

Développeur Big Data HDFS & MapReduce



Rencontre avec **Hadoop**

Rencontre avec Hadoop

Une plateforme à grand succès pour lancer massivement des jobs sur des systèmes distribués

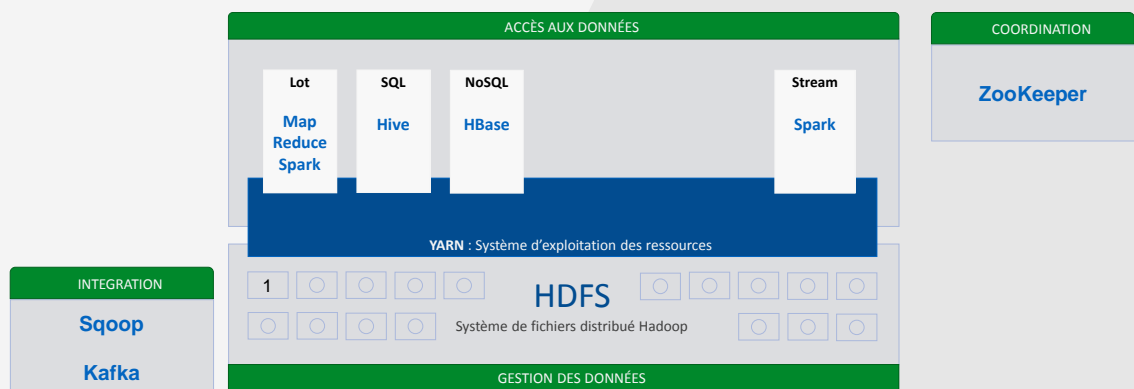
Plusieurs composants dont

- Système de fichier distribué d'Hadoop (**HDFS**)
- Gestionnaire de ressources **YARN**
- Couche **MapReduce**

Hadoop est une implémentation à MapReduce & Google File System (GFS) inventé par Google utilisés dans les data centers pour principalement :

- Construction & maintenance de l'**index inversé**
- Exécution de l'algorithme de **PageRank (de recherche)**

L'écosystème Hadoop



HDFS – Hypothèses & Objectifs

- Panne matériel
- Accès aux données en continu
- Données de tailles importantes
- Model de cohérence de données simple (une seule écriture, plusieurs lectures) -> immutable
- Mieux vaut déplacer un traitement que déplacer une donnée

HDFS – Définitions

- HDFS vient de Hadoop Distributed File System
- C'est un système de fichiers qui maintient des fichiers à travers les nœuds du cluster. Il sauvegarde les fichiers, chaque fichier a un nom et situé dans un répertoire spécifique
- Il supporte la plupart des opérations connues pour les systèmes de fichier ordinaires
- Chaque cluster HDFS est composé :
 - Un ou deux **NameNode** et
 - Plusieurs **DataNode**

HDFS – NameNode & DataNode

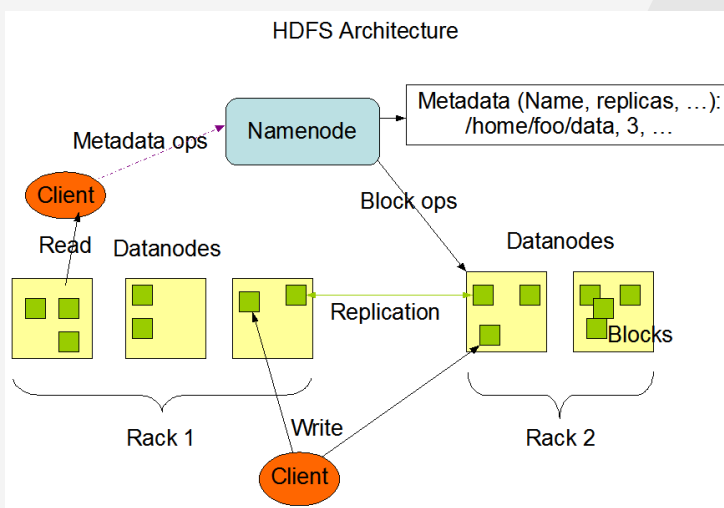
NameNode (un ou deux par cluster) :

- Gère les métadonnées et les répertoires du cluster de manière centralisée
- Il est le nœud maître du cluster
- Détermine et maintient comment les données sont découpées et distribuées à travers le cluster
- Il est unique mais dispose d'un backup en cas de panne

DataNode (un ou plusieurs par cluster)

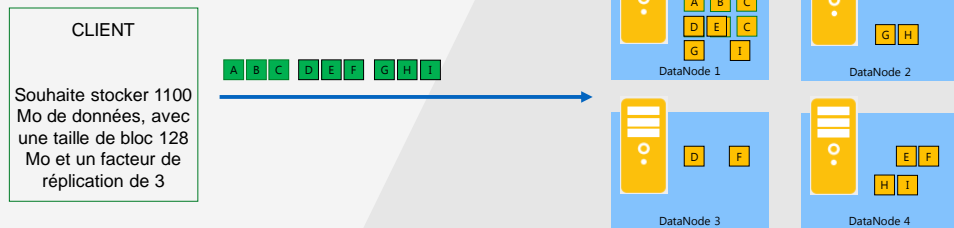
- Il stocke et restitue une partie des données (chunk of data) et responsable de la réplication de la donnée sur les autres nœuds
- Gère les opérations sur les blocs (création, suppression)
- Nombre de réplicas est de 3 par défaut (mais peut être personnalisé par fichier)
- La taille par défaut d'un bloc est de 128MB sur la plupart des clusters

HDFS – Architecture



HDFS – Réplication

1. Découpage en blocs
2. Ecriture sur les DataNode
3. Réplication au niveau des DataNode

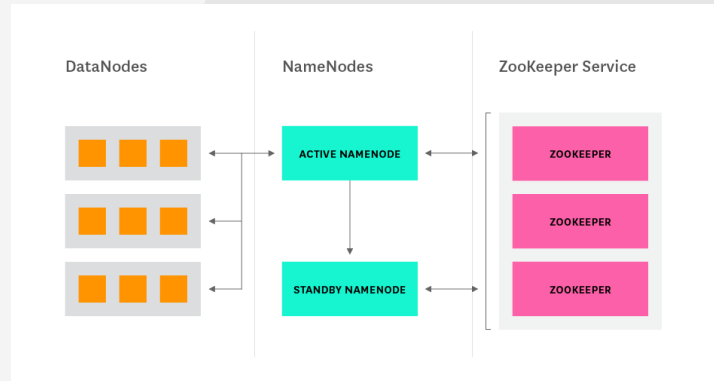


HDFS – Pannes

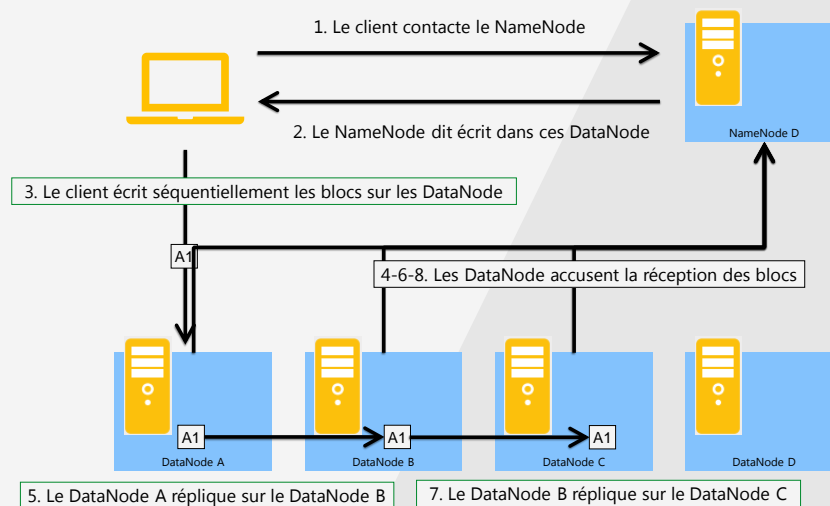
- Chaque DataNode envoie périodiquement un signal de vie « heartbeats » au NameNode
- Si le NameNode ne réceptionne pas de signal de vie au bout de X minutes, alors le DataNode sera considéré comme éteint/hors service
- Le NameNode initie la réplication des blocs qui se trouvaient sur le DataNode tombé sur un autre DataNode

Apache ZooKeeper & Interaction avec HDFS

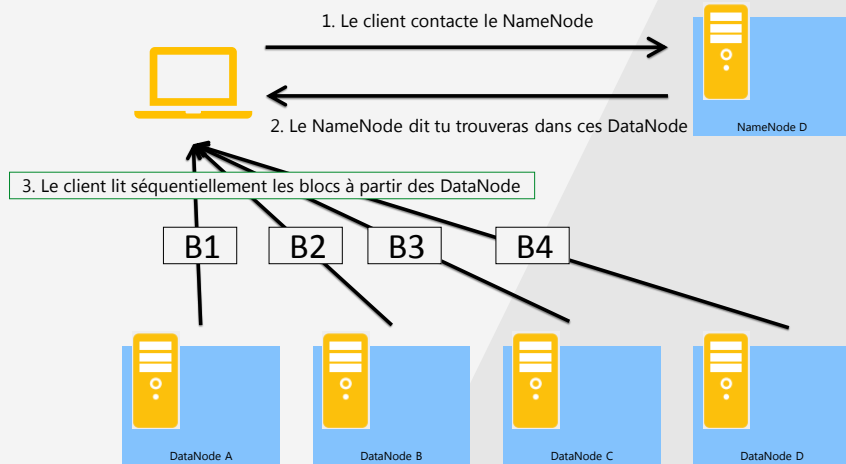
- Server open source qui coordonne de manière fiable les traitements distribués
- Fournit des services opérationnels pour le cluster Hadoop
- Fournit :
 - un service de configuration distribué,
 - un service de synchronisation
 - un registre de nommage pour les systèmes distribués.
- Les applications distribuées l'utilisent pour stocker et gérer les mises à jour des informations de configuration importantes.



HDFS – Ecriture



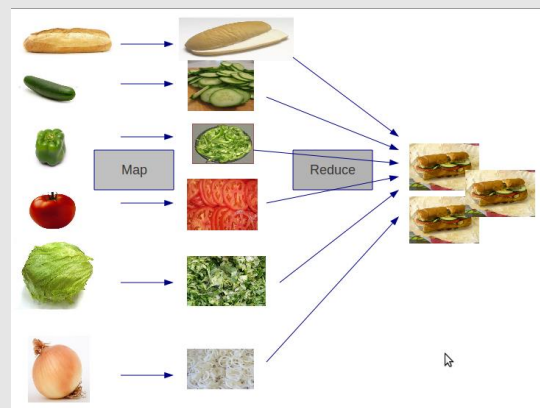
HDFS – Lecture



Important : la lecture des blocs peut se faire sur des réplicas

MapReduce – Intuition

- C'est un framework de traitements parallélisés
- Décompose des gros traitements en un ensemble de traitements plus petits qui produisent des sous résultats -> **Map**
- Agrège l'ensemble des résultats -> **Reduce**
- Les tâches sont exécutées par des **Workers**

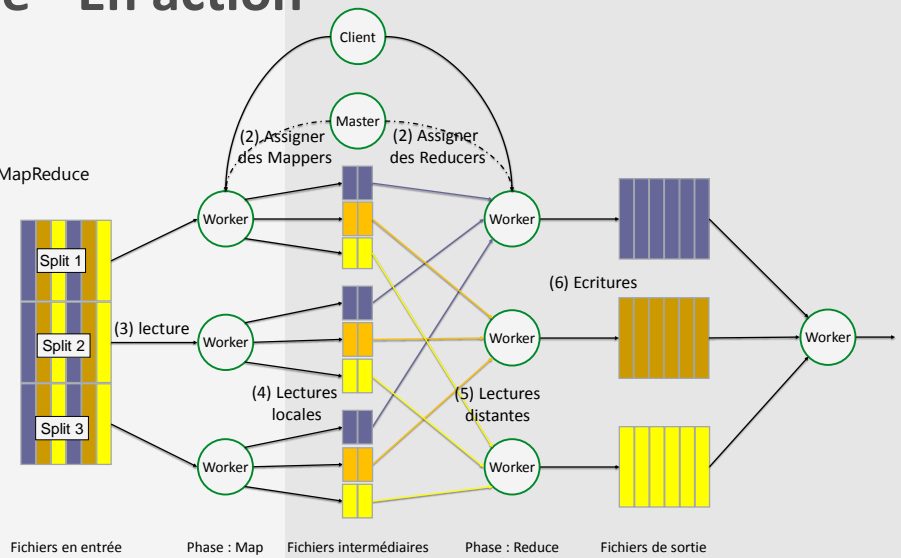


MapReduce - En action

Workers : Traitement des tâches

Master : Orchestrateur de tâches

Client : Lanceur des traitements MapReduce



MapReduce - En action