

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

UNIVERSITE BADJI MOKHTAR ANNABA
FACULTE DES SCIENCES DE L'INGENIORAT
DEPARTEMENT INFORMATIQUE

HADOOP MAPREDUCE

Master : Gestion et Analyse des Données Massives (GADM)

2^{me} année

Dr. Klai Sihem

AVANT PROPOS. . .

Le Big Data est une science récente qui a surgie avec l'évolution et la variation des données et d'applications mises et échangées en ligne. Cette science consiste à prendre en charge d'une manière efficace un volume important des données hétérogènes en intégrant des techniques et outils nouveaux, vu que la technologie disponible ne répond plus aux besoins.

Ce polycopié est un support pédagogique qui permet d'initier l'étudiant au domaine des Big Data. Ce cours composé de plusieurs chapitres permet aux étudiants de comprendre la problématique et la motivation du domaine, et de maîtriser l'outil Hadoop avec le modèle MapReduce associé à ce domaine.

Chaque chapitre est élaboré pour répondre à un but pédagogique bien précis, se matérialisant par des explications, définitions accompagnées d'exemples et des illustrations par des figures suivies par des exercices, des solutions envisageables ou des fiches de travaux pratiques bien guidés.

1. Le chapitre I met l'étudiant dans le contexte du Big Data, consiste à lui donner des connaissances générales sur le domaine ;
2. Le chapitre II est consacré à l'étude de Hadoop, le framework qui permet le développement d'applications traitant les données massives. Ce chapitre donne les notions les plus générales avec la procédure d'installation du logiciel ;
3. Le chapitre III détaille la partie qui s'occupe du stockage des données "HDFS", avec la possibilité de la manipulation de ces données selon deux manières différentes à savoir : les commandes et l'API JAVA ;

4. Le chapitre IV étudie en détail la partie traitement des données massives "MapReduce", le modèle qui permet de traiter des blocs de données séparément et parallèlement dans des machines connectées. La modélisation selon le paradigme MapReduce est une étape importante avant le développement des programmes ;
5. Le chapitre V détaille l'implémentation des programmes MapReduce dans Hadoop. Dans le cadre de ce chapitre, nous étudions l'implémentation des programmes en utilisant le langage Java. D'autres langages peuvent être utilisés pour écrire des programmes mapreduce, mais cette partie n'est pas traitée dans ce cours.

L'élaboration de ce polycopié a été inspirée de plusieurs documents, j'ai pris le soin de les citer dans la partie bibliographie, d'autres documents aussi ont été cités afin d'apporter aux lecteurs plus de commandes et plus de détails sur les parties traitées dans ce cours.

TERMINOLOGIE. . .

JVM : Java virtuelle machine

HDFS : Hadoop Distributed File System

YARN : Yet Another Ressource Negotiator

AM : Application Master

API java : Application Programming Interfaces java

TABLE DES MATIÈRES

PRÉFACE	3
TABLE DES MATIÈRES	6
LISTE DES FIGURES	7
1 GÉNÉRALITÉS SUR LE BIG DATA	1
1.1 INTRODUCTION	2
1.2 DÉFINITION ET SPÉCIFICITÉ DU BIG DATA	2
1.3 PROBLÉMATIQUE ET MOTIVATION	4
1.4 EXIGENCE DE LA PLATEFORME BIG DATA	5
1.5 TECHNOLOGIES LIÉES AU BIG DATA	5
1.5.1 Système de gestion de fichier et mécanisme MapReduce de Google .	5
1.5.2 Hadoop	6
1.5.3 Bases de données NoSQL	6
1.5.4 Spark	8
A ANNEXES	9
BIBLIOGRAPHIE	11

LISTE DES FIGURES

GÉNÉRALITÉS SUR LE BIG DATA

1

Objectif Pédagogique

A la fin de ce chapitre, l'étudiant sera capable de :

1. Déterminer les problématiques qui ont amené à l'émergence du Big Data ;
2. Préciser sa définition, son principe et ses objectifs ;
3. Distinguer les techniques liées au Big Data.

Pré-requis

L'étudiant doit avoir :

- Des connaissances sur l'informatique générale; les systèmes de bases de données relationnelles.
- Des connaissances sur les applications Web, les réseaux sociaux.

1.1 INTRODUCTION

Dés l'année 1980, il y a eu une perception de l'évolution importante et massive des applications informatiques, des données manipulées et générées par ces applications ainsi que le volume de leur utilisation à travers les micro-ordinateurs, les téléphones portables et le Web.

Ces utilisations se matérialisent dans l'indépendance d'Internet pour des services ordinaires dans la vie courante, l'exploitation permanente des équipements équipés de GPS ou capteurs connectés ainsi que le partage de ces informations avec d'autres utilisateurs que ça soit par SMS, e-mail ou via les réseaux sociaux.

Tout cela nécessite la manipulation d'une quantité considérable de données. Les sources suivantes ^{1,2,3,4} donnent des exemples des tailles de données générées par des applications exploitées par les utilisateurs à travers le web ⁰⁶.

- 4 octets \cong un caractère au format utf-8;
- 1 kilooctet (Ko) = $10^3 \approx$ une page de texte;
- 1 mégaoctet (Mo) = $10^6 \approx$ vitesse de transfert par seconde;
- 1 gigaoctet (Go) = 10^9 et 1 Go \approx DVD, clé USB;
- 1 téraoctet (To) = $10^{12} \approx$ disque dur;
- 1 pétaoctet (Po) = 10^{15} et 4 Po \approx Données générées par Facebook, par jour;
- 1 exaoctet (Eo) = $10^{18} \approx$ Facebook et Amazon;
- 1 zétaoctet (Zo) = $10^{21} \approx$ Données générées par le web en 2020, par an.

1.2 DÉFINITION ET SPÉCIFICITÉ DU BIG DATA

Le Big Data, appelé aussi mégadonnées ou données massives désigne des ensembles de données volumineux et variés dont la prise en charge dépasse

-
1. wikipedia. https://en.Wikipedia.org/wiki/World_population
 2. facebook. <https://research.fb.com/facebook-s-top-open-data-problems/>
 3. ercim-news <https://ercim-news.ercim.eu/en89/special/managing-large-data-volumes-from-scientific-facilities>
 4. dellemc <https://www.dellemc.com/en-us/index.htm>

les capacités de traitement et d'analyse des outils informatiques existants de gestion de bases de données ou de l'information.

Inventé par les géants du Web, le big data peut être considéré comme une solution pour l'analyse et le traitement en temps réel des données volumineuses et non structurées [01](#).

La définition de big data varie selon les communautés qui s'y intéressent en tant qu'utilisateur ou fournisseur de service. L'expression « Big Data » date de 1997 selon l'Association for Computing Machinery. En 2001, l'analyste du cabinet Meta Group, Doug Laney décrivait les big data d'après le principe des « trois (3)V » :

- Le **Volume** désigne la taille des données générées qui ne cesse d'augmenter et nécessite de nouvelles manières pour leurs stockages et leurs traitements ;
- La **Vélocité** ou la vitesse à laquelle nous parvenons ces données implique de mettre en place des solutions de traitement en temps réel qui ne paralysent pas le reste de l'application ;
- La **Variété** de ces données, elles peuvent être non structurées, semi-structurées ou structurées. La technologie big data, permet de faire l'analyse, la comparaison, la reconnaissance, le classement des données de différents types comme des conversations, ou messages sur les réseaux sociaux, les photos, les vidéos, etc.

D'autres communautés et notamment les gestionnaires et les économistes décrivent big data selon cinq (5)V (Volume, Vélocité, Variété, Véracité, Valeur) [01](#).

- La **Véracité** évoque la nécessité de vérifier la crédibilité et la fiabilité des informations collectées notamment s'il s'agit d'abréviation, de langage familier, ect.
- La **Valeur** : il s'agit d'être capable de se concentrer sur les données qui peuvent être converties en réelles valeurs apportant un plus aux entreprises concernées.

Les données massives nécessitent des outils capables de fournir de nouvelles manières de traitement pour la capture, le stockage, la recherche, le traitement et le partage.

1.3 PROBLÉMATIQUE ET MOTIVATION

Google, Apple, Facebook, Amazon et d'autres plateformes existantes sur le Web sont ceux qui manipulent les données les plus volumineuses, les plus variées et qui nécessitent un temps réel pour le traitement de ces données, mais il faut noter qu'ils ne sont pas les seuls en mesure de faire du big data. Le problème de big data se pose quand les volumes des données sont très variées et dépassent la capacité des machines et des applications disponibles pour le stockage et le traitement de ces données en un temps acceptable.

Quelle est la solution à ce problème ? Est ce qu'il faut augmenter la capacité des machines et des applications ? Non, car on doit augmenter la capacité des machines et des applications pour chaque nouveau dépassement, ce qui est très coûteux et inconcevable.

La solution serait plutôt d'augmenter le nombre de machines, de découper les données en blocs, distribuer ces blocs à travers les machines et les traiter parallèlement sur cet ensemble de machines. Le principe de cette solution consiste à traiter les parties des données sur les machines où elles se trouvaient puis récupérer les résultats pré-agrégés, plutôt que de rassembler les données sur une unique machine.

La concrétisation de cette solution nécessite la réponse aux questions suivantes 06 :

- Comment distribuer le traitement entre les machines ?
- Comment distribuer les données entre les machines ?
- Comment agréger les résultats des différentes machines ?
- Que faire pour gérer les pannes des machines lors de l'exécution du traitement ?

1.4 EXIGENCE DE LA PLATEFORME BIG DATA

- Elle devrait être **scalable** afin de prendre en charge les évolutions des données ;
- Le traitement devrait être réparti et distribué **sur plusieurs machines** ;
- Elle devrait permettre de fournir une réponse rapide aux requêtes et de supporter une grande variété de sources de données ;
- Elle devrait être capable de gérer des données provenant d'environnement hétérogènes ;
- Elle devrait être capable de gérer les pannes ;
- etc.

1.5 TECHNOLOGIES LIÉES AU BIG DATA

La **programmation parallèle et distribuée** est une solution intéressante qui permet de traiter en temps raisonnable de gros volumes de données. Le développement d'une application distribuée est très complexe et cette complexité réside dans le stockage, l'échange et le transfert des données entre les machines ainsi que les problèmes posés dans le cas où une des machines tombe en panne, autrement dit, la tolérance aux pannes. Pour rendre transparente cette complexité au développeur, des plateformes distribuées et spécialisées dans le stockage et l'analyse de données ont émergé, comme **Hadoop** , qui est destinée à faciliter le stockage distribué des gros volumes de données et à supporter une chaîne de traitement de type **MapReduce** .

1.5.1 Système de gestion de fichier et mécanisme MapReduce de Google

Google a inventé en 2003 (symposium informatique de l'ACM) un système de fichiers distribués nommé Google File System (GFS) pour résoudre le problème de stockage et de gestion de très gros volumes de données. Puis, en 2004, Google présente son principe et sa réalisation de MapReduce, un mé-

canisme permettant de distribuer des traitements à appliquer sur les données distribuées à l'aide de GFS.

Le terme "MapReduce" s'inspire directement des fonctions map et reduce des langages fonctionnels. La fonction map prend deux arguments une fonction f et un ensemble de données E , et applique la fonction f à tous les éléments de l'ensemble E , et retourne une liste de résultats. La fonction reduce prend aussi deux arguments : une fonction g et la liste d'éléments retournés par la fonction map. La fonction g s'applique à cette liste et la réduit. Par exemple, on peut réduire une simple liste de valeurs numériques à sa somme, ou à son élément maximum, etc... Google réalise cette double opération Map Reduce à très large échelle à travers son système de fichier distribué GFS (voir chapitre ?? pour plus de détail).

1.5.2 Hadoop

Le système de fichiers distribué et son mécanisme de MapReduce distribué avaient fait une forte impression, mais ces outils restaient une propriété et exclusivité de Google et une implémentation libre a vu le jour peu de temps après.

Hadoop⁵ fut développé initialement par Doug Cutting pour mettre au point un moteur de recherche libre. Il repose sur le système de fichier distribué **HDFS** (Hadoop Distributed File System) , et un mécanisme de **MapReduce** distribué exploitant HDFS.

1.5.3 Bases de données NoSQL

A partir d'un système de fichier distribué à large échelle puis d'un mécanisme de MapReduce au dessus, les Bdd NoSQL ont pu voir le jour. Google avait d'abord développé BigTable en 2004 : une Bdd distribuée bâtie sur son système de fichiers distribués GFS et sur son propre mécanisme de MapReduce pour traiter les requêtes d'interrogation de données. Il s'agissait d'un

5. Hadoop. <https://www.apache.org/dyn/closer.cgi/hadoop/>

entrepôt de données orienté colonnes. Les données sont encore organisées en lignes et colonnes, mais toutes les lignes n'ont pas forcément les mêmes colonnes (modèle beaucoup plus souple que celui des Bdd relationnelles 02. Les Bdd NoSQL présentent des avantages par rapport aux Bdd relationnelles :

- Une grande souplesse dans la nature des données stockées ;
- L'exploitation d'un très gros volume dans des temps restant raisonnables, grâce notamment à un relâchement des contraintes d'intégrité ;
- L'augmentation des performances par la distribution massive du stockage et des traitements ;
- D'autres différences avec les Bdd relationnelles existent aussi sur les stratégies de contrôle des accès aux données (sécurisation des données), et de résistance aux pannes.

Il existe 5 grandes classes des Bdd NoSQL :

- Bdd stockant des paires clé-valeur. La plupart des applications demandent à lire des données à partir de leurs identifiants engendre le besoin de Bdd stockant des paires clé-valeur ;
- Bdd orientées document (MongoDB) où des clés sont associées à des documents à structure hiérarchique. Ceci permet la manipulation des documents web au format HTML-XML, ou document JSON ;
- Bdd orientées colonne (BigTable, HBase, Cassandra) : Ressemblent à des tables relationnelles, mais beaucoup plus souples. Les lignes peuvent avoir des colonnes différentes et en nombres différents. Les colonnes d'une ligne peuvent évoluer dynamiquement en nombre et en nom. Pas de champ « NULL » contrairement à une table relationnelle (pas de colonne inutile dans une ligne).
- Bdd réalisant un index inversé (Elasticsearch) : Un index inversé est indispensable pour traiter rapidement les requêtes de recherche de documents par mots clés. Mais l'index inversé peut devenir très volumineux (plus que les documents analysés) : il faut le compresser mais pas trop pour que la décompression à la volée soit rapide et/ou trou-

ver un format de compression permettant de travailler dans le format compressé. Les Bdd NoSQL spécialisées en index inversés apportent : des algorithmes optimisés pour construire ces index, des algorithmes de compression/décompression adaptés.

- Bdd orientées graphes (Neo4J) : Spécialement adaptées pour fouiller le web et les réseaux sociaux. Apportent des stockages de graphes efficaces (par références). Apportent des algorithmes d'analyse de graphes optimisés et adaptés au stockage réalisé. Les autres bases NoSQL pourraient stocker des graphes mais seraient moins efficaces ou inefficaces pour les analyses 02.

1.5.4 Spark

Spark est un projet Apache et aussi un outil de Big Data, il permet d'écrire des applications distribuées avec beaucoup de performance et rapidité. Il peut travailler sur des données sur disque ou des données chargées en RAM. Il est considéré comme le successeur de MapReduce, il a l'avantage aussi de fusionner une grande partie des éléments nécessaires dans Hadoop 01.

ANNEXES

A

BIBLIOGRAPHIE

- [01]. site : Le big data, 2019. URL <https://www.lebigdata.fr/definition-big-data>. (Cité pages 3 et 8.)
- [02]. Stephane Vialle, CentraleSuplec. Big Data : Informatique pour les donnees et calculs massifs. 24 mai 2017 . (Cité pages 7 et 8.)
- [03]. mohammed zuhair al taie. hadoop ecosystem : an integrated environment for big data. in big data, guest posts 2015. URL <http://blog.agroknow.com/?p=3810>.
- [04]. site : Hadoop. 2016. URL <http://www.opentuto.com/category/web-2/big-data/hadoop/>.
- [05]. amal abid. cours big data : Chapitre2 : Hadoop. 2017. URL <https://fr.slideshare.net/AmalAbid1/cours-big-data-chap2>.
- [06]. celine hudelot, regis behmo. realisez des calculs distribue sur des donnees massives. 2019. URL <https://openclassrooms.com/fr/courses/4297166-realisez-des-calculs-distribues-sur-des-donnees-massives>. (Cité pages 2 et 4.)
- [07]. Benaouda, Sid Ahmed Amine, implantation du modele mapreduce dans l'environnement distribue, 2015 . URL <http://dspace.univ-tlemcen.dz>.
- [08]. Tom White, Hadoop : The Definitive Guide, June 2009 : First Edition.
- [09]. Srinath Perera, Thilina Gunarathne, Hadoop MapReduce Cookbook, First published : February 2013.

- [10]. Pierre Nerzic, Outils Hadoop pour le BigData, mars 2018.
- [11]. marty hall. map reduce 2.0 (input and output). 2013. URL <https://www.slideshare.net/martyhall/hadoop-tutorial-mapreduce-part-4-input-and-output>.
- [12]. build projects, learn skills, get hired. hadoop mapreduce- java-based processing framework for big data. URL <https://www.dezyre.com/hadoop-course/mapreduce>.
- [13]. URL <https://hadoop.apache.org/docs/r2.4.1/api/org/apache/hadoop/mapreduce/>.
- [14]. URL <https://gist.github.com/kzk/712029/9d0833aac03b23ec226e034d98f5871d9580724e>.
- [15]. Jeffrey Dean and Sanjay Ghemawat, MapReduce : Simplified Data Processing on Large Clusters, 2004 .
- [16]. Univ. Lille1- Licence info 3eme annee, Cours n°1 : Introduction à la programmation fonctionnelle, 2013-2014 .
- [17] dataflair team. hadoop architecture in detail hdfs,yarn et mapreduce. 2019. URL <https://data-flair.training/blogs/hadoop-architecture/>.
- [18] vlad korolev. hadoop on windows with eclipse. 2008. URL <http://v-lad.org/Tutorials/Hadoop/>.
- [19]. Donald Miner and Adam Shook. MapReduce Design Patterns. OReilly, 2012.
- [20]. Tom White. Hadoop : The Definitive Guide, 4th Edition. OReilly, 2015.
- [21]. Hadoop Training. 2018. URL <https://www.slideshare.net/AnandMHadoop/session-19-mapreduce>.