



Institut  
Mines-Télécom

# Hadoop

Philippe Lenca, Romain Picot-Clémente,  
Cécile Bothorel, Yvon Kermarrec





- Une approche proposée dans la suite des travaux de Google
- « Un framework Java libre destiné à faciliter la création d'applications distribuées et échelonnables (scalables), permettant aux applications de travailler avec des milliers de nœuds et des péta-octets de données » (wikipédia)
- Un socle pour un **écosystème riche**

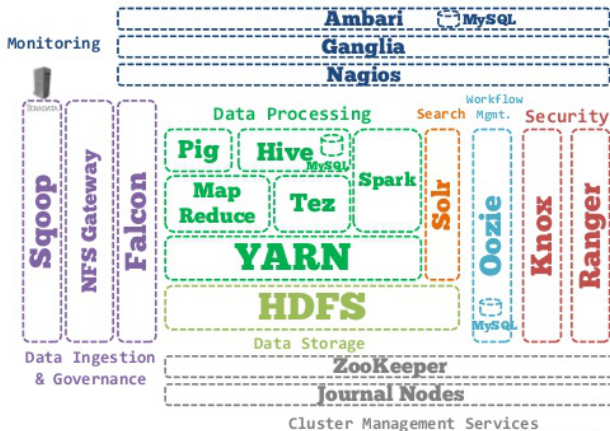
## Cahier des charges pour la conception d'Hadoop



- Un cluster Hadoop doit pouvoir regrouper plusieurs dizaines, centaines ou milliers de nœuds : chaque nœud permet d'offrir du stockage et une puissance de calcul
- Un cluster Hadoop doit pouvoir stocker et traiter des gros volumes de données dans des délais et couts acceptables
- Si un nœud tombe, cela ne doit pas entraîner l'arrêt du calcul ou la perte de données
- Une machine peut être rajoutée dans le cluster pour améliorer les performances

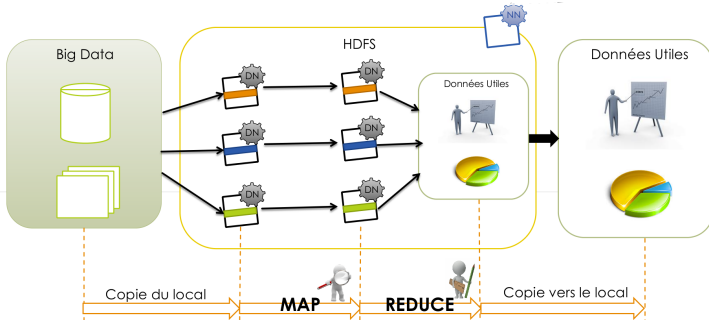
# Un écosystème en expansion

## Apache Hadoop Ecosystem



## Un peu de technique : HDFS et Map-Reduce

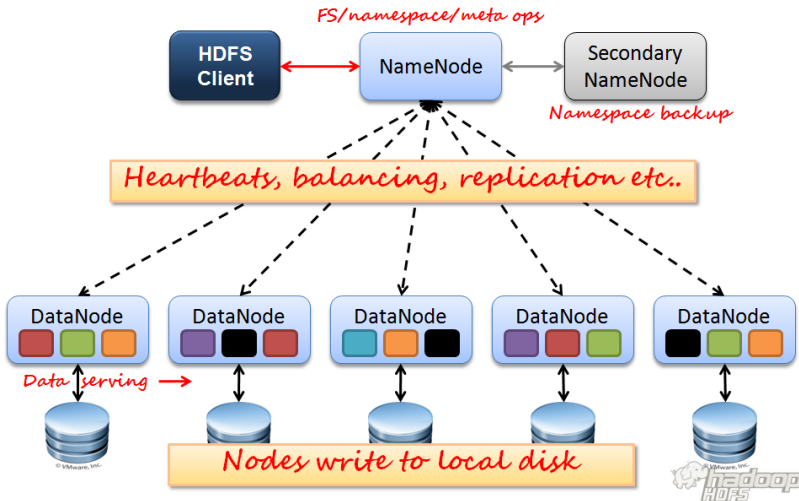
- Le projet Hadoop constitué originellement de deux parties :
  - Stockage des données : **HDFS (Hadoop Distributed File System)**
  - Traitement des données : **Map-Reduce**
- Principe :
  - Diviser les données
  - Les sauvegarder sur un ensemble de machines, appelé cluster
  - Traiter les données directement là où elles sont stockées



## HDFS : Hadoop Distributed File System

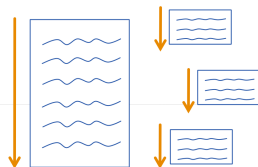
- Un « nouveau » système de gestion de fichiers (SGF) pour lire et écrire des données sur le cluster
- Des **blocs de taille importante** : 64 MO par exemple au lieu de 4Ko pour NFS
- Un SGF particulier de type « write once » **adapté au stockage de flux de données**
- Chaque bloc est sauvegardé 3 fois, au moins, pour augmenter le **disponibilité** et la **sécurité** des données
- HDFS repose sur des SGF classiques et donc des disques standards sont utilisés

## HDFS architecture



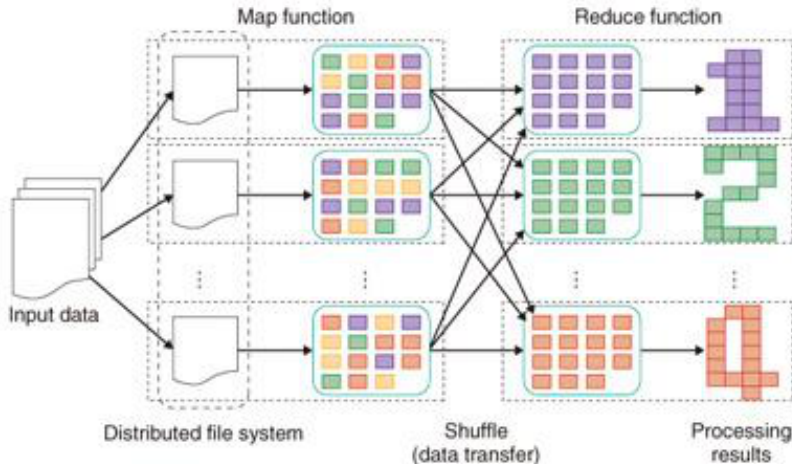
## Map-Reduce de Google

- Patron d'architecture de développement permettant du calcul haute performance (« High performance computing ») sur plusieurs milliers de machines
  - Réduit le déplacement des données entre machines, qui est la source de la complexité en distribué
- Fournit un très haut niveau de transparences aux utilisateurs
  - Masque la parallélisation des traitements
  - Prend en charge la tolérance aux pannes
  - Gère l'équilibrage des charges et la coordination
- Fournit un modèle « relativement » simple à comprendre et à programmer





# Map-Reduce de Google

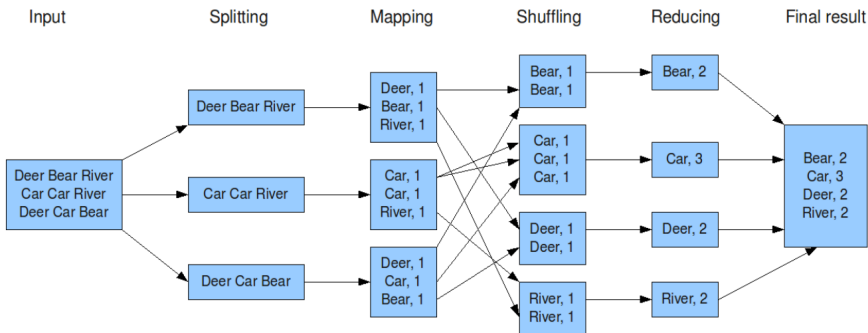




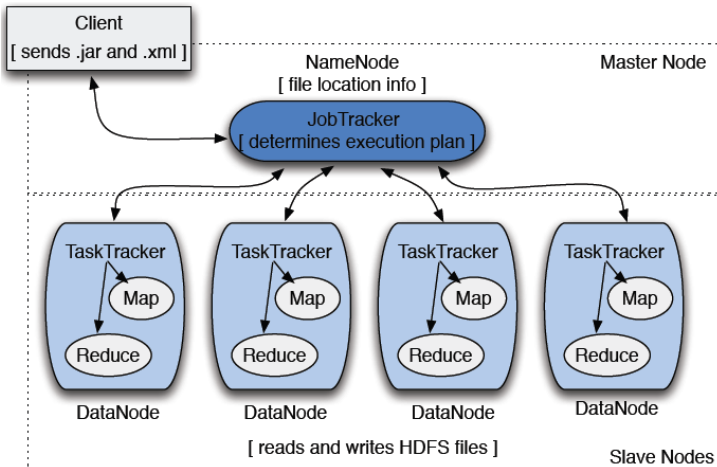
## Map-Reduce en action

- L'entité de base : la paire (clé, valeur)
- Une fonction de map qui traite les entrées et produit des couples (clé, valeur)
- Une fonction de reduce qui applique un traitement sur les données locales de type (clé, valeur)
- Une fonction intermédiaire shuffle & sort entre les fonctions map et reduce

## Map-Reduce en action : comptage de mots

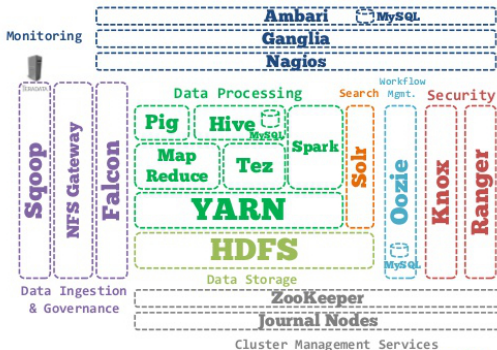


## Derrière la scène : une infrastructure puissante



## Plus loin avec l'écosystème Hadoop

- Des langages de haut niveau pour le traitement des données (étapes Map-Reduce transparentes) : Hive, Pig
- Des alternatives à Map-Reduce qui utilisent la RAM partagée, particulièrement adaptées au machine learning, à l'analyse de graphe : Tez, Spark (très grande communauté)
- Des outils de gestion des données, de monitoring, de gestion du cluster, de sécurité, ...



## Comment se lancer ?

- Apprendre avec une machine virtuelle de type Cloudera :
  - Utilisation de HDFS, premiers programmes avec map-reduce, initiation aux outils de plus haut niveau, etc.
- Utiliser ensuite les différents modes de fonctionnement d'Hadoop
  - Modes local, pseudo-distribué, totalement distribué ou virtualisé
  - Bâtir progressivement son cluster à partir de matériels recyclés puis achetés spécifiquement

## Pour une entreprise : un choix du DSI

- Un cluster dédié : installé physiquement dans l'entreprise ou chez un hébergeur
  - + confidentialité des données, maîtrise du cluster, totale liberté
  - investissement, administration du cluster et « tuning »
- Un cluster dans le cloud
  - + tarif compétitif, fiabilité et disponibilité élevés
  - limitations techniques, configurations imposées, confidentialité des données