Hadoop

Hadoop est un framework java Open Source destiné à faciliter la création d'application distribuées et échelonnables, permettant aux applications de travailler avec des milliers de nœuds et des pétaoctets de données. Hadoop a été créé par Doug Cutting et fait partie des projets de la fondation logicielle Apache depuis 2009. Ce framework a été inspiré par les publications MapReduce, GoogleFS et BigTable de Google.

Hadoop Distributed File System (HDFS) :

Le HDFS est un système de fichiers distribué, extensible et portable développé par Hadoop à partir de GoogleFS. Il est écrit en Java, et conçu pour stocker de très gros volumes de données sur un grand nombre de machines. Il permet l'abstraction de l'architecture physique de stockage, pour manipuler un système de fichiers distribué comme s'il s'agissait d'un disque dur unique.
Son architecture machine repose sur 2 composants majeurs :

- NameNode (noeud de noms) : gère l'espace de noms, l'arborescence du système de fichiers et les métadonnées des fichiers et des répertoires. Il est unique mais dispose d'une instance secondaire qui gère l'historique des modifications dans le système de fichiers, un rôle de backup. Ce NameNode secondaire assure la continuité du fonctionnement du cluster Hadoop en cas de panne du NameNode d'origine.
- DataNode (nœud de données) : stocke et restitue les blocs de données. Lors du processus de lecture d'un fichier, le NameNode est interrogé pour localiser l'ensemble des blocs de données. Pour chacun d'entre eux, le NameNode renvoie l'adresse du DataNode le plus accessible, c'est-à-dire le DataNode qui dispose de la plus grande bande passante . Les DataNode communiquent de manière périodique au NameNode la liste des blocs de données qu'ils hébergent.

Hadoop dispose d'une implémentation complète de l'algorithme de MapReduce.

Fonctionnement :

Hadoop va tout d'abord split les données, c'est-à-dire les segmentées sous forme de bloc de 64Mo ou plus. Ensuite il va transférer sa fonction map sur chaque nœud/bloc de données et l'appliquer ; le fonction map aura été développée en fonction des besoins.
Une fois que toutes les tâches map auront été réalisé sur tous les nœuds, on va passer à l'application des méthodes reduce. L'entrée d'une fonction reduce correspond généralement à la sortie d'une fonction map.
Il est possible d'appliquer une fonction Combiner en sortie de la fonction map pour ne pas enregistrer les résultats et surtout réaliser une succession de 2 fonctions : une fonction shuffle et une fonction sort. Shuffle se charge de regrouper tous les couples possédant la même clé et la fonction sort elle va trier les clés. Ces deux fonction sont déjà implémenter par le framework Hadoop.

L'un des avantages de Hadoop est que tout d'abord en se servant du patron d'architecture MapReduce on peut réaliser des applications réalisant massivement des tâches en parallèles. De plus, ce framework simplifie en grande partie le travail. Tout d'abord car on a pas besoin de gérer les enregistrements. En effet, le développeur doit juste écrire le code du mapper et reducer pour traiter un seul enregistrement et Hadoop se chargera de passer d'un enregistrement à un autre.

Les utilisations possible de ce framework :
Grâce à Hadoop, il est possible de mettre en place des programmes comptant le nombre d'occurrence d'une thématique ou d'une valeur, ou encore chercher le maximum ou le minimum. Cependant, il est plus difficile de réaliser une application de calcul de la moyenne de données.

Il est notamment utilisé par par des entreprises comme Facebook, Yahoo ou encore Microsoft.
Cependant, bien que Hadoop soit le framework le plus connu il possède certains inconvénients qui réduisent ses performances notamment en milieu hétérogène.
Pour palier à ces soucis, l'Apache Software Foundation a réalisé un certains nombre de projet visant à améliorer Hadoop.

Notamment Hbase qui est une base de données distribuées disposant d'un stockage structuré pour les grandes tables. Ce système repose sur des couples clé/valeur appelé aussi stockage binaire.
Il y a également Hive qui met en place une Base de Données relationnel à l'intérieur du HDFS pour permettre de réaliser des requêtes en utilisant un langage proche du Sql nommé le HiveSql traduites ensuite comme des programmes MapReduce. Ce projet permet donc de mettre en œuvre un langage de requête connu par les développeurs.
Tout comme Hive, Pig propose un langage de haut niveau pour s'adresser aux données. Il s'adresse plus aux développeurs habitués à faire des scripts Bash ou Python.
Sqoop permet de dialoguer avec la base de données relationnel, d'importer ou d'exporter des données vers cette base. Ou encore Mahout, qui permet de mettre en place un système d'algorithme décisionnel. Notamment du positionnement de données ou encore de la classification automatique.

Hadoop est distribuée par quatre acteurs qui proposent des services de formation et un support commercial, en ajoutant des fonctions supplémentaires :

- Cloudera, première distribution historique d'Hadoop intégrant des packages classiques et certains développement propriétaires.
- HortonWorks
- MapR Technologies a développé un système de fichier pour Hadoop palliant les limites du HDFS. A également réalisé des technologies permettant la suppression du NameNode qui est un point de contentions dans l'architecture d'Hadoop .
- IBM BigInsights for Hadoop, qui est une distribution 100 % Open Source Apache Hadoop, proposant des extensions analytiques et d'intégration dans le SI d'entreprise.
-

Il existe cependant bien d'autres framework pouvant servir d'alternative à Hadoop :

BlobSeer, propose une solution à l'un des problèmes de Hadoop. En effet, des erreurs peuvent survenir lorsque plusieurs processus accèdent à un même fichier et BlobSeer propose une solution à ce problème en modifiant le HDFS pour le remplacer par son propre système de fichier le BSFS (BlobSeer File System).L'un des points forts de Hadoop est de parcourir des pétaoctets de données en quelques heures. Cependant, cela n'est possible que parce que les fichiers ont une taille de plusieurs centaines de gigaoctets. De ce fait, il devient possible d'accéder à de petites parties d'un même fichier de façon concurrente. Il est cependant impossible de travailler avec des millions de petits fichiers au lieu d'un seul. BlobSeer propose donc un système de fichier permettant d'accéder à des petites parties d'un grand fichier, pour que des milliers de « client » puissent modifier le même fichier sans problème de conflit. BSFS permet également le « versioning » permettant de supprimer les changements indésirable et la création de branche indépendantes.

Phoenix est une API basé sur le modèle MapReduce, proposé par Google. La grande différence avec Hadoop est que Phoenix est utilisé sur des ordinateurs multi cœurs et donc il n'utilise pas des serveurs mais des threads pour utiliser MapReduce. Phoenix est donc basé sur un langage fonctionnel pour rendre la parallélisation totalement transparente ; utilisé en C et en C++. En ce qui concerne ces performances, les résultats de Phoenix montre qu'avec un processeur 4 cœurs on peut accélérer les calculs de 40 % et avec un 8 cœur de 50 %. Cependant, bien que la vitesse de calcul soit augmentée, sur des machines simple elle reste équivalente. L'implémentation du modèle n'est pas assez générale pour couvrir la totalité des programmes.

Mars est un framework pour implémenter le modèle MapReduce sur des processeurs graphiques car ceux-ci ont dix fois plus de mémoire que les CPU et sont aussi 10 fois plus rapide. En ce qui concerne les performances de Mars, après avoir été comparé avec Phoenix, les performances de Mars sont 1,5 fois plus élevé que celles de Phoenix. Mars n'est cependant pas capable de s'occuper de trop grande données, bloqué par la taille de la mémoire GPU.

PQL serait plutôt une alternative à Hadoop. Il s'agit d'un langage implémenté comme une extension de java, pour augmenter son expressivité et sa modularité. Il s'agit d'un langage de programmation déclarative, ce qui veut dire que le développeur peut spécifier ce qu'il veut faire sans pour autant préciser comment le faire. Pql a pour objectif de trouver la méthode la plus rapide à exécuter. Il saura notamment quels morceaux du code sera parallélisé ou non. PQL a été optimisé pour les tâches parallèles de façon que l'utilisateur ait le moins de manipulations à faire comparé a un code parallèle écrit par l'utilisateur. Suite à des tests de performances, PQL a démontré qu'il est bien plus performant et qu'il utilise moins de ligne de code. Cependant, il faut rappeler que SQL et Hadoop utilisent une table de données et donc que le temps d'exécution est ralenti par l'accès à la base. De plus, Hadoop est optimisé pour des calculs sur des clusters et non sur une seule machine.

framework écologique ce framework est placé dans une situation d'économie des énergies du MapReduce. La plupart des framework présenté précédemment considèrent que le traitement se fait dans un milieu homogène, en réalité il est rare qu'un cluster contienne les mêmes machines.Ce framework a donc été conçu pour pouvoir planifier de façon dynamique les tâches en fonction de la consommation individuelle de chaque nœud du clusters.