le parallélisme fasse partie du système d'exploitation
    - Matériel relativement peu coûteux
- ✓ Apportez le traitement aux données!

# Apache Hadoop

## Les principes de conception de Hadoop

- ✓ Hadoop = HDFS + MapReduce infrastructure
  
- ✓ Optimisé pour gérer
  - Des quantités massives de données par un parallélisme
  - Une variété de données (structuré, non structuré, semi-structuré)
  - Utilisation d'un matériel de commodité peu coûteux
  
- ✓ Fiabilité fournie par réPLICATION

# Apache Hadoop

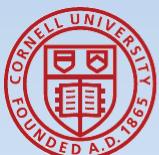
## Hadoop n'est pas pour tous les types de travail

- ⇒ Ne traite pas les transactions - Transaction processing (accès aléatoire)
- ⇒ Pas bon lorsque le travail ne peut pas être parallélisé
- ⇒ Pas bon pour l'accès aux données à faible latence
- ⇒ Pas bon pour traiter beaucoup de petits fichiers
- ⇒ Pas bon pour les calculs intensifs avec peu de données

# Apache Hadoop

Qui utilise Hadoop?

Aol.



Cornell University

YAHOO!

hulu



Google



SecureWorks



NOKIA



# Apache Hadoop

## Un peu d'histoire

Google  
développe  
Google File  
system et  
MapReduce

Google  
publie un  
article sur  
MapReduce

Cutting/Yahoo!  
renommer leur  
effort Hadoop

Hadoop  
rendu Open  
Source par  
Yahoo!

Doug Cutting  
nommé président  
du conseil  
d'administration  
d'ASF

2001

2003

2006

2008

2011

Doug Cutting (Yahoo! Employé à  
l'époque) développeur de Nutch (Web  
crawler) commence à utiliser  
MapReduce pour l'indexation.

IBM  
présente  
BigInsights  
basé sur  
Hadoop

# Apache Hadoop

## Qu'est-ce que Hadoop?

- Prise en charge flexible de gros volumes de données
  - ✓ Inspiré par les technologies Google (MapReduce, GFS, BigTable, ...)
  - ✓ Initié à Yahoo
    - À l'origine conçu pour résoudre les problèmes d'évolutivité de Nutch, une technologie de recherche Web open source
  - ✓ Bien adapté aux applications à lecture intensive
  - ✓ Prend en charge une grande variété de données
- Permet aux applications de travailler avec des milliers de nœuds et petabytes de données de manière parallèle et rentable

# Apache Hadoop

## Les composants de Hadoop

### Les composants de base

- ✓ Système de fichiers distribué Hadoop (HDFS)
- ✓ MapReduce
- ✓ Hadoop Commun

### Les versions :

- V1 : Classique
- V2 : Introduction de YARN
- V3 : Courante

# Apache Hadoop

## Composants de base de Hadoop

### ↳ Framework MapReduce

- ✓ Comment Hadoop comprend et affecte le travail aux nœuds (machines)



### ↳ Hadoop Distributed File System = HDFS

- ✓ Système de fichiers distribués Hadoop
- ✓ Où Hadoop stocke les données
- ✓ Un système de fichiers qui couvre tous les nœuds dans un cluster Hadoop
- ✓ Il regroupe les systèmes de fichiers sur de nombreux nœuds locaux pour les transformer en un grand système de fichiers



# Apache Hadoop

## Composants de base de Hadoop

### ↳ Hadoop Common



- Anciennement connu sous le nom Hadoop Core
- Contient des utilitaires et des bibliothèques qui soutiennent les autres sous projets de Hadoop
  - ✓ Système de fichiers
  - ✓ Appel de procédure à distance (RPC)
  - ✓ Sérialisation
- Par exemple : shell du système de fichiers
  - ✓ Pour interagir directement avec les fichiers HDFS, vous devez utiliser

**/bin/hdfs dfs <args>**

# Apache Hadoop

## Hadoop - Exigences d'installation

Types d'installation :

- Single-node:
  - opérations simples
  - test local et débogage
- Multi-node cluster:
  - opération de niveau de production
  - des milliers de nœuds

# Apache Hadoop

## Hadoop - Exigences d'installation

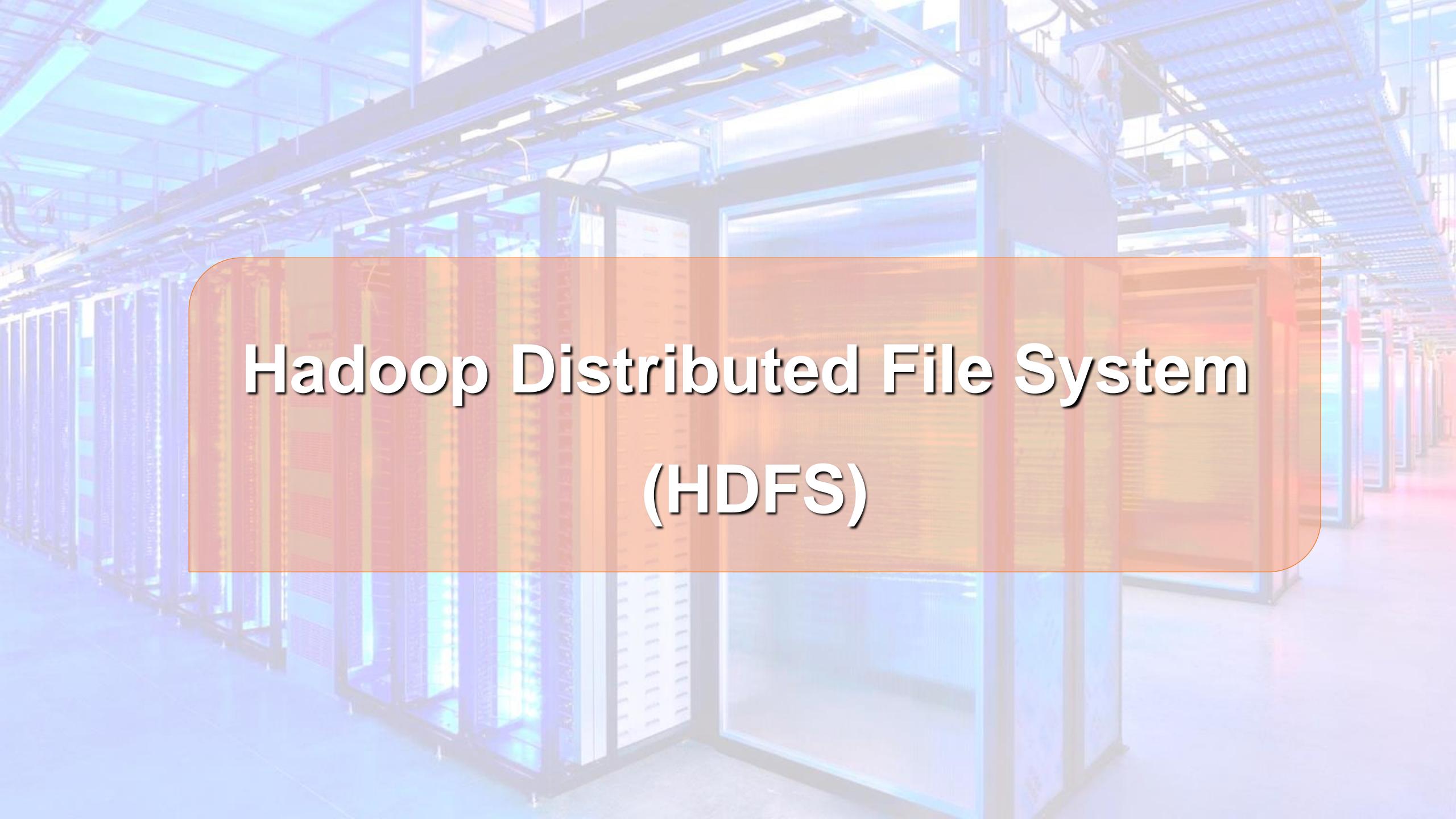
Matériel:

- ✓ RAM: MapReduce jobs sont principalement liés aux E/S ⇒ planifiez suffisamment de RAM
- ✓ CPU: les processeurs haut de gamme ne sont souvent pas rentables
- ✓ Disks: utilisez des disques de haute capacité
- ✓ Network: dépend de la charge de travail, choisissez l'équipement de réseau haut de gamme pour les grands clusters

▪ Software:

- ✓ OS: GNU / Linux pour le développement et la production / Windows / Mac pour le développement
- ✓ Java
- ✓ ssh

# Hadoop Distributed File System (HDFS)



# Hadoop Distributed File System (HDFS)

## Hadoop Distributed File System (HDFS)

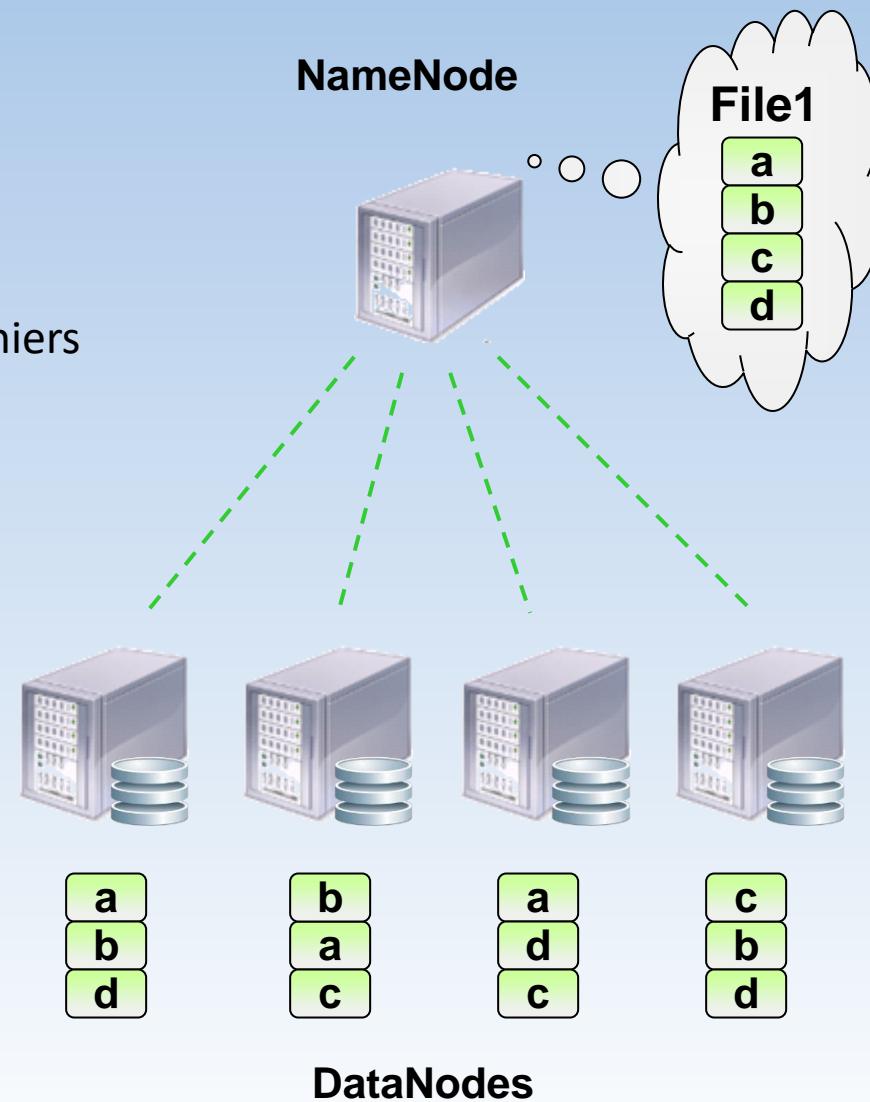
- Distribué, évolutif, tolérant aux pannes, débit élevé
- Accès aux données via MapReduce
- Fichiers divisés **en blocs**
- **3 répliques** pour chaque donnée par défaut
- Peut **créer, supprimer, copier**, mais PAS mettre à jour
- Conçu pour la lecture en continu, pas un accès aléatoire
- Localité des données: traitement des données sur ou près du stockage physique pour réduire la transmission des données



# Hadoop Distributed File System (HDFS)

## Architecture de HDFS :

- Architecture maître / esclave (Master / Slave architecture)
- Master: **NameNode**
  - ✓ gère l'espace de noms et les métadonnées du système de fichiers
    - FslImage
    - EditLog
  - ✓ régule l'accès client aux fichiers
- Slave: **DataNode**
  - ✓ beaucoup par Cluster
  - ✓ gère le stockage attaché aux nœuds
  - ✓ rapporte périodiquement l'état à NameNode



# Hadoop Distributed File System (HDFS)

## HDFS – Blocks

- HDFS est conçu pour prendre en charge les fichiers très volumineux
- Chaque fichier est divisé en blocs (Par défaut : 64 Mo)
- Les blocs résident sur différents DataNode physiques



- Si un fichier ou un morceau du fichier est inférieur à la taille du bloc, seul l'espace requis est utilisé. Exemple: un fichier de 210 Mo est divisé en :



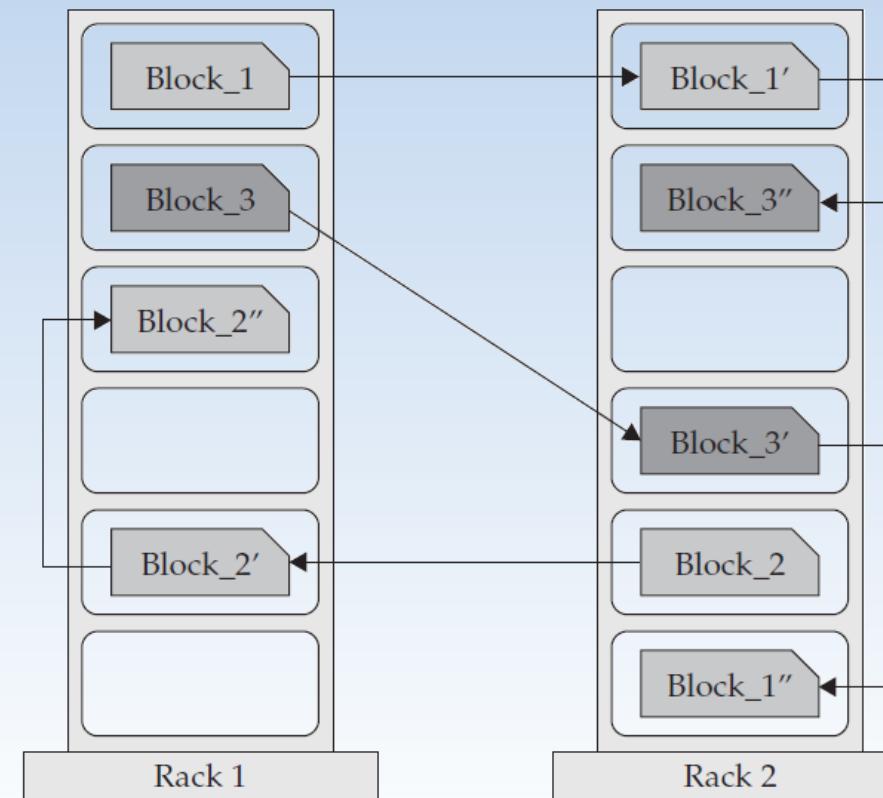
# Hadoop Distributed File System (HDFS)

## HDFS – Replication

- ✓ Les blocs de données sont reproduits sur plusieurs nœuds
- ✓ Le comportement est contrôlé par le facteur de réPLICATION (replication factor), configurable par fichier
- ✓ La valeur par défaut est de **3 répliques**

- une réplique sur un nœud dans le rack local
- une autre réplique sur un nœud dans un rack différent
- et le dernier sur un nœud différent dans le deuxième rack

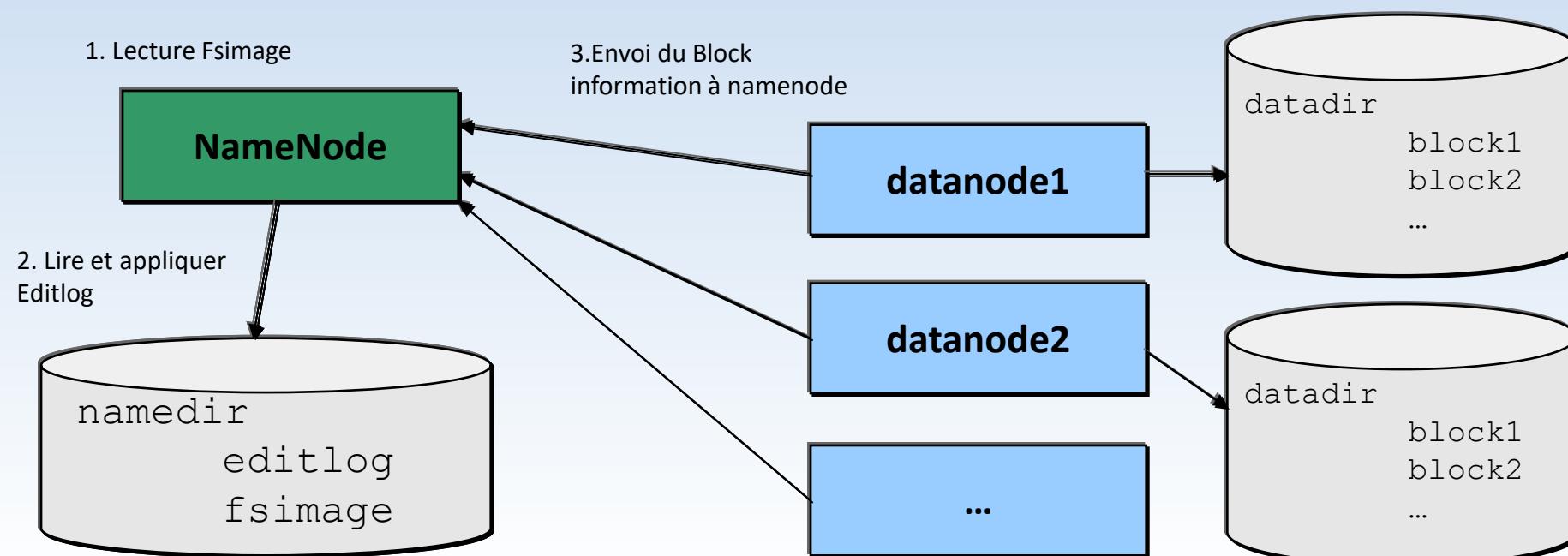
*Cela réduit la bande passante du réseau inter-rack, ce qui améliore les performances d'écriture*



# Hadoop Distributed File System (HDFS)

## HDFS – Namenode

- ✓ NameNode lit fsimage dans la mémoire
- ✓ NameNode applique les changements dans le journal d'édition (EditLog)
- ✓ NameNode attend des données de blocs à partir de nœuds de données
  - Namenode ne stocke pas les informations de bloc
  - Namenode sort en mode Safemode lorsque 99,9% des blocs ont au moins une copie représentative



# Hadoop Distributed File System (HDFS)

## File System Shell

fs – file system shell

- **File System Shell (fs)**

- Appelé comme suit :

```
hadoop fs <args>
```

- **Exemple:**

- Liste le contenu du répertoire actuel dans hdfs

```
hadoop fs –ls .
```

# Hadoop Distributed File System (HDFS)

## File System Shell

- De nombreuses commandes POSIX
  - cat, chgrp, chmod, chown, cp, du, ls, mkdir, mv, rm, stat, tail
- Quelques commandes spécifiques à HDFS
  - copyFromLocal, put, copyToLocal, get, getmerge, setrep

# Hadoop Distributed File System (HDFS)

## Exemple de problème :

- **copyFromLocal / put**

- Copie les fichiers du système de fichiers local dans fs

```
hadoop fs -copyFromLocal <localsrc> .. <dst>
```

Ou

```
hadoop fs -put <localsrc> .. <dst>
```

- **copyToLocal / get**

- Copier des fichiers de fs dans le système de fichiers local

```
hadoop fs -copyToLocal <src> <localdst>
```

Ou

```
hadoop fs -get <src> <localdst>
```

# MapReduce

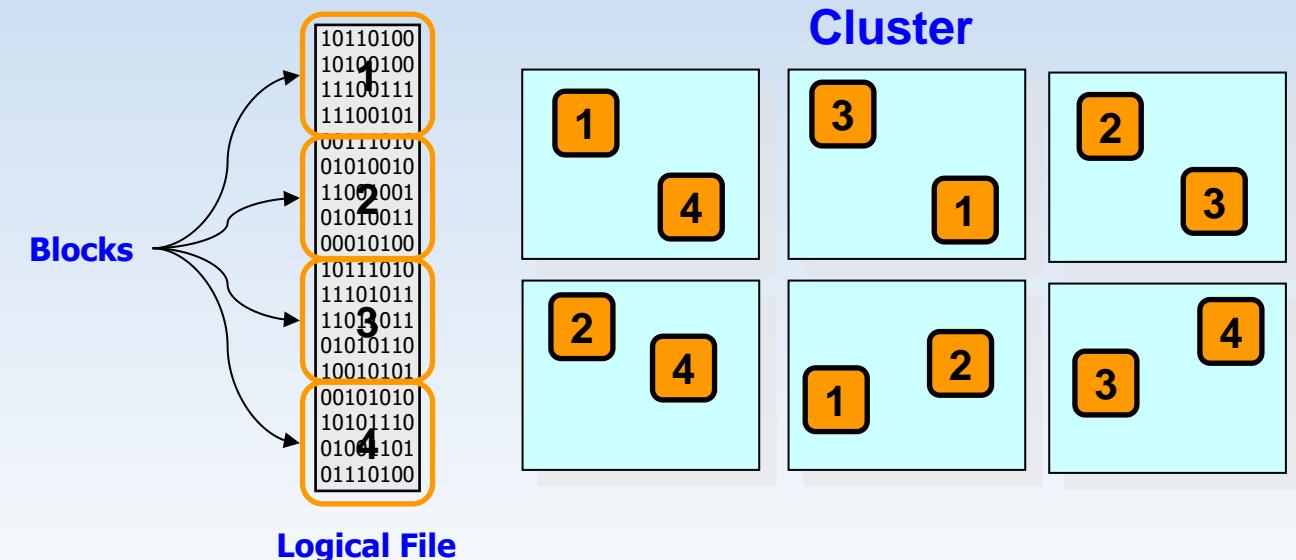
# MapReduce

- Introduction à MapReduce
- Tâches MapReduce (Tasks)
- Exemple WordCount
- Splits
- Exécution
- Planification

# MapReduce

## Introduction à MapReduce

- Principes :
  - ✓ Les données sont stockées sur l'ensemble du cluster
  - ✓ Les programmes sont transmis aux données et non les données aux programmes
- Les données sont stockées sur l'ensemble du cluster (le DFS)
  - ✓ L'ensemble du cluster participe au système de fichiers
  - ✓ Les blocs d'un seul fichier sont répartis sur le cluster
  - ✓ Un bloc donné est généralement répliqué pour la résilience



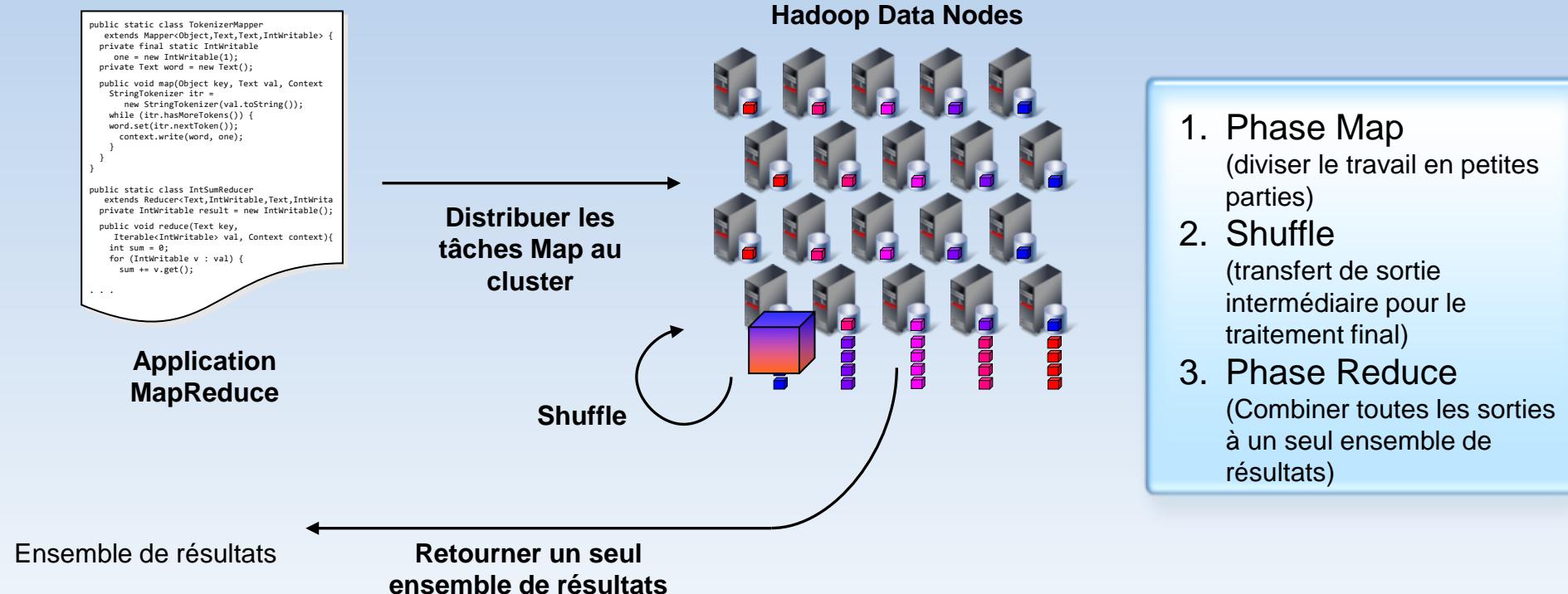
# MapReduce

## Introduction à MapReduce

- Modèle de calcul Hadoop
  - ✓ Données stockées dans un système de fichiers distribué couvrant de nombreux ordinateurs peu coûteux
  - ✓ Apporter la fonction aux données
  - ✓ Distribuer l'application aux ressources de calcul où les données sont stockées
- Adaptable à des milliers de nœuds et petabytes de données

# MapReduce

## Data Science :



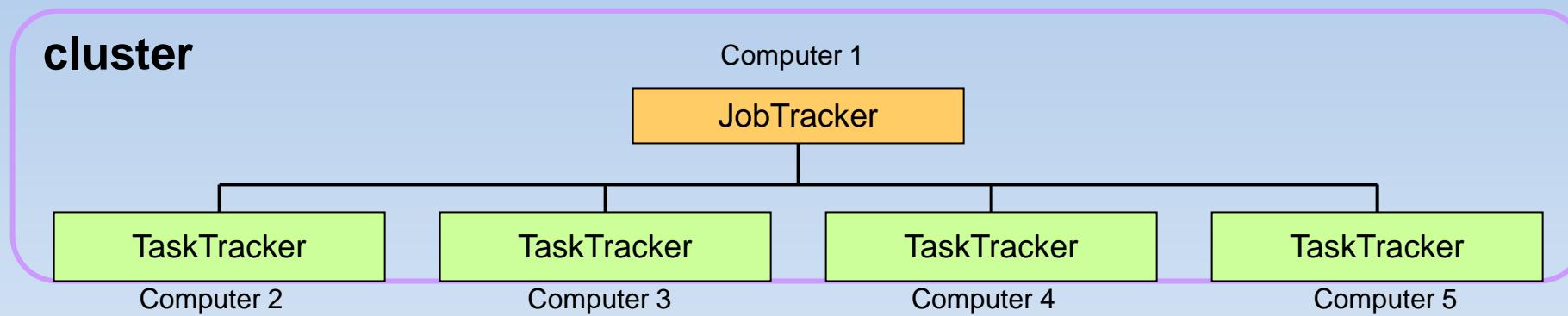
# MapReduce

## Architecture de MapReduce

- Architecture maître / esclave
  - ✓ Le maître (JobTracker) unique contrôle l'exécution du travail sur plusieurs esclaves (TaskTrackers).
- **JobTracker**
  - ✓ Accepte les emplois MapReduce soumis par les clients
  - ✓ Pousse les tâches MAP et REDUCE dans les nœuds TaskTracker
  - ✓ Maintient le travail physiquement proche des données
  - ✓ Surveille les tâches et le statut de TaskTracker
- **TaskTracker**
  - ✓ Exécute les tâches MAP et REDUCE
  - ✓ Envoi son statut à JobTracker
  - ✓ Gère le stockage et la transmission de la sortie intermédiaire

# MapReduce

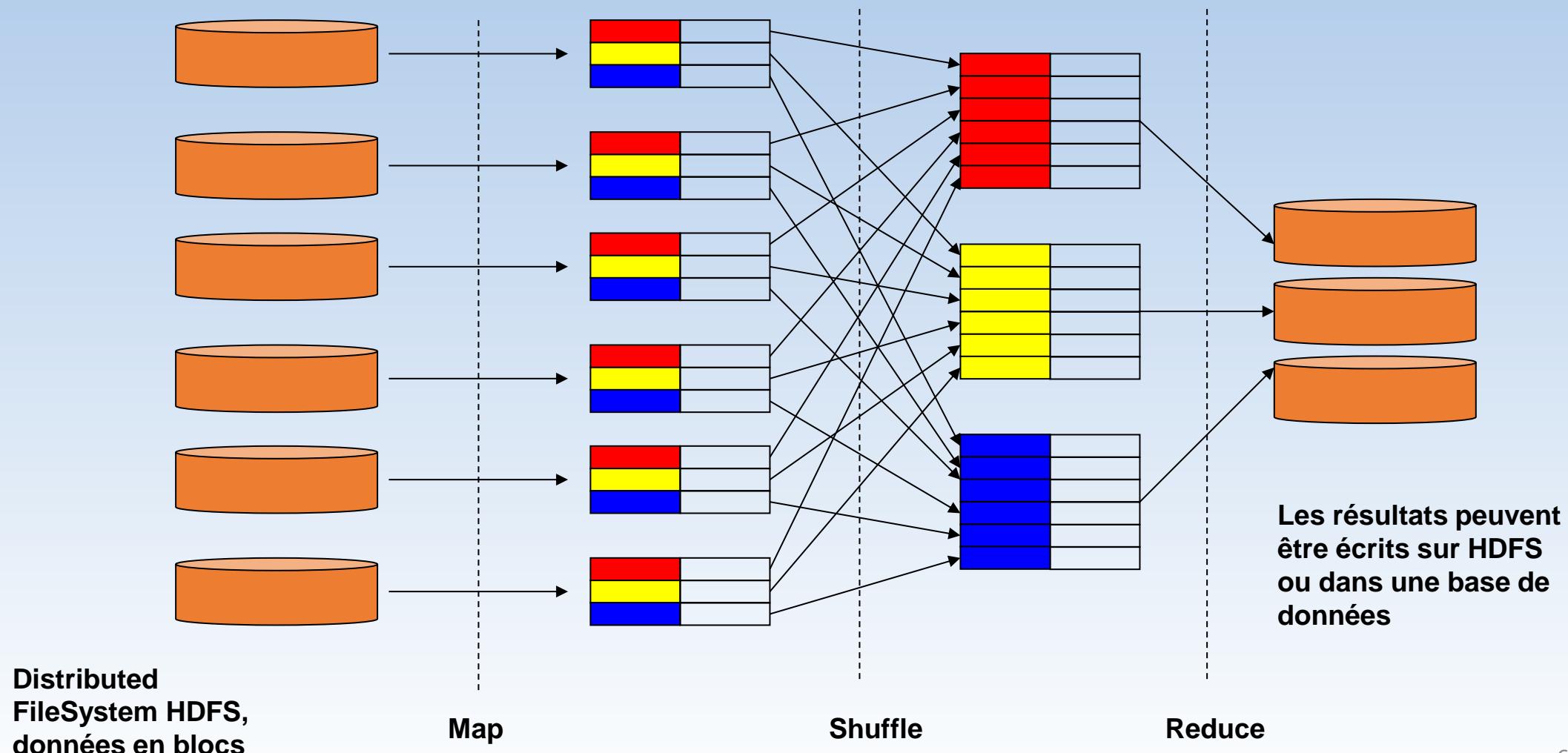
## Architecture de MapReduce



- Si un TaskTracker est très lent, il peut retarder l'ensemble du travail MapReduce, en particulier vers la fin d'un travail, où tout peut finir par attendre la tâche la plus lente. Avec l'exécution spéculative activée, cependant, une seule tâche peut être exécutée sur plusieurs noeuds esclaves.
- Pour la planification des travaux, par défaut, Hadoop utilise FIFO (First in, First Out) et 5 priorités de planification optionnelles pour programmer des tâches à partir d'une file d'attente de travail

# MapReduce

## Présentation de MapReduce

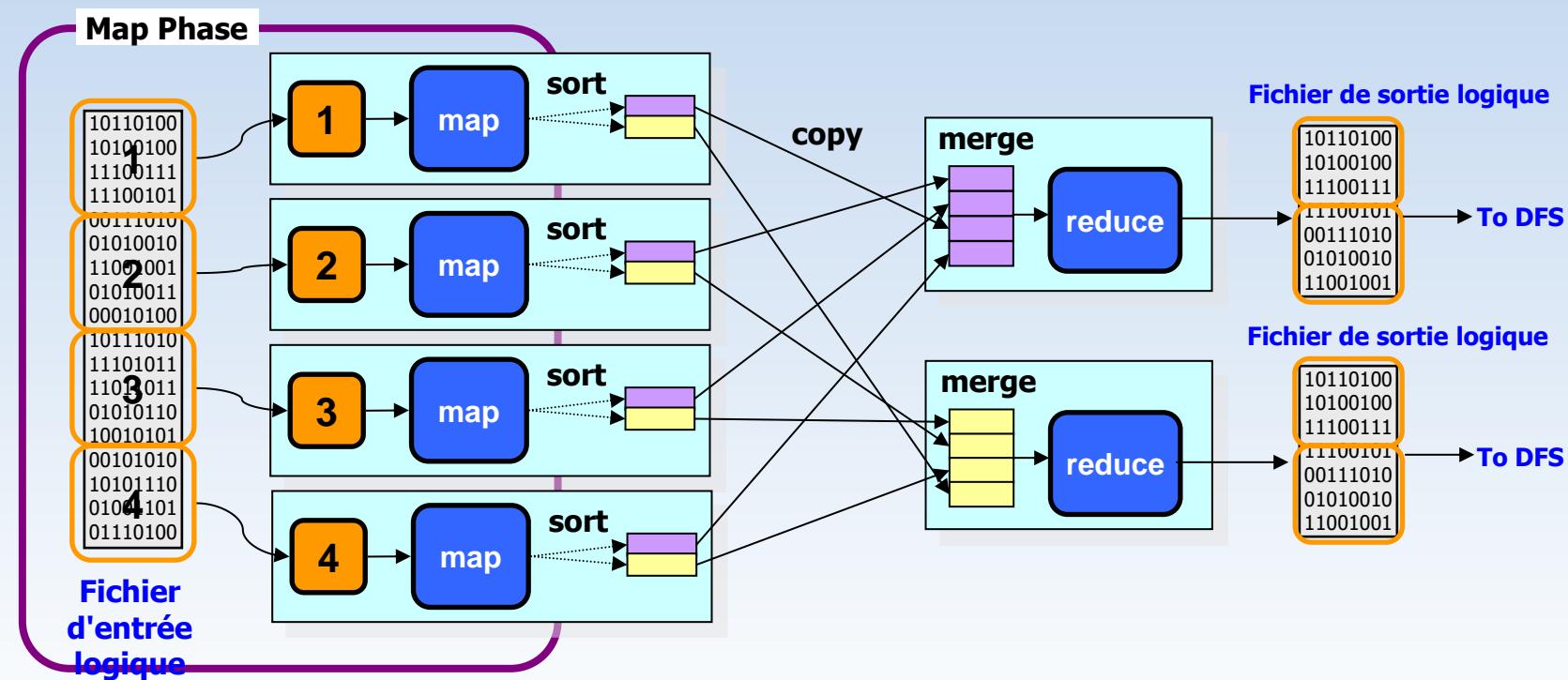


# MapReduce

## Phase Map

### ➤ Mappers

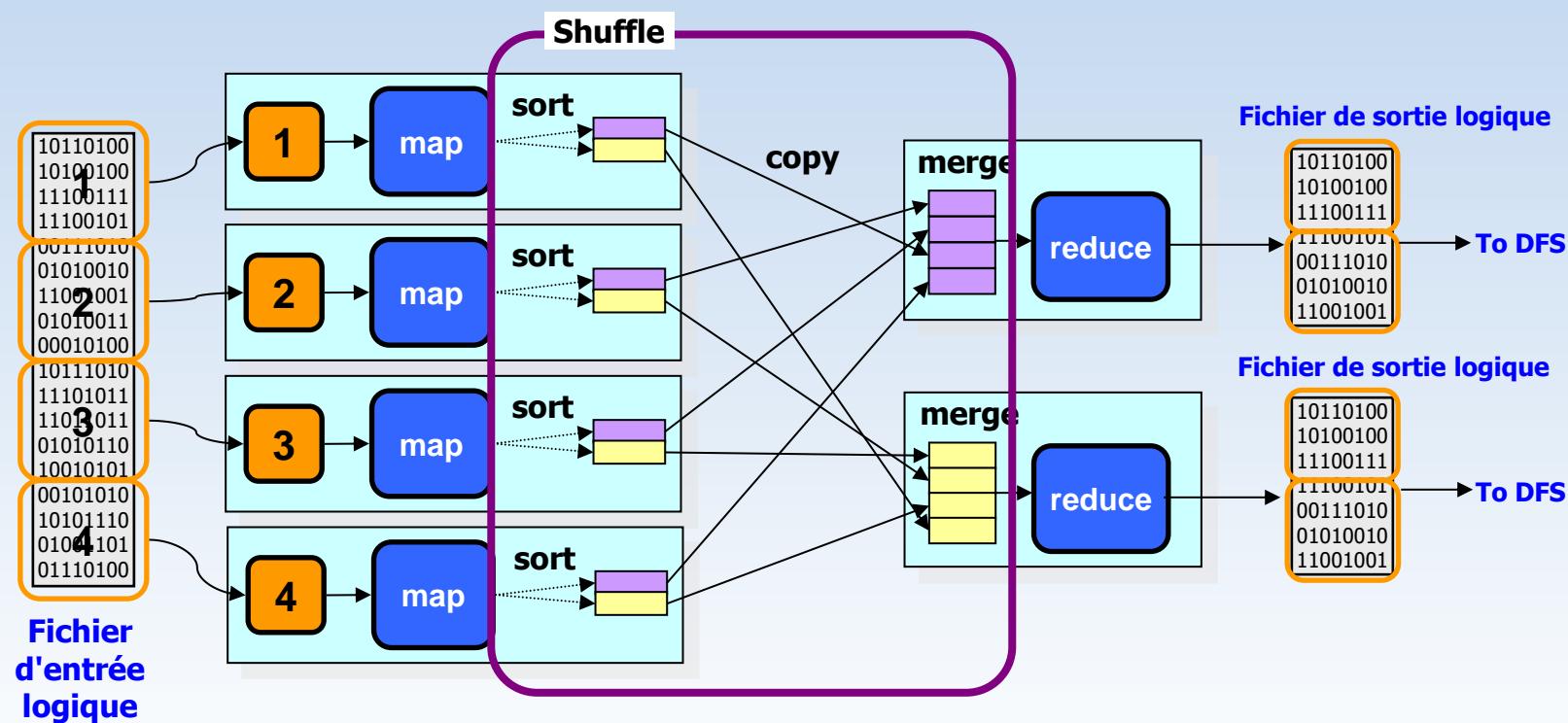
- ✓ Généralement un programme relativement petit avec une tâche relativement simple: il est responsable de lire une partie des données d'entrée, d'interpréter, de filtrer ou de transformer les données au besoin et enfin de produire un flux de paires <key, value>.



# MapReduce

## Phase Shuffle

- ✓ La sortie de chaque Mapper est regroupée localement par clé : key
- ✓ Un nœud est choisi pour traiter les données pour chaque clé unique
- ✓ Tout ce mouvement (shuffle) des données est orchestré de manière transparente par MapReduce

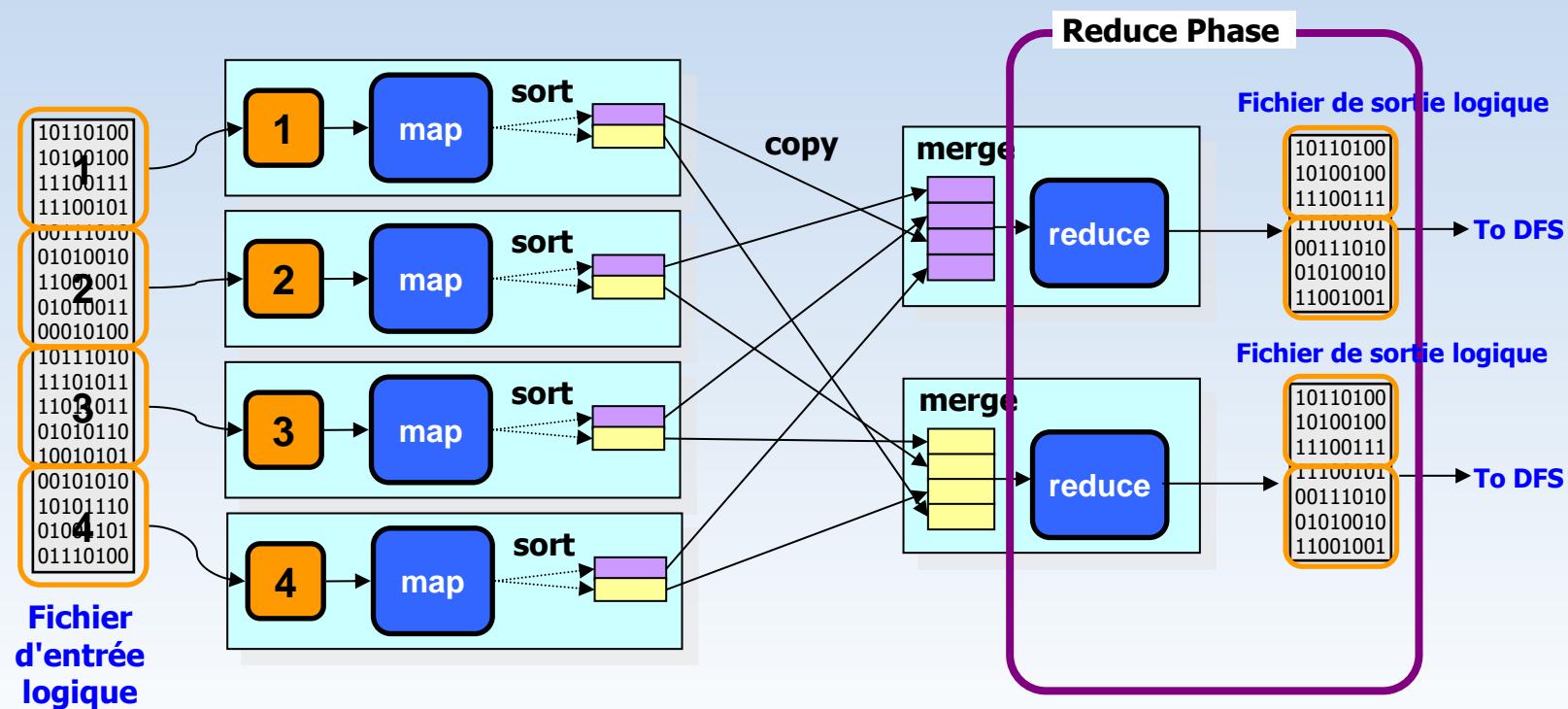


# MapReduce

## Phase Reduce

### ➤ Reducers

- ✓ Petits programmes (généralement) qui regroupent toutes les valeurs de la clé dont ils sont responsables
- ✓ Chaque Reducer écrit sa sortie dans son propre fichier



# MapReduce

## Splits

- Les fichiers dans HDFS sont stockés dans des blocs
- MapReduce divise les données en fragments ou en Splits
  - ✓ Une tâche de Map est exécutée sur chaque Split
- La plupart des fichiers ont des enregistrements avec des points de fin d'enregistrement définis
  - ✓ Le caractère le plus commun est le caractère de fin de ligne
- La classe InputSplitter est responsable de prendre un fichier HDFS et de le transformer en Splits
  - ✓ L'objectif est de traiter autant de données que possible localement

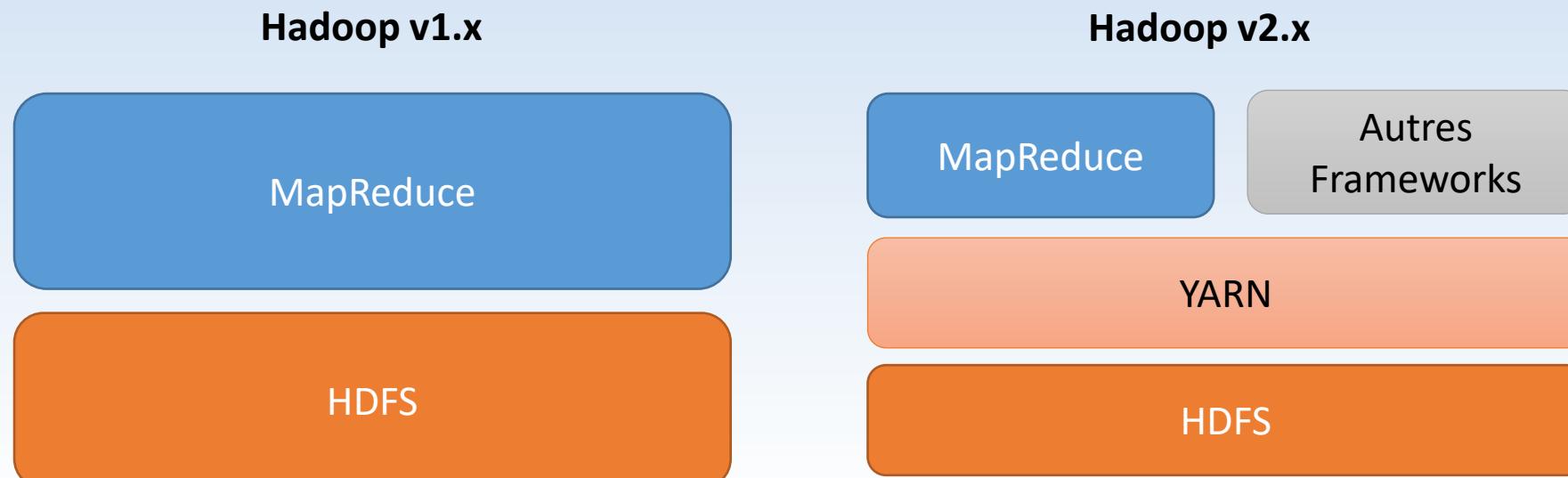
# YARN



# YARN

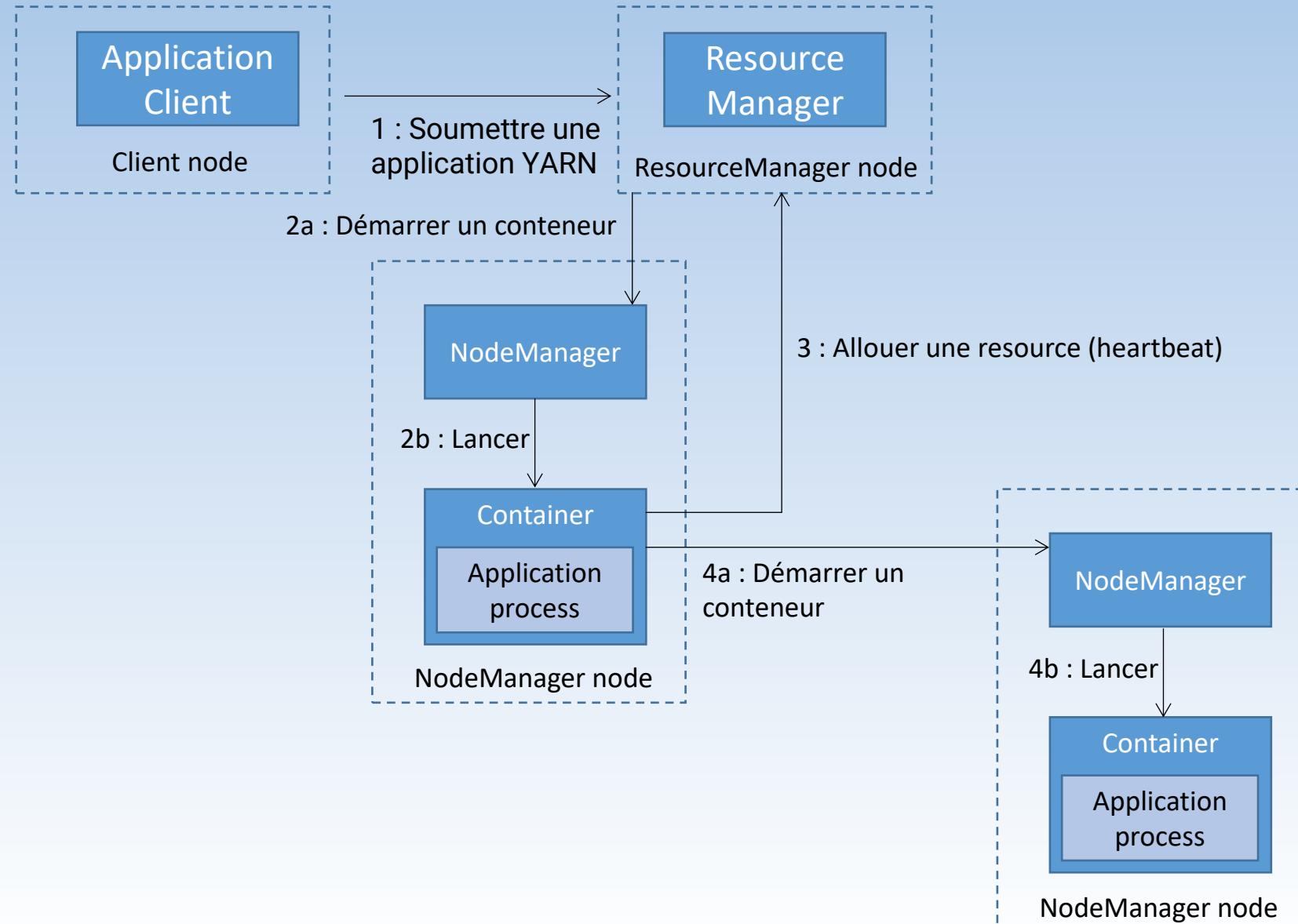
## Qu'est-ce que YARN ?

- ✓ Acronyme de Yet Another Resource Negotiator.
- ✓ Le nouveau gestionnaire de ressources est inclus dans Hadoop 2.x et versions ultérieures.
- ✓ Dissocie la charge de travail Hadoop et la gestion des ressources.
- ✓ Hadoop 2.2.0 inclut la première version de YARN.



# YARN

Exécuter une application dans YARN



# Apache SPARK

# Apache SPARK

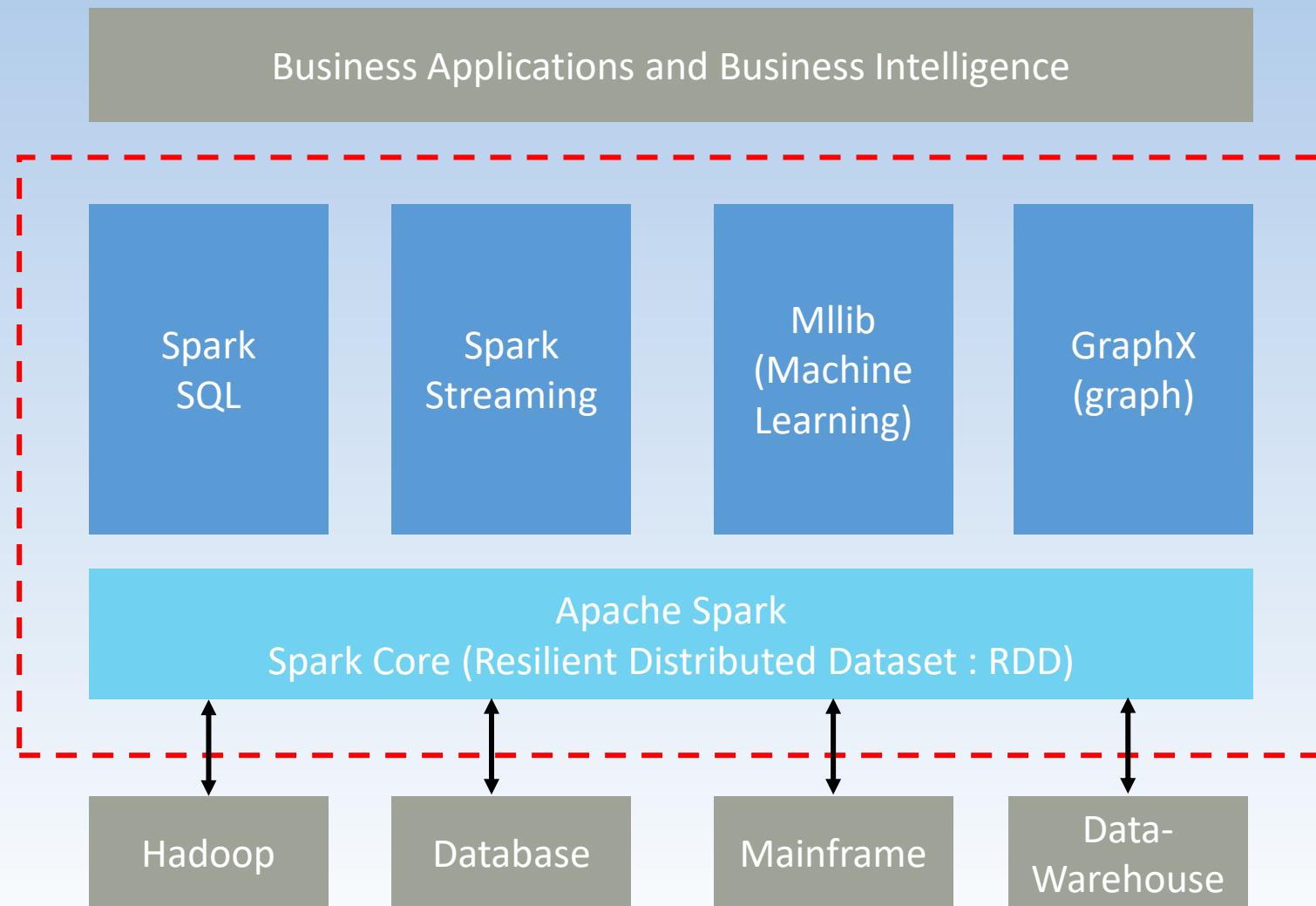
## Qu'est-ce que Apache SPARK ?

- Apache SPARK est un paradigme similaire à MAP-REDUCE
- Apache Spark est nettement plus rapide que MAP-REDUCE !!
- Les développeurs de Spark expliquent que le produit peut exécuter des tâches 100 fois plus vite que MapReduce en cas de traitement en mémoire, et 10 fois plus vite sur disque.
- MAP REDUCE est destiné pour le traitement par lots
- alors que SPARK est mieux pour un composant en temps réel
- Ils sont complémentaires et généralement utilisés ensemble
- Apache Spark prend en charge les langages de programmation Scala, Python, Java et R.



# Apache SPARK

## Qu'est-ce que Apache SPARK ?



# Apache SPARK

## Qu'est-ce que Apache SPARK ?

- **Spark SQL** : Spark SQL est le module d'Apache Spark pour travailler avec des données structurées, fournit des API qui permettent d'intégrer des requêtes SQL dans des programmes Java, Scala ou Python dans Apache Spark.
- **Spark Streaming** : une extension de l'API principale de Spark qui permet un traitement évolutif et tolérant aux pannes des flux de données en direct. Apache Spark streaming : écrit des applications pour traiter les données de streaming en Java, Scala ou Python.
- **Mlib** : une bibliothèque optimisée pour Apache Spark qui prend en charge les fonctions d'apprentissage automatique. Apache Spark Mlib fournit une solution prête à l'emploi pour la classification et la régression, le filtrage collaboratif, le clustering, l'algèbre linéaire distribuée, les arbres de décision, les forêts aléatoires, les arbres à gradient renforcé, l'exploration fréquente de modèles, les métriques d'évaluation et les statistiques.
- **GraphX** : API pour les graphes et le calcul parallèle. GraphX est une abstraction de graphe qui étend les RDD pour les graphes et le calcul parallèle aux graphes.

# Bases de données NoSql

## Bases de données NoSql



# Bases de données NoSql

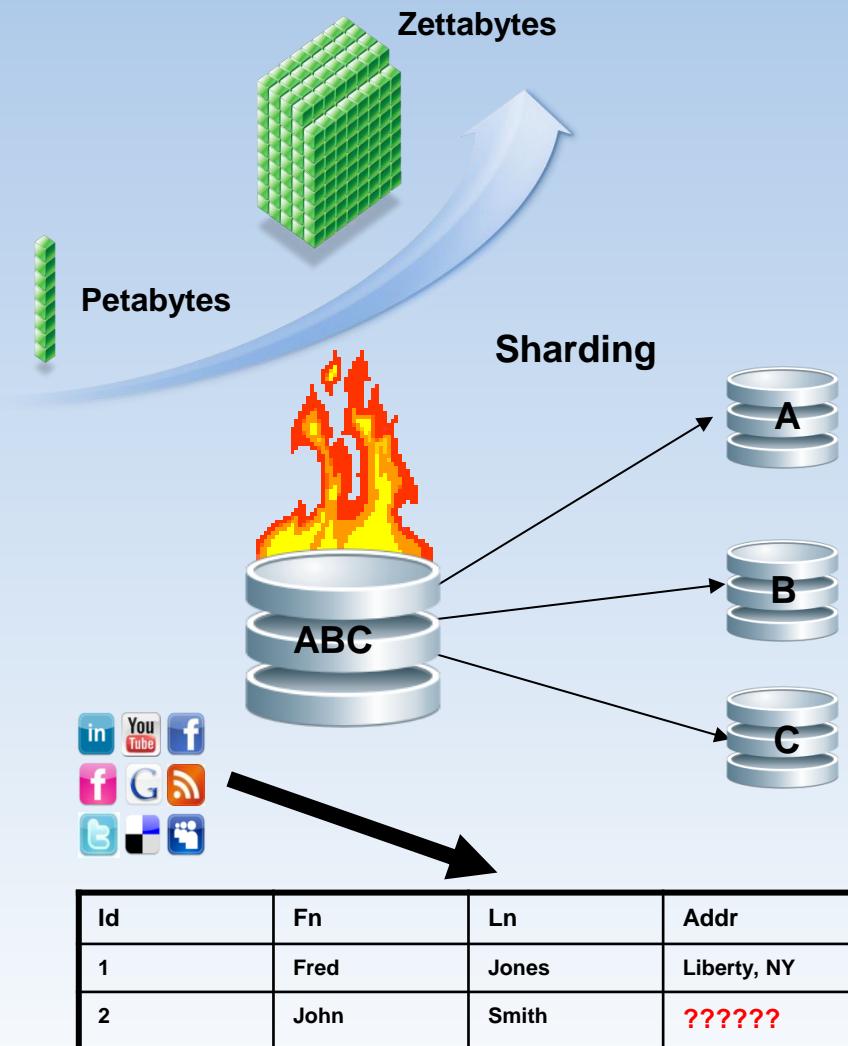
## Qu'est-ce que NoSQL?

- NoSQL signifie "**Not Only SQL**" et se réfère à une nouvelle classe de technologies de base de données créée pour résoudre des problèmes au niveau de BigData
- Les technologies NoSQL ne remplaceront pas les SGBDRs !
- NoSQL :
  - ✓ Pas de schéma / No Schema
  - ✓ RW (read write)
  - ✓ Temps réel (en direct) / Real Time (live)
- Apparues dans les entreprises du web (Google, Yahoo , Amazon, Facebook. . . )
- Pas de règles ACID (Atomicité – Consistance – Intégrité – Durabilité)

# Bases de données NoSql

## Pourquoi NoSQL?

Une technologie rentable est nécessaire pour gérer de nouveaux volumes de données



L'augmentation des volumes de données a entraîné une fragmentation des SGBDR

Des modèles de données flexibles sont nécessaires pour prendre en charge les applications BigData

# Bases de données NoSql

## Modèles

➤ 4 grands modèles :

- ✓ Modèle clé-valeur
- ✓ Modèle Colonnes
- ✓ Modèle Document
- ✓ Modèle Graphe

# Bases de données NoSql

## Modèle clé-valeur

- Le modèle le plus simple.
- A une clé, on associe une valeur.
- La valeur peut être de n'importe quel type (chaîne de caractères, entier, structure, objet sérialisé. . .).
- Chaque objet est identifié par une clé unique
- Les données sont représentées par un couple clé-valeur.



## Bases de données NoSql

### Modèle clé-valeur : Implémentations les plus connues



développé par Amazon



projet sponsorisé par VMWare



implémentation open source  
inspiré de Dynamo

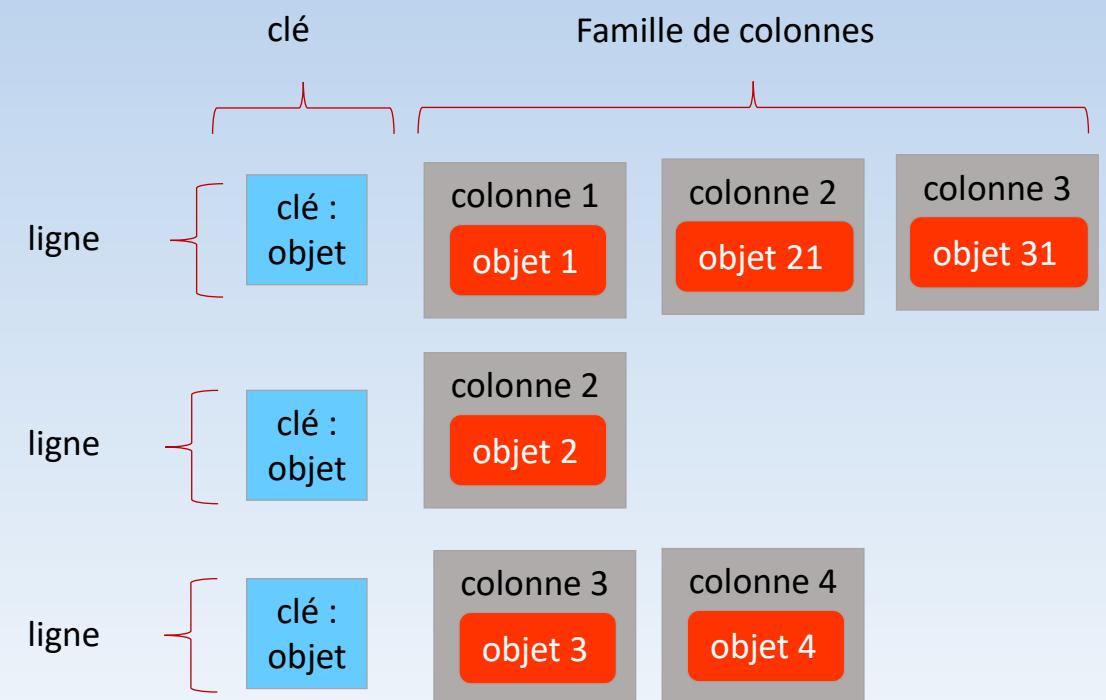


développé par LinkedIn

# Bases de données NoSql

## Modèle orienté colonnes

- Les données sont stockées par colonne
- Ressemble logiquement aux bases de données relationnelles (tables) mais le nombre de colonnes :
  - ✓ est dynamique
  - ✓ peut varier d'un enregistrement à un autre
- Utile pour les données éparses

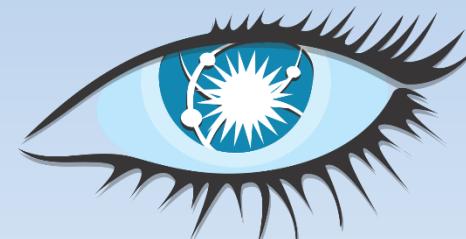


# Bases de données NoSql

## Modèle orienté colonnes : Implémentations les plus connues



Open Source de BigTable de Google utilisé pour l'indexation des pages Web, Google Earth, Google analytics



*cassandra*

Fondation Apache qui respecte l'architecture distribuée de Dynamo d'Amazon, projet né de chez Facebook

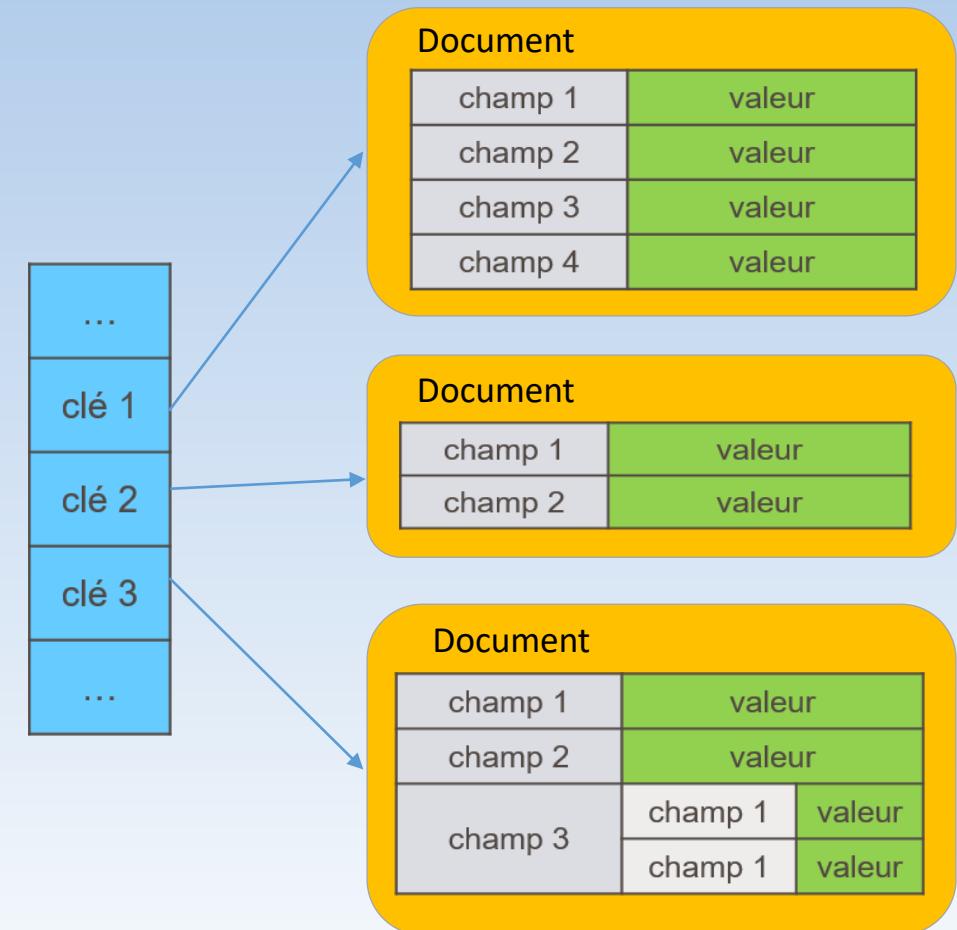


De Amazon

# Bases de données NoSql

## Modèle Document

- Collection de « documents »
- Modèle « clé/valeur », la valeur est un document semi-structuré hiérarchique de type JSON ou XML
- Pas de schéma pour les documents mais une structure arborescente : une liste de champs, un champ a une valeur qui peut être une liste de champs, ...
- Utilisé principalement dans le développement de CMS (Content Management System)



# Bases de données NoSql

## Modèle Document : Implémentations les plus connues



Libre et open source



Distribué sous licence Apache

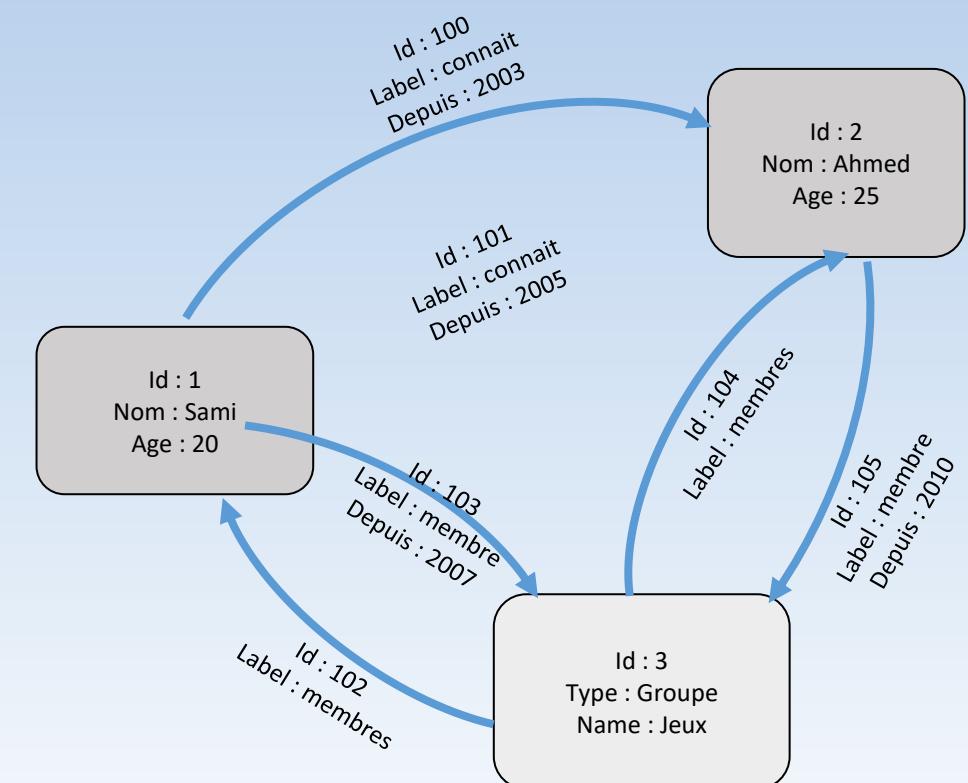


Open Source pour la plate-forme .NET / Windows

# Bases de données NoSql

## Modèle Graphe

- Modèle de représentation des données basé sur la théorie des graphes.
- S'appui sur les notions de nœuds, de relations et de propriétés qui leur sont rattachées.
- Moteur de stockage pour les objets (qui se présentent sous la forme d'une base documentaire, chaque entité de cette base étant un noeud)
- Adapté à la manipulation d'objets complexes organisés en réseaux : cartographie, réseaux sociaux,...



# Bases de données NoSql

## Modèle Graphe : Implémentations les plus connues



open source

développé en Java



open source

développé en Java



Implémentée en Java