

4. Gestion des données urbaines dans les nuages informatiques

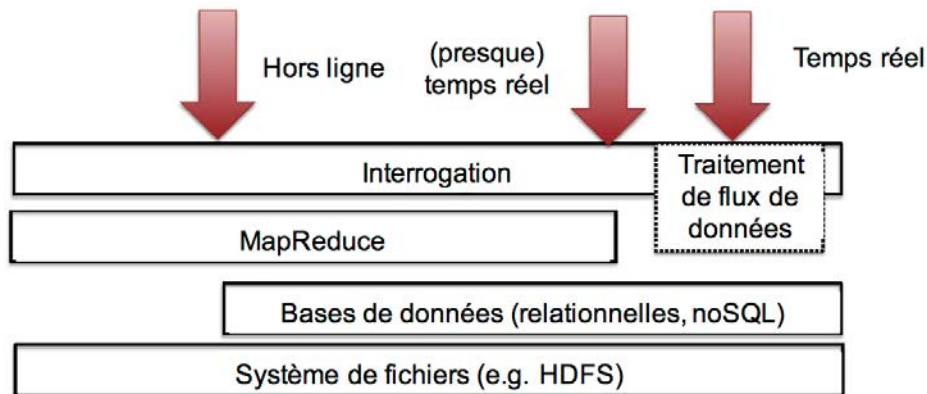
- Brève histoire des nuages informatiques
- Modèles de service et de déploiement
- Technologie clé : la virtualisation
- IaaS : les points de vue utilisateur et fournisseur
- PaaS : programmation et déploiement des applications
- Stockage de données
- **Traitement de données**
- Traitement de flux de données

Christine Morin

VILLES INTELLIGENTES : DÉFIS TECHNOLOGIQUES ET SOCIÉTAUX

Dans cette séquence et la suivante, nous aborderons les deux principaux types de traitement de données dans le Cloud, ...

Architecture du système en fonction du type d'analyse



3

L'architecture du système de traitement de données dépend des types d'analyses à réaliser. Donc les données peuvent être stockées dans un système de fichier distribué, comme HDFS, ou dans une base de données relationnelle ou noSQL, deux notions qui seront revues ultérieurement.

Selon qu'il s'agit d'un traitement de données en mode batch ou d'une analyse de flux de données en temps quasi réel, on va utiliser différents types d'environnements.

Donc typiquement pour le mode batch ça va être des environnements Mapreduce, et pour le traitement des flux de données ce seront des environnements spécifiques pour ce type de traitement.

Différentes interfaces d'interrogations peuvent être proposées aux utilisateurs selon les types de requêtes, donc les types d'application.

MapReduce : un modèle de programmation pour le traitement de données

- **Programmation fonctionnelle**
 - Deux fonctions : Map et Reduce
- **Modèle maître-esclave**
 - Données traitées en parallèle par différentes tâches
 - Chaque tâche traite un sous-ensemble des données

4

Et dans cette séquence nous allons nous pencher sur l'environnement Mapreduce, qui est le plus utilisé pour les traitements de données en mode batch.

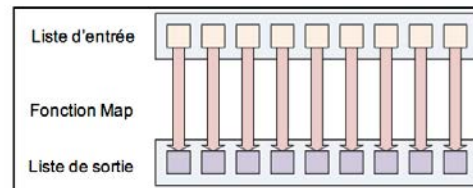
Le modèle de programmation Mapreduce est issu de la **programmation fonctionnelle** et a été remis au goût du jour par Google, dans la première moitié des années 2000. Un traitement était exprimé par **deux fonctions: map et reduce**.

La **mise en œuvre** se fait selon un **modèle Maître-Esclave** qui gère l'**exécution en parallèle des tâches**. Chaque tâche étant en charge du traitement d'une partie des données à traiter. Une tâche exécute soit la fonction Map soit la fonction Reduce, le nœud maître distribue les tâches à exécuter sur des nœuds esclaves.

Un exemple : comptage de mots dans un texte

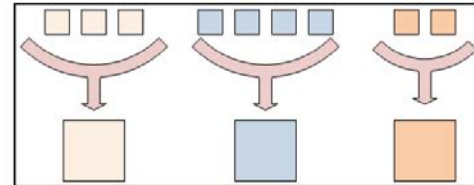
Map (fichier)

pour chaque ligne du fichier :
pour chaque mot de la ligne :
émettre (mot, 1) ;



Reduce (mot, valeurs)

somme = 0 ;
pour chaque valeur dans valeurs :
somme = somme + valeur ;
émettre (mot, somme) ;



5

Prenons l'exemple d'une **application toute simple, qui compte le nombre d'occurrences de chaque mot dans un texte.**

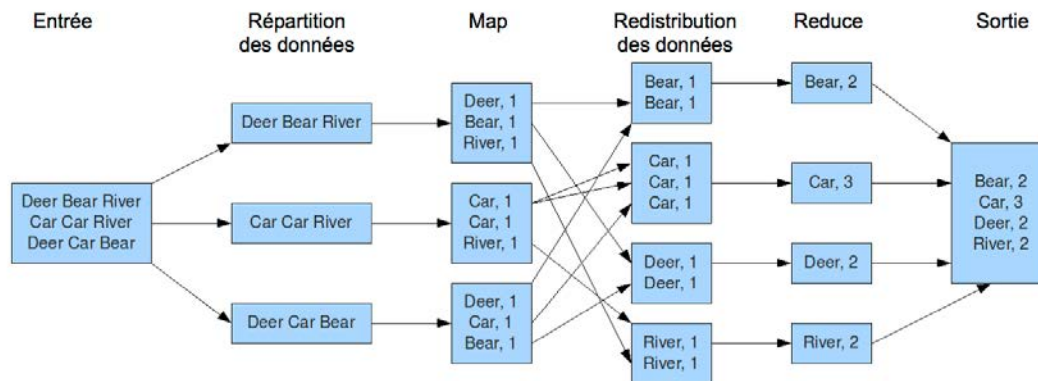
Donc la fonction Map est appliquée à un texte passé en paramètres d'entrée, donc pour chaque mot du texte le traitement consiste à émettre une paire clé-valeur : la clé est un mot trouvé dans le texte et la valeur étant un.

La fonction Reduce reçoit en entrée une liste de paires clé-valeur et fait la somme des valeurs pour une clé donnée.

Le résultat produit est le nombre d'occurrences pour chaque mot.

Un exemple d'application MapReduce

MapReduce – Application de comptage du nombre d'occurrences des mots d'un texte



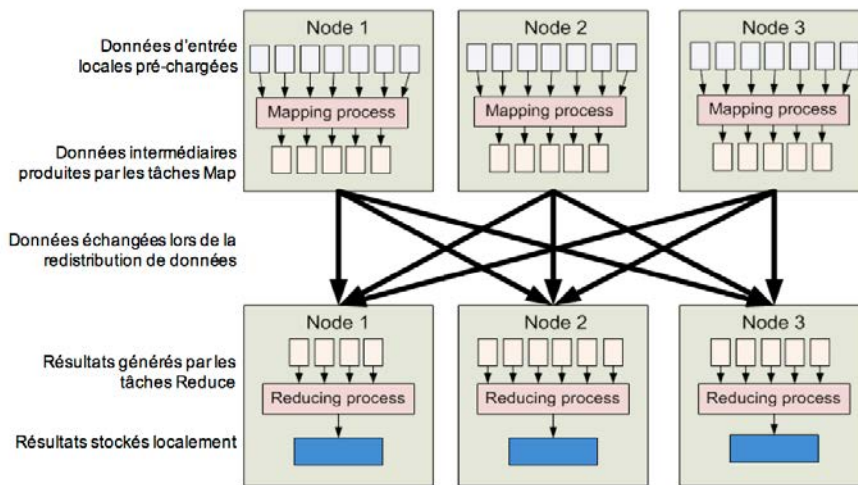
6

Le schéma ci-après montre la structure de l'application de comptage de mots, sur l'exemple d'un petit texte en langue anglaise.

Le fichier d'entrée est découpé en blocs de taille fixe, chaque tâche Map reçoit un bloc à traiter, les résultats intermédiaires donc les couples Mots-Valeurs sont transmis aux tâches Reduce, qui calculent alors la somme des valeurs.

Et les résultats des tâches Reduce sont ensuite regroupés pour fournir le résultat final en sortie.

Environnement d'exécution MapReduce



7

Nous présentons ici l'exécution de l'application pour en voir les différents flux de données.

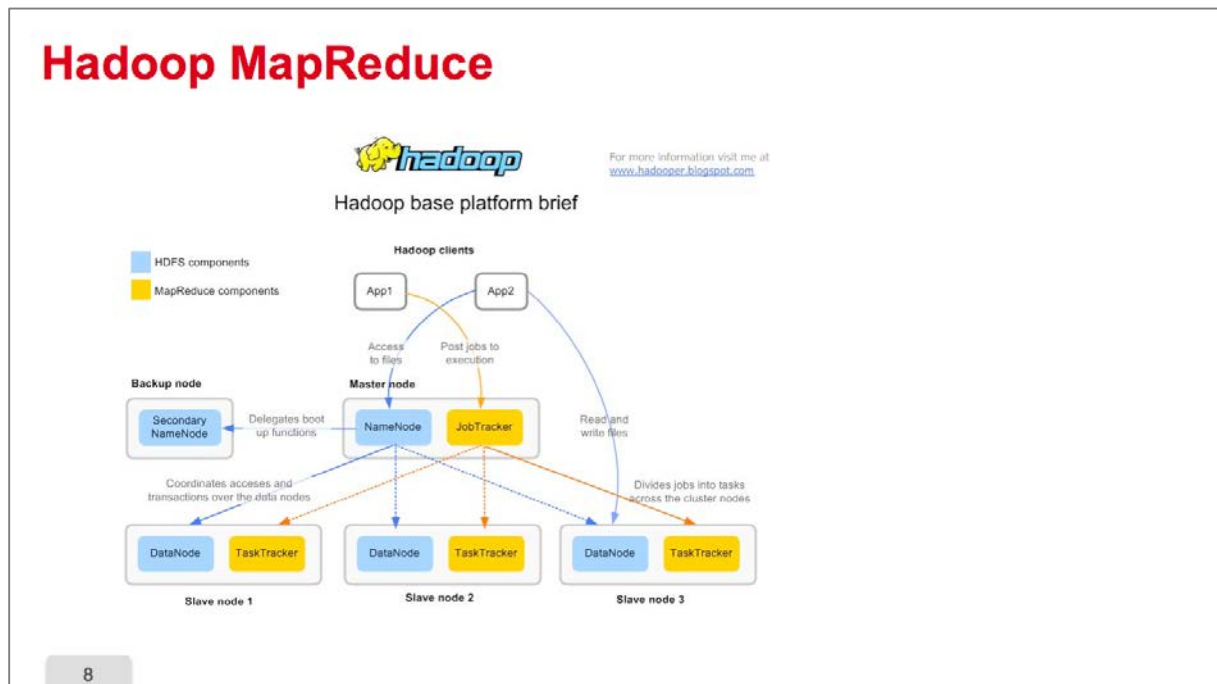
Donc les tâches Map et Reduce sont exécutées par différents nœuds, donc les données en entrée doivent tout d'abord être acheminées depuis un système de stockage dans lequel elles se trouvent, donc ça peut être le système S 3 ou un stockage externe. Elles doivent donc être acheminées sur les nœuds d'exécution des tâches Map et Reduce dans des disques locaux.

Une fois que les tâches Map ont terminé leur traitement, les données sont distribuées aux tâches Reduce, qui peuvent s'exécuter sur des nœuds différents.

Donc c'est la phase de redistribution de données, de Shuffle, et ensuite les tâches Reduce s'exécutent en parallèle et produisent localement sur les disques locaux des nœuds sur lesquels elles exécutent leurs résultats.

Ces résultats peuvent ensuite être transférés vers un stockage soit S 3 dans le Cloud, ou même vers l'extérieur vers l'utilisateur.

Hadoop MapReduce



Un exemple d'exécution de traitement de données en mode batch est l'environnement Open source à Hadoop Mapreduce. C'est celui qui est le plus utilisé actuellement.

Il peut être exécuté sur des Clusters physiques ou bien dans le Cloud au-dessus de machine virtuelle.

Cet environnement s'appuie sur le système de fichiers distribués HDFS, qui est représenté en bleu sur le schéma, et l'environnement Map Reduce qui est représenté en jaune sur le schéma.

HDFS implémente deux types de services : Le Namenode, qui gère le placement des données sur les Data Node qui les stocke.

L'environnement Mapreduce quant à lui possède également deux types de services: le Job Tracker qui est le nœud maître de l'environnement et qui va distribuer les tâches à exécuter sur les autres nœuds esclaves, donc ces tâches à exécuter vont être gérées par les Task Trackers

Les données sont répliquées par HDFS potentiellement sur plusieurs nœuds, et du coup le Job Tracker va chercher à exécuter une tâche de préférence sur les nœuds qui stockent les données qu'elle doit traiter. Donc pour se faire le Job Tracker implémente des algorithmes d'ordonnancement de tâches.

Donc plusieurs applications Mapreduce peuvent s'exécuter simultanément sur un même Cluster Hadoop.

De nombreux paramètres qui impactent les performances

- Caractéristiques des nœuds (nombre de cœurs, capacité mémoire, taille disque)
- Latence et débit réseau
- Taille des blocs
- Degré de réplication des blocs
- Nombre de tâches Map et Reduce par nœud

9

L'environnement Mapreduce comporte de nombreux paramètres dont les valeurs vont impacter les performances de l'application. Donc il faut choisir ces paramètres judicieusement. Donc on peut citer :

- les caractéristiques du Cluster sous-jacent, donc la taille du Cluster, la capacité des nœuds, le nombre de cœurs, la taille mémoire, la taille disque
- les caractéristiques du réseau, en termes de débit et de latence.

Sont à prendre en compte également les paramètres de HDFS, le système de fichiers, donc la taille des blocs, le nombre de répliquas par exemple.

Et puis les paramètres de l'environnement Mapreduce, à savoir le nombre de tâches MAP et Reduce exécutées sur chaque nœud en parallèle.

Hadoop MapReduce dans le cloud

- **Choix des caractéristiques des machines virtuelles du cluster Hadoop**
 - Capacité disque importante
- **Impact du placement des machines virtuelles sur les performances et la disponibilité des données**
 - Exécution de plusieurs VMs sur une même machine physique
- **Lourdeur de la migration des machines virtuelles contenant les nœuds de données HDFS**
- **Elasticité du cluster Hadoop MapReduce**
 - Exemple : Amazon EMR, Sahara, Resilin

10

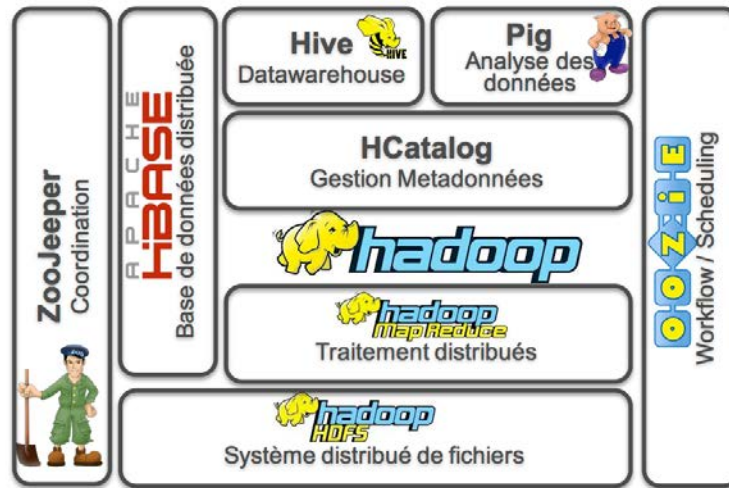
Lorsqu'on exécute Mapreduce dans le Cloud, la situation est encore plus complexe. Outre le choix d'une solution à base de machine virtuelle ou bien à base de conteneurs, ce qui peut avoir un impact sur les performances, il s'agit de **choisir aussi le type des machines virtuelles** donc en particulier la taille disque, mais aussi le nombre de cœurs.

Certains algorithmes de gestion de ressources peuvent être amenés à déplacer des machines virtuelles en cours d'exécution. Dans le cas d'environnements Hadoop et en particulier HDFS, déployés dans le Cloud, il faut prendre garde aux machines virtuelles qui hébergent les Data Node. Car migrer une machine virtuelle qui comporte beaucoup de données est une opération lourde qui va impacter les performances.

Enfin des environnements Mapreduce élastiques ont été développés pour le Cloud. On peut citer comme exemple Elastic Map Reduce, qui permet de déployer facilement un environnement Map Reduce sur un Cloud IaaS Amazon, EC2. Sahara est un logiciel en open source qui offre le même type de fonctionnalité sur un Cloud IaaS Openstack. Et enfin Resilin est un système open source qui permet de déployer une application Mapreduce sur n'importe quel Cloud offrant une interface EC2, et permet à une application d'exploiter les ressources de plusieurs Clouds simultanément.

Tous ces environnements ont une interface qui permet de gérer l'élasticité du Cluster Hadoop, tirant profit des capacités des Clouds IaaS qui offrent des ressources à la demande. L'écosystème Apache Hadoop Mapreduce est en constante évolution et s'enrichit en fonction des besoins.

Hadoop : un écosystème en constante évolution



11

Indéniablement c'est un écosystème de choix pour traiter les données en mode batch, dans le contexte de la ville intelligente, qu'il s'agisse des données ouvertes mises à disposition du public par les collectivités territoriales, ou des données d'historiques qui sont possédées par les opérateurs des différentes infrastructures de la ville, ou par les fournisseurs d'applications mobiles à destination des citoyens.

Illustrations & photos : crédits

p. 2 : © Julien Eichinger, Fotolia ; By Camelia.boban CC BY-SA 3.0, via Wikimedia Commons ; Domaine public, Pixabay.

p. 3 : d'après Lavanya Ramakrishnan, *Cloud Computing and Big Data : Overview*, LBNL, 2012 workshop Discovery Workshops covering High Performance Computing, Cloud Computing and Big Data topics.

p. 5 : "Hadoop Tutorial from Yahoo!" by [Yahoo! Inc.](#) – CC BY - <https://developer.yahoo.com/hadoop/tutorial/module4.html#basics>

p. 6 : <http://www.rabidgremlin.com/data20>

p. 7 : "Hadoop Tutorial from Yahoo!" by [Yahoo! Inc.](#) – CC BY - <https://developer.yahoo.com/hadoop/tutorial/module4.html#basics>

p. 8 : droits réservés