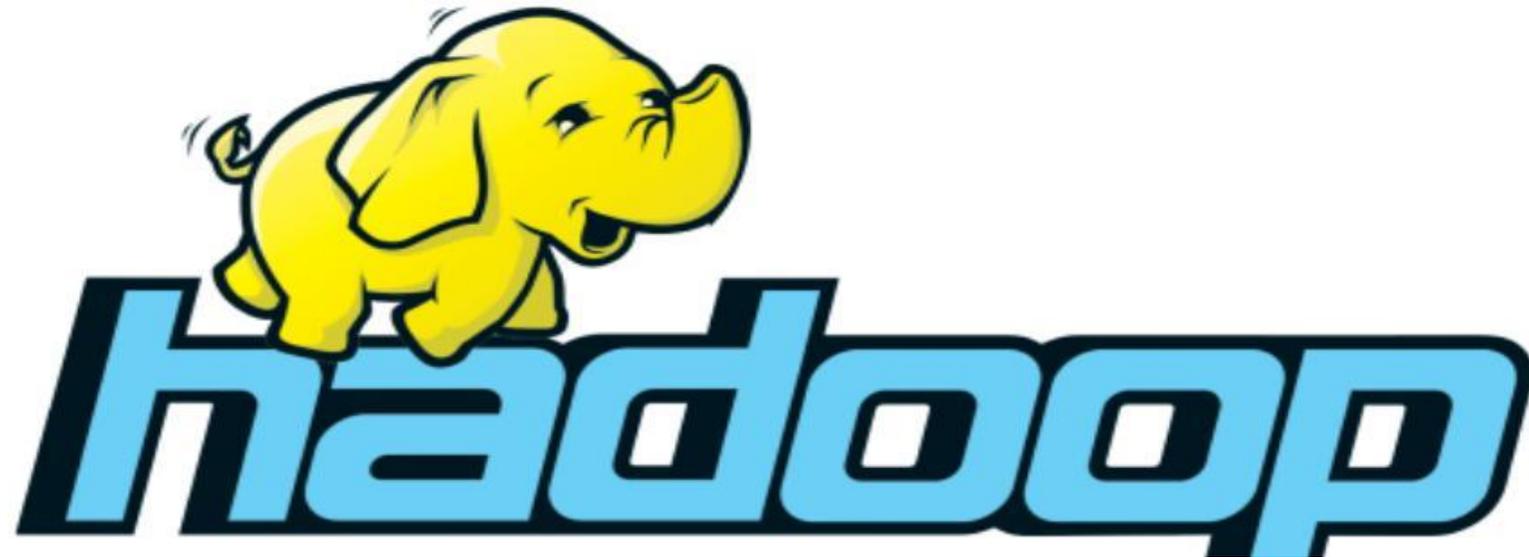


# Le Framework Hadoop



# Traitement parallèle des données

L'exécution d'un traitement informatique sur multitude de machines différentes (un cluster de machines) de manière transparente.

- **Tolérance aux pannes** : une machine en panne faisant partie du cluster ne doit pas produire d'erreur pour le calcul dans son ensemble
- **Transparence** : le cluster dans son ensemble doit être utilisable comme une seule et même machine traditionnelle
- **Accès et partage des ressources** : pour toutes les machines
- **Extensibilité** : on doit pouvoir ajouter de nouvelles machines pour le calcul si nécessaire
- **Hétérogénéité**: les machines doivent pouvoir avoir différentes architectures

# Cluster

## Un cluster de calcul (grappe de serveurs)

imaginez 5000 ordinateurs connectés entre eux formant un cluster) est conçu spécifiquement pour stocker et analyser de grandes quantités de données non structurées au sein d'un environnement de calcul distribué.

Le travail d'analyse de données est réparti entre les différents nœuds du cluster (serveurs). Ces clusters permettent de lancer le logiciel de traitement distribué Hadoop sur des ordinateurs low-cost.



# Calcul distribué

Le calcul distribué désigne l'exécution d'un traitement informatique sur une multitude de machines différentes (un cluster de machines) de manière transparente.

## Problèmes soulevés

- Accès et partage des ressources pour toutes les machines.
- Extensibilité : on doit pouvoir ajouter de nouvelles machines pour le calcul si nécessaire.
- Tolérance aux pannes : une machine en panne faisant partie du cluster ne doit pas produire d'erreur pour le calcul dans son ensemble,
- Transparence : le cluster dans son ensemble doit être utilisé comme une seule et même machine
- Hétérogénéité : les machines doivent pouvoir avoir différentes architectures et caractéristiques.

# Calcul distribué

Deux approches ont vu le jour :

## Cluster local

Mise en place d'un cluster de machines local pour accélérer le traitement.

Solution coûteuse et complexe à mettre en œuvre.

### Exemple : le supercalculateur Blue Gene 1999

- Premier supercalculateur commercialisé par IBM (en plusieurs exemplaires).
- Connecte plus de 130000 CPUs et 32 téra-octets de RAM.
- Contrôle centralisé pour assurer l'exécution des tâches distribuées.

## Cluster via Internet

Exploiter la démocratisation de l'informatique moderne et la bonne volonté des utilisateurs du réseau pour créer un cluster distribué via Internet à moindre coût.

# Calcul distribué

Pour mettre en place un système distribué il faudrait recourir :

Soit à des super calculateurs : **Très couteux, souvent trop puissants par rapport aux besoins...**

Soit à des solutions développés en interne : **Nécessite des investissements et des compétences couteuses.**

## **Alternative proposée**

Mettre en place un outil qui puisse être installé et configuré rapidement et qui permettrait à des développeurs d'exécuter des tâches distribuées avec un minimum de formation requise.

L'outil doit être facile à déployer, simple à supporter, et pouvant permettre la création de clusters de taille variables extensibles à tout moment.

# Hadoop

**Hadoop** : Framework facile à déployer, qui permet l'exécution de tâches parallélisables de manière rapide et simple à mettre en œuvre . Projet de la fondation Apache Open Source, composants complémentaires ouverts (tout le monde peut participer).

Modèle simple pour les développeurs, il suffit de développer des tâches depuis des interfaces accessibles via des bibliothèques dans de multiples langages.

S'occupe de toutes les problématiques liées au calcul distribué :

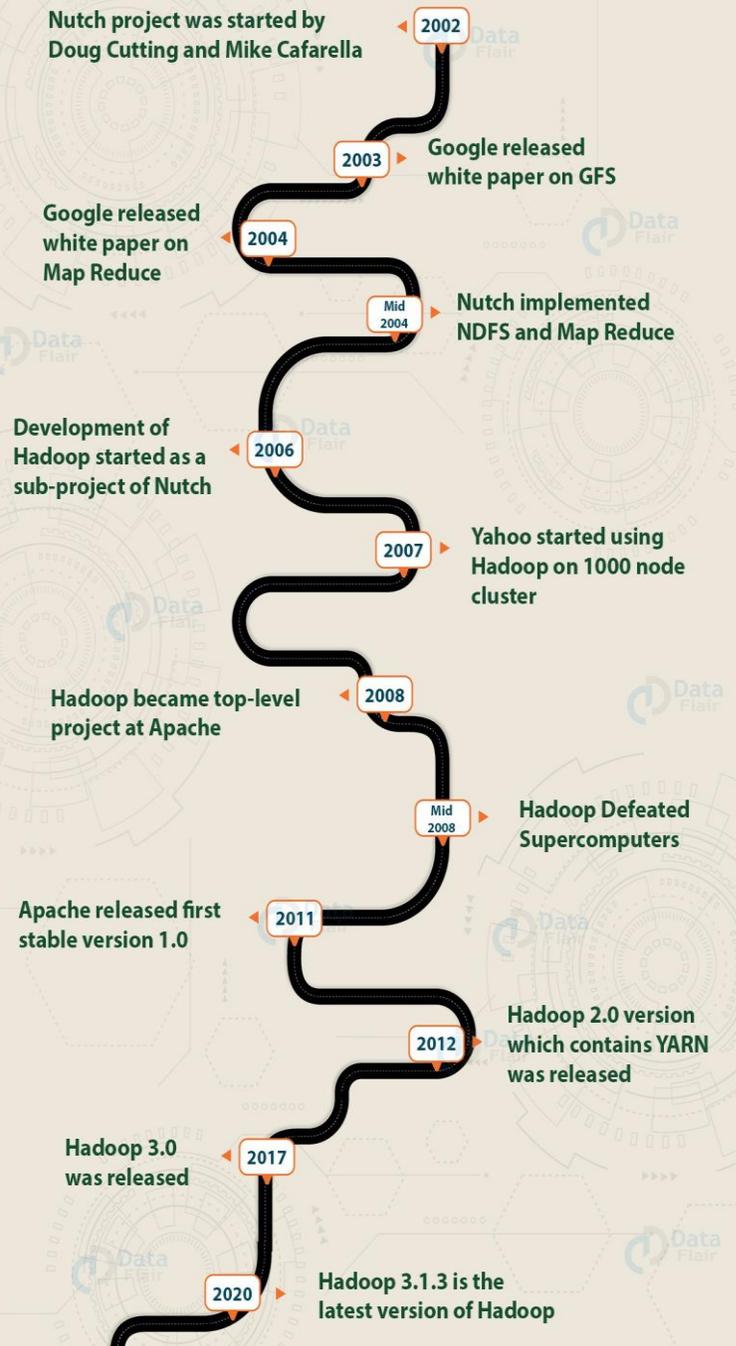
- L'accès et le partage des données
- La tolérance aux pannes
- Répartition des tâches entre les différentes machines du cluster

# Hadoop

Hadoop est une plateforme (framework) open source conçue pour réaliser d'une façon distribuée des traitements sur des volumes de données massives, de l'ordre de plusieurs pétaoctets. Ainsi, il est destiné à faciliter la création d'applications distribuées et échelonnables (scalables). Il s'inscrit donc typiquement sur le terrain du Big Data.

À l'instar des autres systèmes opérationnels, Hadoop possède les structures de base nécessaires à effectuer des calculs: un système de fichiers, un langage de programmation, une méthode pour distribuer les programmes ainsi générés à un cluster, un mode pour obtenir les résultats et arriver à une consolidation unique de ces derniers, ce qui est le but.

# Hadoop – Evolution historique



Version	Release date
3.3.4	2022 Aug 8

# Hadoop - Avantages

La plateforme Hadoop permet de vérifier les critères les plus importants des solutions Big Data à savoir :

- **La scalabilité** : assurée par la performance délivrées des clusters de haut niveau.
- **Parallélisme** des données: un même traitement et calcul peut être effectué sur toutes les données.
- **Performance** : Hadoop délivre un support de traitement de grands volumes de données et d'énormes data sets à travers son **modèle de calcul distribué**.
- **Hadoop offre une flexibilité** : permettant de stocker des données non structurées en tout genre telles que des textes, des symboles, des images ou des vidéos.
- **L'élasticité** : Il suffit de changer le nombre de nœuds dans un cluster pour étendre ou réduire le système.

# Hadoop - Limites

La plateforme Hadoop présente aussi des faiblesses :

- **l'algorithme MapReduce n'est pas toujours adéquat.** Il convient pour les requêtes d'informations les plus simples, mais pas pour les tâches itératives.
- Hadoop n'offre pas d'outils adaptés et compréhensibles. On déplore aussi **le manque d'outils pour la standardisation des données** et la détermination de qualité.
- **Hadoop est difficile à maîtriser.** Il y a donc peu de programmeurs suffisamment compétents pour utiliser MapReduce. C'est la raison pour laquelle beaucoup de fournisseurs ajoutent une technologie de base de données SQL par dessus Hadoop. Les programmeurs maîtrisant SQL sont nettement plus nombreux.
- **Sécurité des données.** Le protocole d'authentification Kerberos aide toutefois à sécuriser les environnements Hadoop.

# Qui utilise Hadoop ?



# Hadoop-Composants

Le fonctionnement de Hadoop repose sur deux parties essentielles :

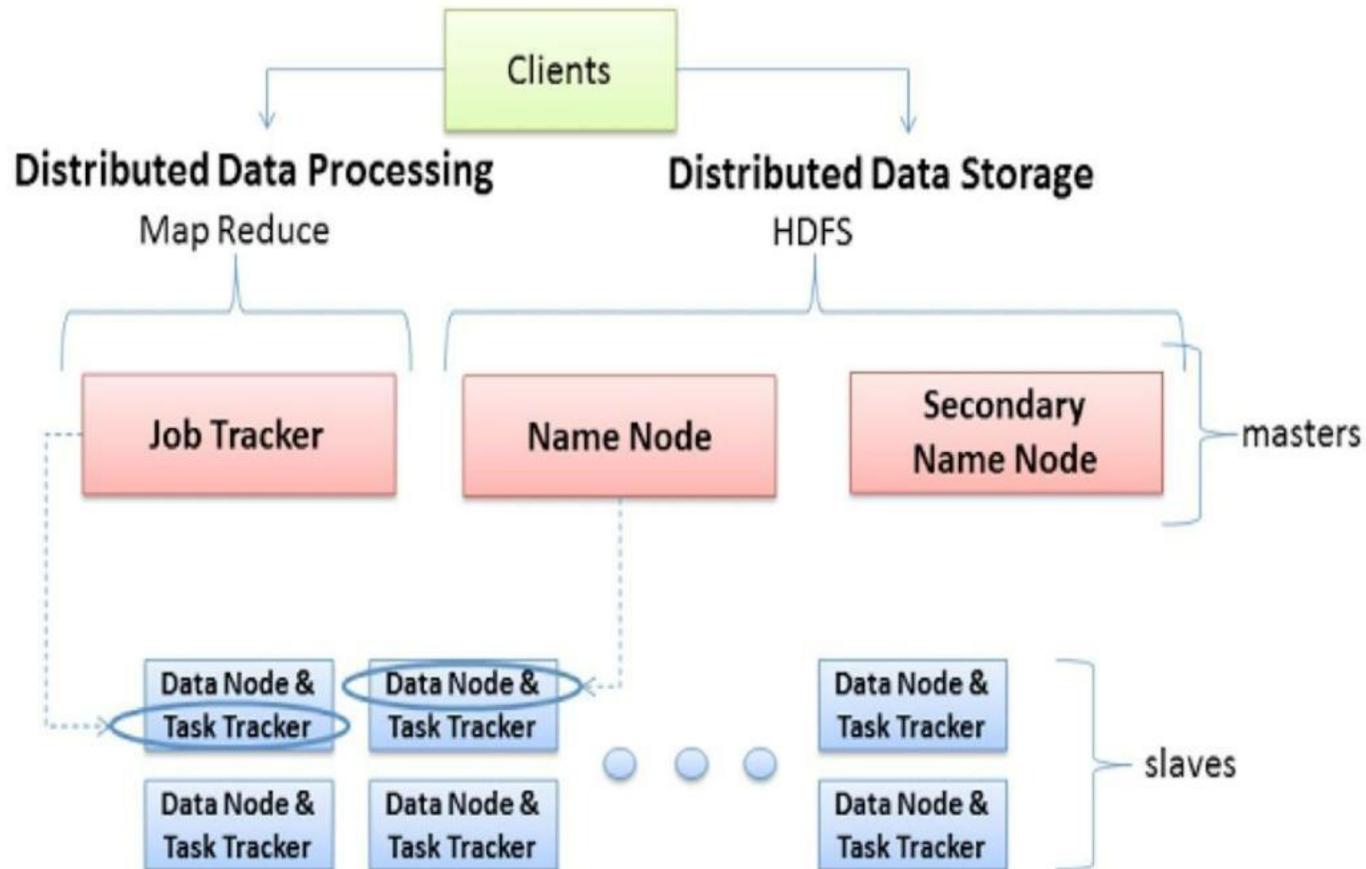
- **Stockage des données** : HDFS (Hadoop Distributed File System) **Cours 4**
- **Traitement des données** : MapReduce / Yarn **Cours 5**

## Principe :

- Diviser les données
- Les sauvegarder sur une collection de machines, appelées cluster
- Traiter les données directement là où elles sont stockées, plutôt que de les copier à partir d'un serveur distribué
- Il est possible d'ajouter des machines à votre cluster, au fur et à mesure que les données augmentent

# Hadoop – Architecture

Il faut tout d'abord savoir qu'une architecture Hadoop est basée sur les deux principaux rôles maître / esclave. Des sous-rôles relatifs au système de fichiers et à l'exécution de tâches distribuées sont associés à chaque machine.



# Hadoop – Architecture (Machines maitres)

- **JobTracker** : Il s'agit de la responsabilité pour une machine maître de lancer des tâches distribuées, en coordonnant les esclaves. Le JobTracker, planifie les exécutions, gère l'état des machines esclaves et agrège les résultats de calculs dans le cadre de MapReduce.
- **NameNode** : Le rôle de NameNode intervient dans le fonctionnement du système de fichiers distribué HDFS. Une machine maître associée à ce rôle a pour responsabilité de répartir les données sur les machines esclaves et de gérer l'espace de noms du cluster. Elle contient les méta-datas lui permettant de savoir sur quelle machine chaque fichier est hébergé.
- **SecondaryNameNode** : Ce rôle intervient dans le cadre de la redondance du NameNode. Généralement assumé par une autre machine physique que le NameNode, il permet, en cas de panne de celui-ci, la continuité de fonctionnement du cluster Hadoop.

# Hadoop – Architecture (Machines esclaves)

- **TaskTracker** : Il s'agit du rôle permettant à un esclave d'exécuter une tâche MapReduce sur les données qu'elle héberge. Le TaskTracker est piloté par le JobTracker d'une machine maître qui lui envoie la tâche à exécuter.
- **DataNode** : Comme son nom l'indique, il s'agit d'un nœud de l'architecture hébergeant une partie des données du cluster. Les nœuds de données sont généralement répliqués dans le cadre d'une architecture Hadoop afin d'assurer la haute disponibilité des données.

Lorsqu'un client a besoin d'accéder à une donnée ou d'exécuter une tâche distribuée, elle passe par la machine maître jouant les rôles de JobTracker et de NameNode.

# L'écosystème Hadoop

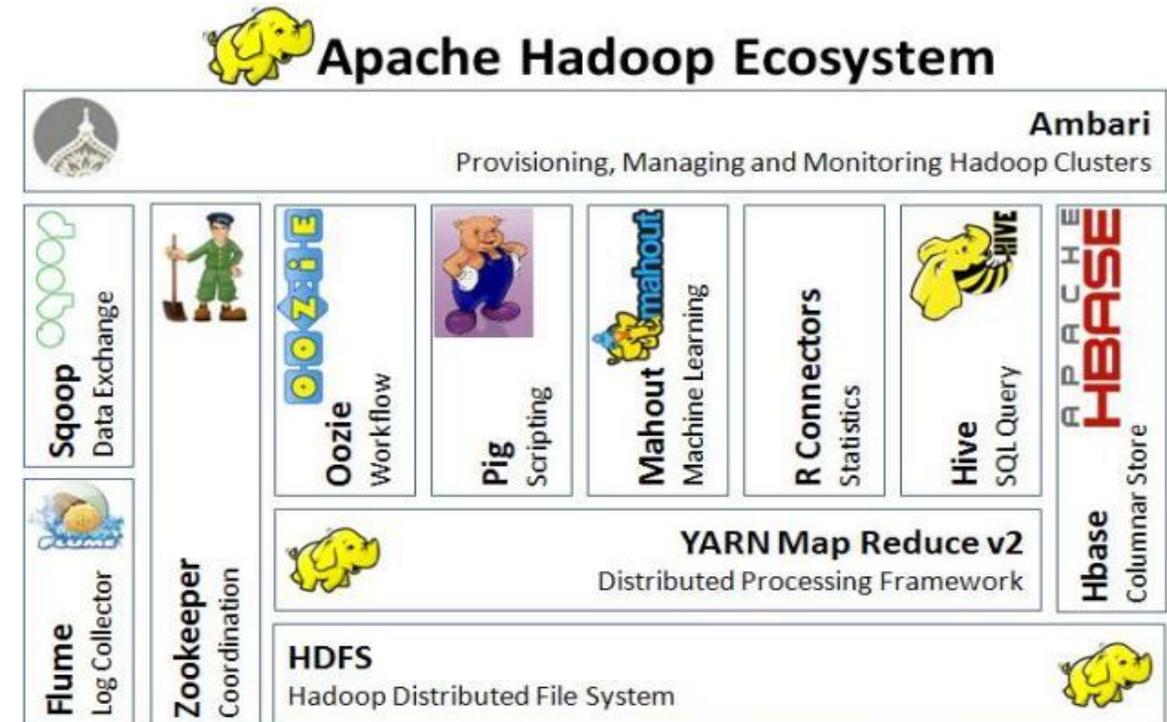
Projet	Sommaire
Hadoop Core	HDFS et MapReduce
HBase	Base de données distribuée
Hive	Logiciel d'analyse de données avec syntaxe similaire au SQL
Impala	Logiciel d'analyse de données à très faible latence, compatible avec Hive et HBase
Mahout	Collection d'algorithmes d'apprentissage machine
Pig	Logiciel d'analyse de données utilisant un langage de script
Sqoop	Utilitaire pour transférer des données entre des base de données relationnelles (RDBMS) et Hadoop.
ZooKeeper	Service de coordination et de gestion de configurations pour systèmes distribués
etc.	

# Hadoop : Stockage de données

**HDFS (Hadoop Distributed File System)** : Système de stockage de fichiers distribué de Hadoop.

**HBASE** : Base NoSQL orientée colonnes, système de gestion de base de données pour un environnement distribué.

**Hive** : Une infrastructure d'entrepôt de donnée intégrée sur Hadoop permettant l'analyse, le requêtage via un langage proche syntaxiquement de SQL ainsi que un langage la synthèse de données.



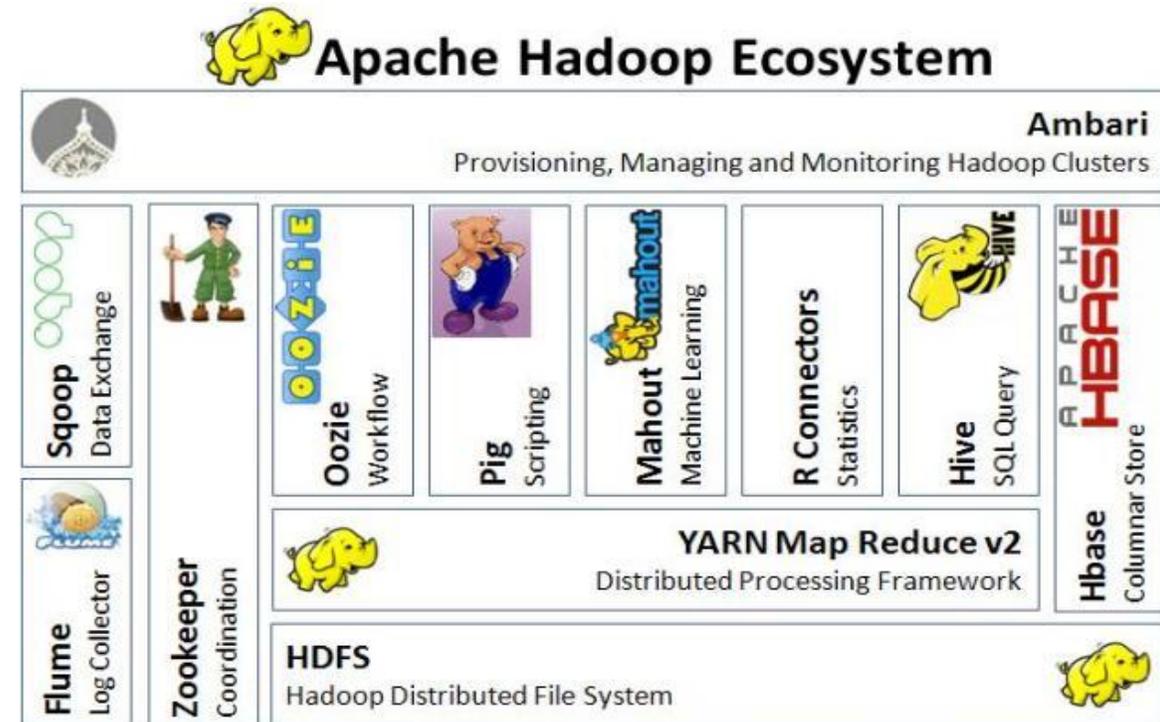
# Hadoop : Récupération de données

**Apache Flume** : Une solution de collecte, agrégation et transfert de gros volumes de logs.

**Apache Storm** : Système distribué qui permet la récupération et le traitement en temps réel.

**Spark Streaming** : Librairie de spark permettant la récupération et le traitement de donnée en temps réel.

**Apache Sqoop** : Ce projet permet de faire le lien entre un système de gestion de base de données relationnel (SGBDR) et HDFS.

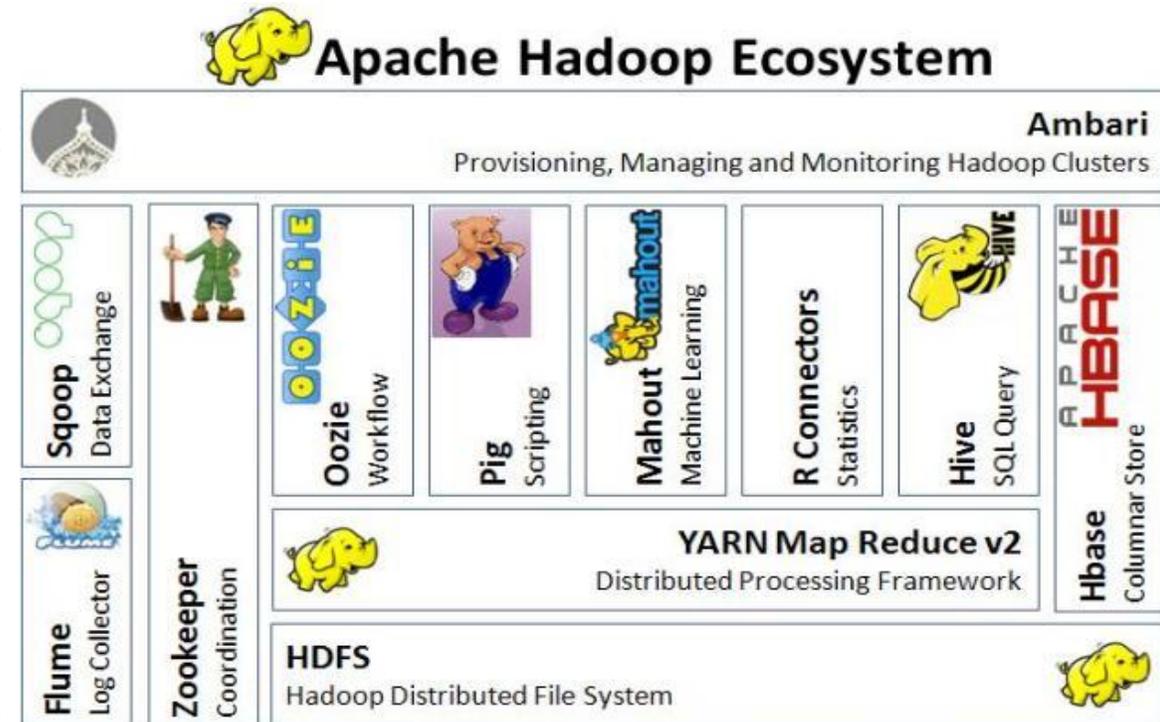


# Hadoop : Moteurs d'interrogation et de manipulation des données

**MapReduce** : Framework open source java, permettant la manipulation des données dans un environnement distribué.

**Impala** : Moteur SQL distribué qui va 20 fois plus vite que MapReduce mais n'est pour l'instant qu'un complément car il n'offre pas encore toutes les fonctionnalités de MapReduce.

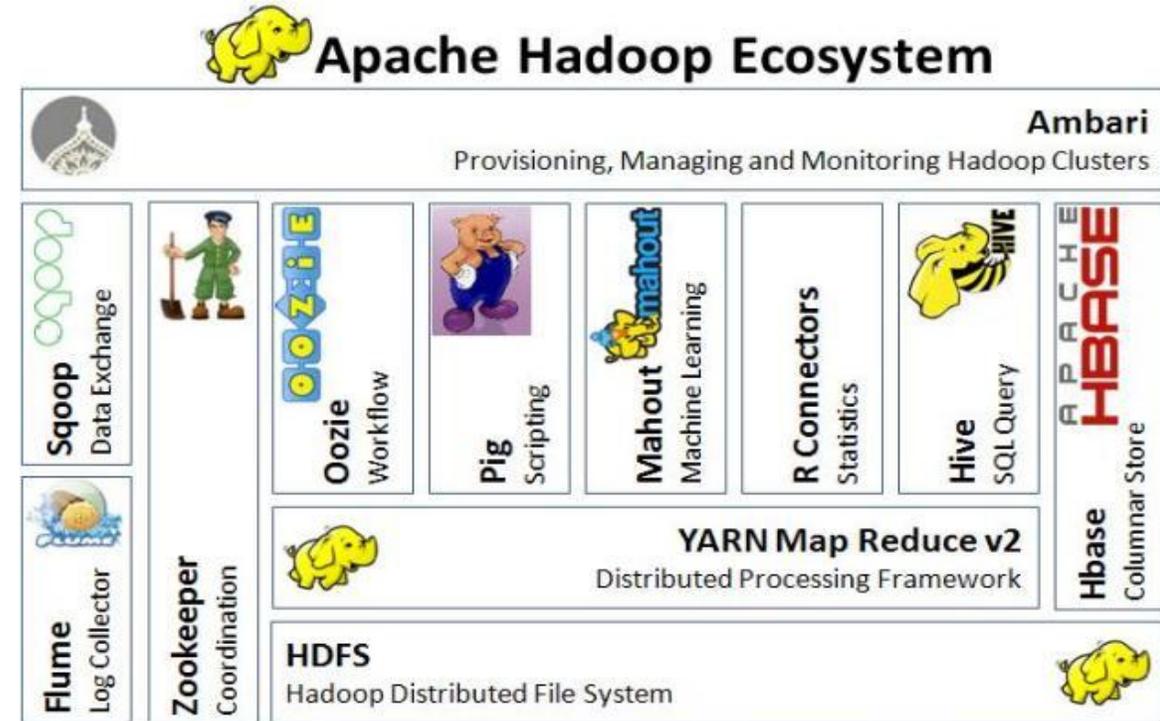
**Spark** : Permet la manipulation des données dans un environnement distribué. Il peut aussi bien faire les traitements sur disque ou tout en mémoire. Il va 10 fois plus vite que MapReduce sur disque et 100 fois plus vite en mémoire.



# Hadoop : Moteurs d'interrogation et de manipulation des données

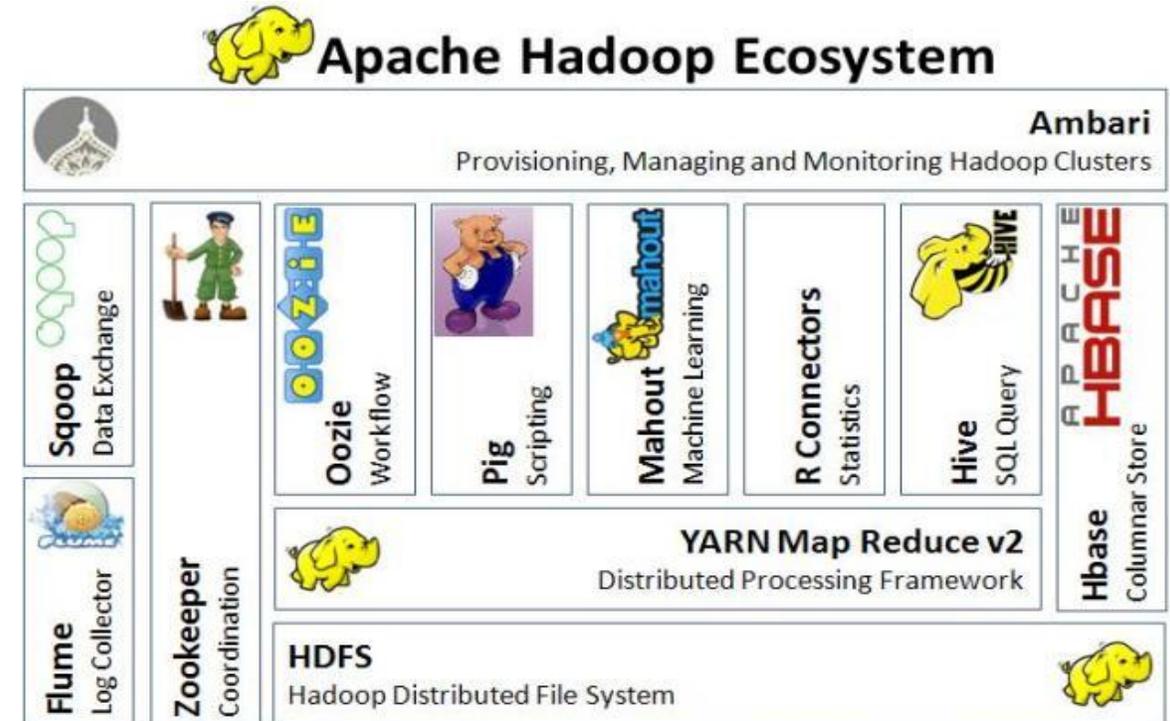
**Pig** : Moteur permettant de manipuler tous types de données avec un langage beaucoup plus intuitif que le java MapReduce. Une fois exécuter le code Pig génère du MapReduce.

**Mahout** : Librairie java contenant des algorithmes de machine Learning déjà codé en MapReduce.



# Hadoop : Ordonnanceurs

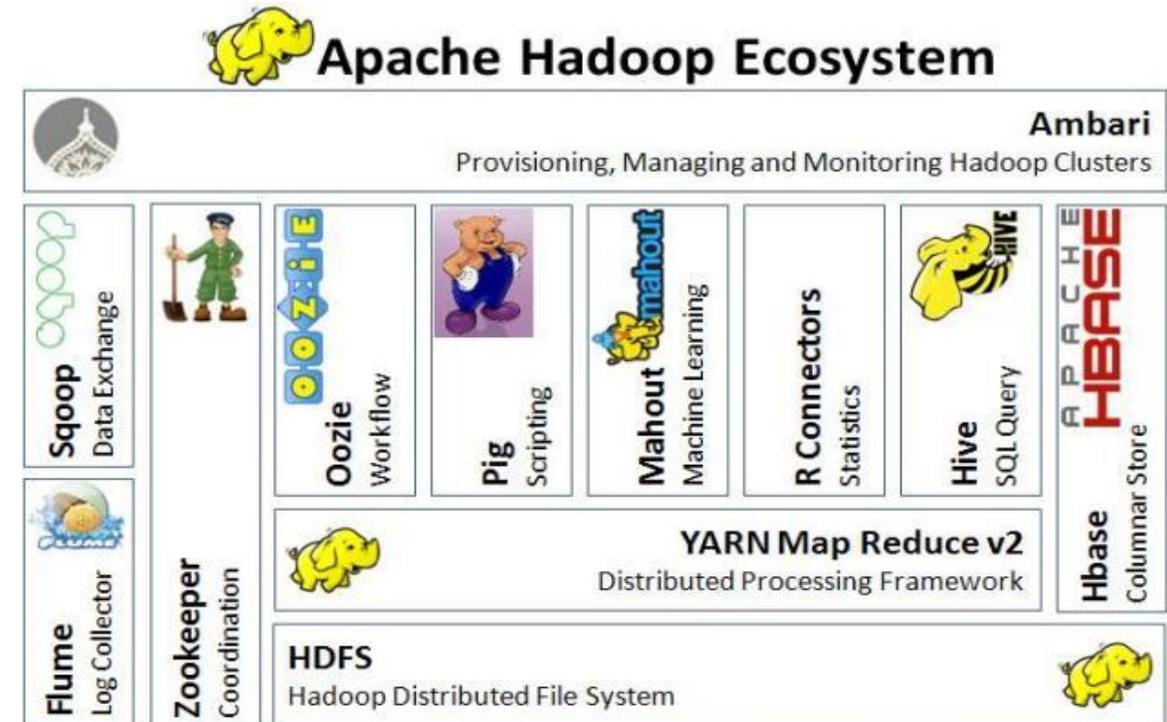
**Oozie** : Permet d'ordonnancer et de lancer tous types d'applications (spark, impala, shell, hive, pig ....) dans l'écosystème Hadoop.



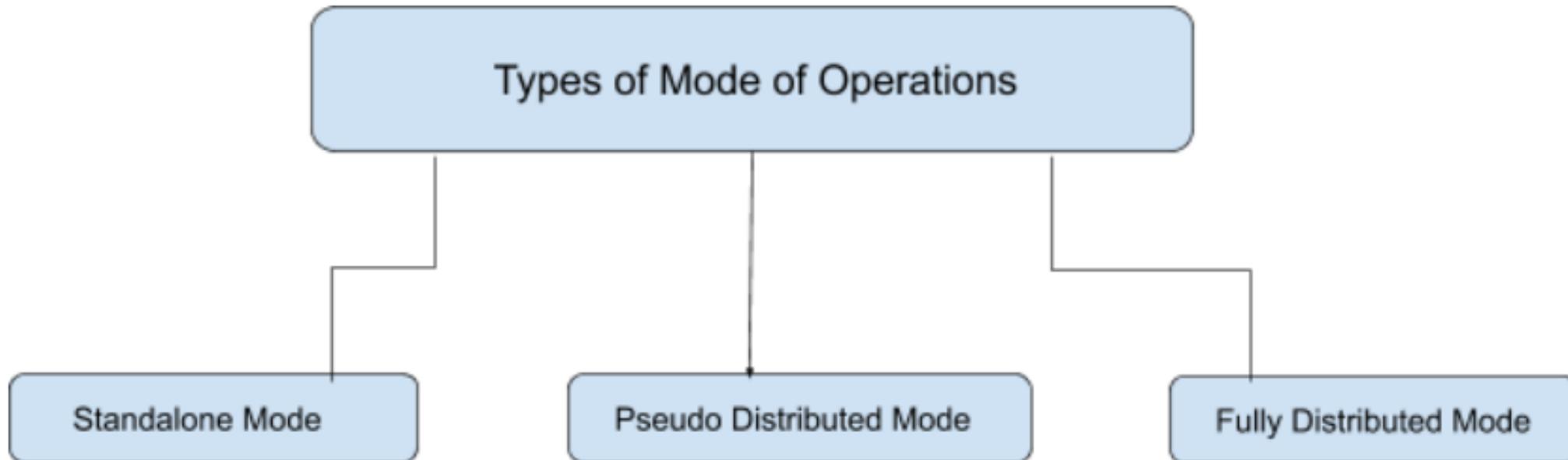
# Hadoop : Interfaces et coordinations

**Zookeeper** : Permet de mettre en œuvre la coordination des services de l'écosystème et leur synchronisation dans un environnement distribué. Seul les administrateurs peuvent être amené à interagir avec lui.

**Hue** : Interface web qui permet d'accéder directement aux principaux modules (hive, pig, sqoop, spark, impala, etc.....). Nativement on peut accéder aux modules en ligne de commandes. Hue apporte une interface commune et agréable d'utilisation.



# Modes de fonctionnement de Hadoop



Nativement hadoop est compatible Linux. Ce n'est que depuis la version hadoop 2.0 que l'on peut l'installer sur Windows.

L'utilisateur peut opter pour le mode de fonctionnement d'adoptant le mieux à ses besoins.

# Fonctionnement : Mode Local

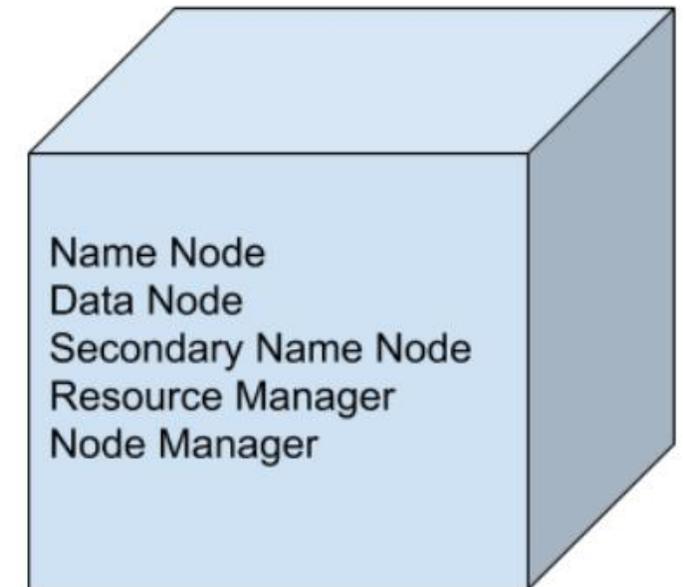
En mode local, Hadoop fonctionne sur une seule station de travail et les cinq daemons de Hadoop (*NameNode*, *SecondaryNameNode*, *DataNode*, *JobTracker* et *TaskTracker*) s'exécutent tous dans la même JVM (*Java Virtual Machine*).

Si l'on ne prête pas attention à ces points, un programme fonctionnant parfaitement en mode local pourra ne pas fonctionner du tout en mode pseudo-distribué ou totalement distribué.

*Le fonctionnement en mode local est donc très éloigné du fonctionnement d'un vrai cluster Hadoop et donc rarement utilisé, sauf pour une première approche de Hadoop, des tests très simples, ou pour déboguer un programme.*

# Fonctionnement : Mode pseudo-distribué

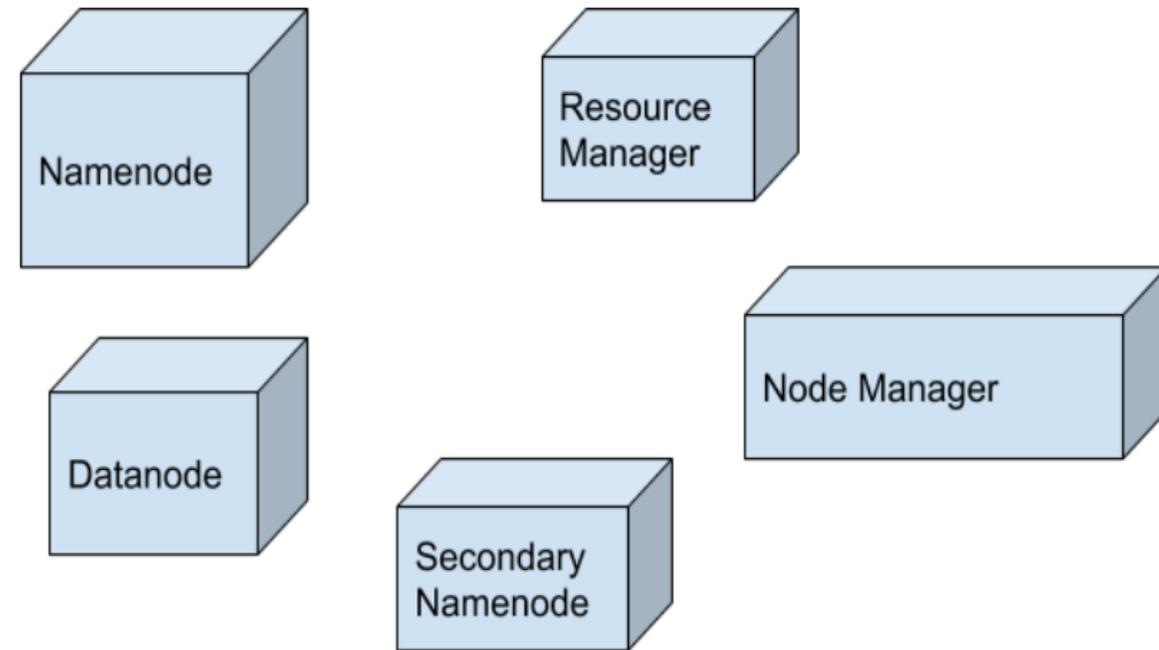
Dans ce mode, Hadoop fonctionne également sur une seule machine tout comme le mode local mais la différence s'inscrit au niveau des daemons dans la mesure ou chaque daemon s'exécute dans sa propre JVM.



Pseudo-distributed Mode

# Fonctionnement : Totalelement distribué

Dans ce mode, on s'aligne plus au fonctionnement réel d'un cluster Hadoop avec un ensemble de machines interconnectées dans un réseau et avec des daemons sur des JVM indépendantes.



**Fully-distributed mode**