Centralistion/Stockage des données

Le Big Data, pour recueillir toutes les données, utilise divers processus de centralisation et de stockage des données. Comment a lieu le transfert des données jusqu'au Big Data ? Dans un premier temps, nous allons étudier le fonctionnement de MapReduce, puis dans un second temps nous allons voir comment agit le cadriciel Hadoop.

Tout d'abord, MapReduce est un modèle d'architecture de développement informatique, qui effectue des calculs parallèles et souvent distribués sur des données pouvant être très volumineuses. Il repose sur deux fonctions : « Map » et « Reduce », empruntées aux langages de programmation fonctionnelle. De façon générale, la fonction Map, exécutée par un nœud spécifique, analyse un problème, le découpe en sous-problèmes, et ensuite délègue la résolution de ces sous-problèmes à d'autres nœuds de traitements pour être traités en parallèle, ceci à l'aide de la fonction Reduce. Ces nœuds font ensuite remonter leurs résultats au nœud qui les avait sollicités (équivalent du multitâche).

Ainsi le modèle MapReduce permet de manipuler de grandes quantités de données en les distribuant dans un cluster de machines pour être traitées. Notons que MapReduce est aussi de plus en plus utilisé dans le Cloud Computing. De nombreux cadriciels implémentant MapReduce ont vu le jour, dont le plus connu est Hadoop.

Hadoop est un cadriciel de référence libre et open source, intégrant MapReduce et permettant d'analyser, stocker et manipuler de très grandes quantités de données.

Le noyau d'Hadoop est constitué d'une partie stockage consistant en un système de fichiers distribué, extensible et portable appelé HDFS, et d'une partie traitement appelée MapReduce.

Hadoop fractionne les fichiers en gros blocs et les distribue à travers les nœuds du cluster. Pour traiter les données selon le modèle MapReduce, il transfère le code à chaque nœud et chaque nœud traite les données dont il dispose. Cela permet de traiter un volume important de données plus rapidement et plus efficacement que dans une architecture super-calculateur classique.

Hadoop est généralement utilisé par les bases de données NoSQL liées à la gestion et au stockage des mégadonnées. Mais de nouveaux cadriciels spécifiques améliorent les performances d'Hadoop, que se soit en termes de vitesse ou de consommation électrique même en milieu hétérogène.

Afin de stocker des mégadonnées il faut mettre en place un partitionnement. Mais pour gérer les

mégadonnées partitionnées un système incapable d'assurer la cohérence et la disponibilité des données simultanément est forcément utilisé. Sinon pour assurer ces deux besoins il faut utiliser un système relationnel mais il faudra alors faire un choix entre la cohérence et la disponibilité. En effet les systèmes relationnels demandent des données structurées or, les mégadonnées sont généralement peu voir pas du tout structurées.

Pour palier à ces problèmes, de nouveaux modèles de stockage ont été créés et on mis en avant les base de données NoSQL. Ces bases de données ne sont pas un substitut des bases de données classiques mais un complément étant plus intéressant selon les besoins.

Les systèmes NoSQL permettent une plus grande performance avec les applications Web qui ont une quantité de données exponentielles. Ces systèmes utilisent la technique du sharding et du consistent hashing en plus de MapReduce (vu précédemment).

Glossaire

- **Cadriciel :** Mot valise formé à partir de deux mots, cadre et logiciel.
- **Hadoop :** « High-Availability Distributed Object-Oriented Plat-form » a été créé par Doug Cutting et fait partie des projets de la fondation logicielle Apache depuis 2009.
- **HDFS :** « Hadoop Distributed File System »
- **Cluster :** Dans un système informatique, un agrégat, ou « cluster », est un groupe de ressources, telles que des serveurs.
- **NoSQL :** « Not Only SQL »
- **Sharding :** très forte distribution des données et des traitements associés sur de nombreux serveurs
- **Consistent Hashing :** partitionnement horizontal des données sur plusieurs nœuds ou serveurs

Lien(s) :

http://www.lsis.org/espinasseb/Supports/BD/Article-BigData-TI-2016.pdf