17 HADOOP

1.1 Définition d'Hadoop

• **Hadoop :** framework libre et open source écrit en Java qui facilite la création d'applications distribuées et échelonnables pour le stockage et le traitement de gros volumes de données

• Hadoop est apparu pour résoudre les problèmes des 3V (Volume, Vélocité, Variété) des données. Il permet aux applications de travailler avec des milliers de nœuds et des pétaoctets de données.

1.2 Avantages d'Hadoop

• **Gestion de gros volumes de données :** Hadoop peut traiter des quantités massives de données en les fractionnant en blocs et en les distribuant sur des nœuds.

• **Efficacité de stockage :** Hadoop utilise le système de stockage distribué HDFS pour stocker les données de manière redondante et fiable.

• **Bonne capacité de récupération de données :** En cas de défaillance d'un nœud, les données sont automatiquement répliquées sur d'autres nœuds.

• **Évolutivité horizontale :** Hadoop permet d'ajouter facilement de nouveaux nœuds au cluster pour augmenter la capacité de traitement.

• **Moindre cout :** Hadoop s'exécute sur du matériel standard et utilise des logiciels open source, ce qui réduit les couts par rapport aux solutions propriétaires.

1.3 Architecture d'Hadoop

• Hadoop est composé de plusieurs modules :

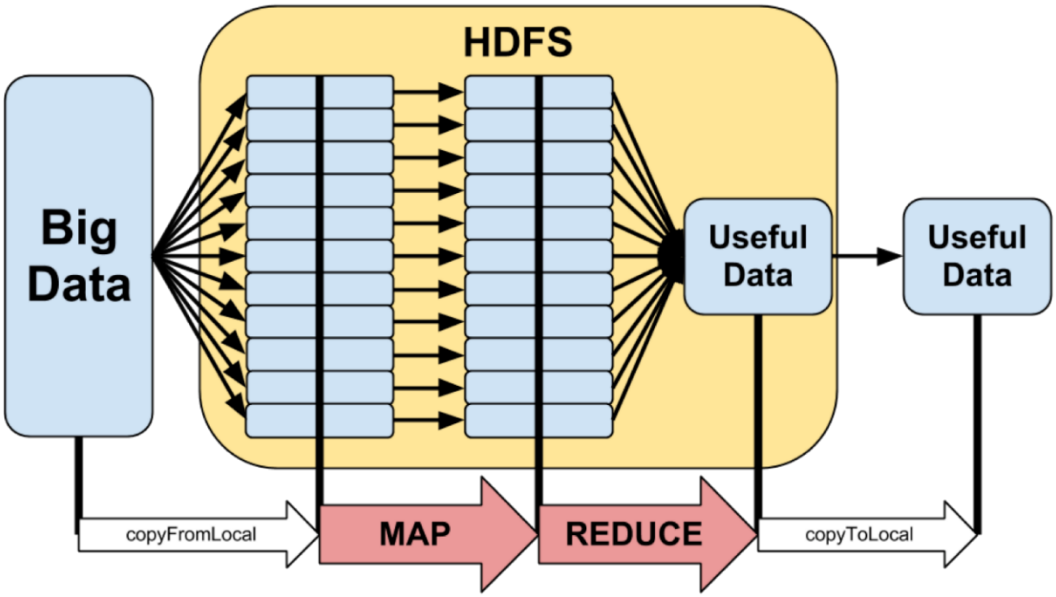
— **Hadoop YARN :** gestionnaire de ressources qui surveille les ressources disponibles dans le cluster (il peut y avoir plusieurs gestionnaires de ressources)

— **Hadoop MapReduce :** moteur de traitement de données intégré à Hadoop, responsable de la gestion des fichiers et du traitement distribué

— **Hadoop Common :** ensemble de fonctionnalités pour l'administration et la planification du système

— **Hadoop Distributed File System (HDFS) :** système de stockage distribué pour les données d'Hadoop

• Chacun des éléments est remplaçable. Ex : on peut utiliser Spark à la place de MapReduce ou utiliser NoSQL à la place de HDFS.



• Hadoop fractionne les fichiers en gros blocs et les distribue à travers les nœuds du cluster. Pour traiter les données, il transfère le code à chaque nœud et chaque nœud traite les données dont il dispose. Cela permet de traiter l'ensemble des données plus rapidement et plus efficacement que dans une architecture supercalculateur plus classique.

1.4 Outils basés sur Hadoop

• **MapReduce :** outil de mise en œuvre du paradigme de programmation parallèle du même nom

• **HBase :** base de données distribuée avec stockage structuré pour les grandes tables

• **Hive :** logiciel d'analyse de données permettant d'utiliser Hadoop avec une syntaxe proche du SQL, initialement développé par Facebook

• **Pig :** logiciel d'analyse de données utilisant le langage Pig Latin, initialement développé par Yahoo!

• **Spark :** framework de traitement de données distribué avec mémoire partagée, compatible avec Hadoop : il permet d'envoyer du code dans Hadoop pour qu'il soit interpréter directement en MapReduce

2 COMMANDES SHELL COURANTES POUR HDFS

• **Créer un répertoire :** hadoop fs -mkdir -p nom_dossier

• **Lister le contenu du répertoire où on est :** hadoop fs -ls

• **Lister le contenu d'un répertoire :** hadoop fs -ls répertoire

• **Copier un fichier du système local dans HDFS :** hadoop fs -put chemin/fichier/source chemin/fichier/destination

• **Copier un fichier depuis HDFS vers le système local :** hadoop fs -get chemin/fichier/source chemin/fichier/destination

• **Afficher les premières lignes d'un fichier :** hadoop fs -cat fichier | head

• **Afficher les dernières lignes d'un fichier :** hadoop fs -tail fichier

• **Renommer un fichier ou le déplacer vers un nouvel environnement :** hadoop fs -mv ancien_nom nouveau_nom

• **Supprimer un fichier du système HDFS :** hadoop fs -rm fichier

3 AUTRES COMMANDES

• Pour exécuter des programmes MapReduce ou Spark ; on peut utiliser la commande « hadoop jar » ou « spark-shell ». Exemple :

— **Exécuter un programme Java (fichier JAR) dans le cluster Hadoop avec les arguments spécifiés :** hadoop jar fichier.jar programme chemin/dossier/source chemin/dossier/destination

— **Lancer l'interpréteur de commandes Spark pour exécuter des instructions Spark en mode interactif :** spark-shell