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

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ DES MATHEMATIQUES ET DE l'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Génie Logiciel

Par :

BELKHODJA Rania
GUELFOUT Nour Elhouda

Sur le thème

Stockage distribué et Traitement parallèle de données médicales pour le diagnostic du cancer

---

Soutenu publiquement le 18 / 09 / 2022 à Tiaret devant le jury composé de :

Mr TALBI Omar	Grade Université MCB	Président
Mr MERATI Medjded	Grade Université MCA	Encadrant
Mr BAGHDADI Mohamed	Grade Université MCB	Examinateur

2021-2022

Remerciement

Je voudrais tout d'abord exprimer mes plus profonds remerciements à **ALLAH** *le tout-puissant, pour m'avoir accordé vie, santé et paix de l'esprit sans quoi je n'aurais pu achever ce travail ;*

Et c'est une joie et un agréable devoir quand nous aurons à remercier ceux qui nous ont aidés à réaliser ce travail.

Nous exprimons notre profonde gratitude à notre encadreur Mr. Merati Medjeded pour son encadrement précieux et ses conseils les plus pertinents.

Que les membres du jury trouvent nos profonds remerciements pour avoir accepté de juger ce modeste travail.

Rania & Nour El houda

Dédicaces

Je dédie ce modeste travail :

A mes très chers parents.

A toute ma famille.

A mon binôme Guelfout Nour El houda.

A tous les étudiants de la promo GL 2021/2022.

Belkhodja Rania

Je dédie ce modeste travail :

A mes chers parents.

A toute ma famille.

A mon binôme Belkhodja Rania.

A tous les étudiants de la promo GL 2021/2022

Guelfout Nour El hoouda

Résumé

L’explosion quantitative des données numériques a obligé les chercheurs à trouver de nouvelles manières de voir et d’analyser le monde. Il s’agit de découvrir de nouveaux ordres de grandeur concernant la capture, la recherche, le partage, le stockage, l’analyse et la présentation des données.

À l’heure actuelle, Hadoop est la principale plateforme du Big Data. Utilisé pour le stockage et le traitement d’immenses volumes de données, ce Framework logiciel et ses différents composants sont utilisés par de très nombreuses entreprises pour leurs projets Big Data afin de réduire le temps d’exécution, garantir la fiabilité et la disponibilité d’un système du cluster, et c’est ce que nous avons abordé principalement dans notre projet de fin d’étude qui s’est déroulé en deux étapes : une installation et configuration d’Hadoop sur trois machines virtuelles et l’exécution d’un programme qui traite des données des images médicales du diagnostic du cancer, ce programme est nommé filtre de Canny.

أجبر الانفجار الكمي للبيانات الرقمية الباحثين على إيجاد طرق جديدة لرؤية العالم وتحليله. يتعلق الأمر باكتشاف أوامر جديدة من حيث الحجم لالتقاط البيانات والبحث عنها ومشاركتها وتخزينها وتحليلها وتقديمها.

اليوم ، Hadoopهي منصة البيانات الضخمة الرئيسية. تُستخدم لتخزين ومعالجة كميات ضخمة من البيانات ، يتم استخدام إطار عمل هذا البرنامج ومكوناته المختلفة من قبل العديد من الشركات لمشاريع البيانات الضخمة الخاصة بهم من أجل تقليل وقت التفيذ ، وضمان موثوقية وتوافر نظام الكتلة ، وهذا ما نحن أساسًا تم تناوله في مشروع نهاية الدراسة الخاص بنا والذي تم على مرحلتين: تثبيت وتكوين Hadoop على ثلاثة أجهزة افتراضية وتشغيل برنامج يعالج بيانات الصور الطبية لتشخيص السرطان ، وهذا البرنامج يسمى.Canny filter

The quantitative explosion of digital data has forced researchers to find new ways of seeing and analyzing the world. It is about discovering new orders of magnitude for capturing, searching, sharing, storing, analyzing, and presenting data.

Today, Hadoop is the main Big Data platform. Used for storing and processing huge volumes of data, this software framework and its various components are used by many companies for their Big Data projects in order to reduce execution time, guarantee the reliability and

availability of a cluster system, and this is what we mainly covered in our graduation project which took place in two stages: an installation and configuration of Hadoop on three virtual machines and the execution of a program which processes medical image data of cancer diagnosis, this program is named Canny filter.

Table des matières

Liste de figures

Liste des abréviations

IT: Information Technology

ACM: Association for Computing Machinery

HDFS: Hadoop Distributed File System

NoSQL: Not only SQL

SQL: Structured Query Language

ETL: Extract – Tranform – Load

DSI : Direction des Systèmes d'Information

VM : Virtual Machine

RAID: Redundant Array of Indepandant Disks

ACID : Atomicité, Cohérence, Isolation et Durabilité

API: Application Programing Interface

JSON: JavaScript Object Notation

BSD: Berkeley Software Distribution

SaaS: Software as a Service

PaaS: Plateform as a Service

IaaS: Infrastructure as a Service

HPCC: High Performance Computing Cluster

GFS: Google File System

Fs: File system

YARN: Yet Another Ressource Negotiator

RM: Ressource Manager

JAR: Java Archive

Po: Pétaoctet

Go: Gigaoctet

To: Téraoctet

Pb : Pétabit

IoT : Internet des objets

MLlib: Machine Learning library

RDD: Resilient Distributed Dataset

DAG: Directed Acyclic Graph

SSH: Secure Shell

Introduction générale

Introduction Générale

Nous sommes confrontés depuis quelques années à de nouvelles technologies qui envahissent le monde de l'informatique et l'internet ; cette situation nous oblige à prendre les défis, de connaitre et maitriser ces nouvelles sciences afin de nous permettre de s'adapter aux changements forcés par cette révolution.

Le volume de données exploité par les entreprises a considérablement augmenté émanant de sources diverses (transaction, systèmes d'information automatisés, réseaux…), il est souvent susceptible de croitre très rapidement.

Lorsqu'on parle de manipulation de gros volume de données, on pense généralement à des problématiques sur le volume des données et sur la rapidité de traitement.

Beaucoup de concepts inséparables dominent actuellement le monde de l'IT (Information Technology). On entend souvent de Cloud Computing, la technologie actuelle, hébergeant un big data sous forme NoSQL et traité par un simple programme MapReduce dans des « clusters » distribués partout dans le monde. Le terme Cloud Computing, ou « informatique dans les nuages », est un nouveau modèle informatique qui consiste à proposer les services informatiques sous forme de services à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui. Cette nouvelle technologie permet aux entreprises d'externaliser le stockage de leurs données et de leur fournir une puissance de calcul supplémentaire pour le traitement de grosse quantité de données.

Cette technologie a besoin des plates-formes et des Framework de Cloud Computing , comme Hadoop qui est un Framework libre, géré par la fondation Apache, conçu pour analyser de très grandes quantités de données. Il supporte le passage à l'échelle, il est très performant en termes de tolérance aux pannes, il est composé d'HDFS (Hadoop Distributed File System), son propre système de fichiers et de MapReduce son « moteur » de calcul distribué. Spark est un grand Framework de traitement des données open source qui peut traiter des quantités massives de données à grande vitesse à l'aide de l'informatique en grappe, et d'autres Framework qui existent (HPCC, Storm…)

L'objectif de ce projet de fin d'étude est de faire un traitement parallèle des données massives des images médicales.

Afin de réaliser l'objectif visé, nous avons organisé ce mémoire en trois chapitres. Le premier chapitre présente le Big Data et le Cloud Computing, on a également présenté le NoSQL, l'un des principales technologies du Big Data.

Le deuxième chapitre est consacré aux Framework : Hadoop et Spark en décrivant leurs principaux composants.

Le troisième chapitre se portera sur la mise en œuvre de l'application, nous introduirons les outils et logiciels ayant servi à l'élaboration du projet, ensuite l'installation et la configuration de Cluster, tout en expliquant les différentes étapes à suivre, nous nous passerons enfin au stockage distribué des données et les traiter parallèlement en appliquant le filtre de Canny.

Nous terminerons par une conclusion générale résumant les connaissances acquises durant la réalisation du projet.

Chapitre 1 :

Big data et Cloud Computing

1 Chapitre 1 : Big Data et Cloud Computing

1.1 Introduction

Les volumes de données augmentent constamment et représentent une complexité majeure en termes de conception. Les projets de Big Data commencent généralement par le stockage de données et l'application de modules d'analyse de base. Cependant, à mesure que nous découvrirons des moyens d'extraire des données à une échelle beaucoup plus grande, nous devrons trouver de meilleures méthodes pour traiter et analyser ces données. Cela va sans doute demander des mises à niveau de l'infrastructure. Nous pouvons ajouter plus de capacité à notre base de données interne ou alimenter plus de serveurs pour répondre aux besoins d'analyse en constante augmentation. Mais même en boostant nos systèmes sur site, notre infrastructure risque de ne pas pouvoir suivre le rythme.

C'est là que le cloud entre en jeu, ou de manière plus appropriée, c'est le moment de transférer nos mégadonnées dans le cloud. Les technologies de Cloud offrent un soutien efficace et économique pour obtenir des architectures évolutives et pour fournir à l'administration de l'entreprise les outils nécessaires à l'analyse du volume sans cesse croissant des informations afin de prendre des décisions stratégiques. Pour ce faire, des fournisseurs comme Amazon et Google ont conçu des technologies d'analyse très rapides et utiles pour la collecte et la gestion des Big Data.

1.2 Big Data

1.2.1 Définition

Les Big Data ou mégadonnées désignent l'ensemble des données numériques produites par l'utilisation des nouvelles technologies à des fins personnelles ou professionnelles. Cela recoupe les données d'entreprise (courriels, documents, bases de données, historiques de processus métiers...) aussi bien que des données issues de capteurs, des contenus publiés sur le web (images, vidéos, sons, textes), des transactions de commerce électronique, des échanges sur les réseaux sociaux, des données transmises par les objets connectés (étiquettes électroniques, compteurs intelligents, smartphones...), des données géolocalisées, etc.

En d'autres termes, le Big Data est composé de jeux de données complexes plus variées, arrivant dans des volumes de plus en plus importants et avec une vitesse plus élevée. Cette définition est également connue sous le nom des trois « V ».

L'expression « big data » serait apparue en octobre 1997 selon les archives de la bibliothèque numérique de l'Association for Computing Machinery (ACM), dans un article scientifique sur les défis technologiques à relever pour visualiser les « grands ensembles de données » **[1]**

En 2001, l'analyste du cabinet Meta Group (devenu Gartner) Doug Laney décrivait les Big Data d'après le principe des « trois V » **[2]**

1.2.2 Les trois « V » du Big Data

Volume : La quantité de données a son importance. Avec le Big Data, vous devrez traiter de gros volumes de données non structurées et à faible densité. Il peut s'agir de données de valeur inconnue, comme des flux de données Twitter, des flux de clics sur une page web ou une application mobile ou d'un appareil équipé d'un capteur. Pour certaines entreprises, cela peut correspondre à des dizaines de téraoctets de données. Pour d'autres, il peut s'agir de centaines de pétaoctets.

Vitesse : La vitesse à laquelle les données sont reçues et éventuellement traitées. Normalement, les données sont transmises à haute vitesse directement à la mémoire, plutôt que d'être écrites sur le disque. Certains produits intelligents accessibles via Internet opèrent en temps réel ou quasi réel et nécessitent une évaluation et une action en temps réel.

Variété : La variété fait allusion aux nombreux types de données disponibles. Les types de données traditionnels ont été structurés et trouvent naturellement leur place dans une base de données relationnelles.

Figure 1 - Les 3 V de La Big Data **[3]**

Avec l'augmentation du Big Data, les données ne sont pas nécessairement structurées. Les types de données non structurés et semi-structurés, tels que le texte, l'audio et la vidéo, nécessitent un prétraitement supplémentaire pour en déduire le sens et prendre en charge les métadonnées et pour cela on a vu deux autres valeurs de V : la valeur et la véracité.

Valeur : Dans un contexte d'infobésité (L'infobésité est issue de la contraction entre les termes "information" et "obésité". Elle désigne la surcharge d'informations), il s'agit d'être capable de se concentrer sur les données ayant une réelle valeur et étant actionnables.

Véracité : La véracité ou fiabilité des données est notamment menacée par les comportements déclaratifs (sur formulaires), par les diversités des points de collecte, par la multiplication des formats de données et par l'activité des robots et faux profils innombrables sévissant sur Internet. **[4]**

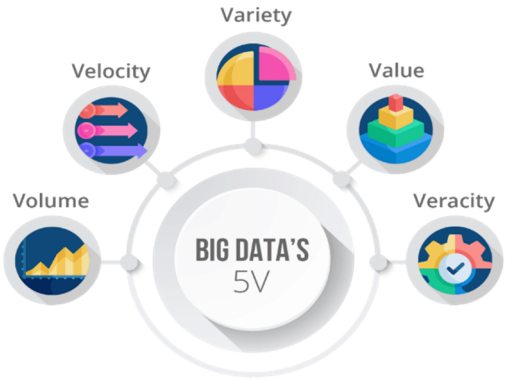

Figure 2 - Les 5 V de la Big Data **[5]**

1.2.3 Big Data et la collecte des données

Le terme " Big data " représente le volume colossal de données structurées, semi-structurées et non structurées que les entreprises peuvent collecter. Charger des ressources Big Data dans une base de données relationnelle classique revient cher et prend beaucoup de temps.

C'est pourquoi de nouvelles approches de collecte et d'analyse des données ont vu le jour. Pour réunir puis analyser les données du Big Data afin d'en tirer des informations exploitables, des données brutes accompagnées de métadonnées sont agrégées dans un lac de données. De là, les programmes d'apprentissage machine et d'intelligence artificielle utilisent des algorithmes complexes pour rechercher des modèles reproductibles. **[6]**

La couche responsable du chargement de données dans Big Data, devrait être capable de gérer d'énorme volume de données, avec une grande vitesse, et une grande variété de

données. Cette couche devrait avoir la capacité de valider, nettoyer, transformer, réduire (compression), et d'intégrer les données dans la grande pile de données en vue de son traitement. La Figure 03 illustre le processus et les composants qui doivent être présent dans la couche de chargement de données dans le Big Data.

La couche de chargement de données de Big Data collecte les informations pertinentes finales, sans bruit, et les charge dans la couche de stockage de Big Data (HDFS ou NoSQL base). Elle doit inclure les composants suivants :

Identification : des différents formats de données connues. Par défaut, le Big Data cible les données non structurées.

Filtration : sélection de l'information entrante pertinente pour l'entreprise.

Validation : et analyse des données en permanence.

Réduction : de bruit implique le nettoyage des données en supprimant le bruit.

La transformation : peut entraîner le découpage, la convergence, la normalisation ou la synthèse des données.

Compression : consiste à réduire la taille des données, mais sans perdre de la pertinence des données.

Intégration : consiste à intégrer l'ensemble des données dans le stockage de données de Big Data (HDFS ou NoSQL base). **[7]**

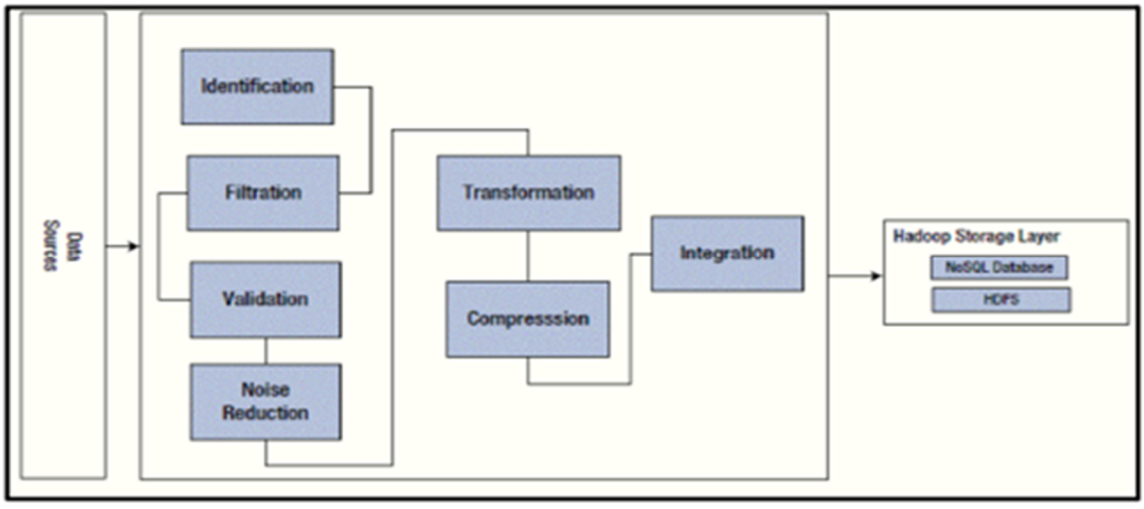

Figure 3 - La couche de chargement de donnée dans le Big Data

1.2.4 Le Big Data et le Data Warehouse

Ne confondez pas le Big Data avec un data Warehouse géant. Certains voudraient appliquer les principes d'un data Warehouse aux flux Big Data. C'est mal comprendre ce qu'est le Big Data. L'inverse, c'est-à-dire traiter sur un mode Big Data des entrepôts de données (data Warehouse) non structurées, est toutefois aujourd'hui rendu possible par les Data Lake. Ce dernier est un concept lié à la mouvance du big data, le lac de données désigne un espace de stockage global des informations présentes au sein d'une organisation. Il s'agit de le faire avec suffisamment de flexibilité pour interagir avec les données, qu'elles soient brutes ou très raffinées.

Un entrepôt de données, ou data Warehouse, est un modèle classique et efficace de traitement d'informations business. Les données sont tirées de diverses sources. Après la collecte, elles sont triées et stockées, avant de servir de base aux solutions décisionnelles. Un data Warehouse utilise un processus d'ETL en entrée pour capter les données issues de systèmes tiers. Lors de la définition de nouveaux scénarios de traitement des informations, la DSI s'aperçoit parfois que les données stockées dans l'entrepôt ne présentent pas tous les attributs nécessaires. Il faut alors repenser la phase d'ETL.

Le Big Data est différent. Les requêtes analytiques s'effectuent directement sur le flot de données non structurées, sans extraction et stockage préalable au sein d'un data Warehouse. Les utilisateurs gagnent ainsi en souplesse dans leurs requêtes. S'ils veulent appliquer un nouveau type de traitement aux données, il leur suffit de créer une nouvelle requête. Problème du Big Data, si vous voulez rejouer une requête, ce n'est pas possible, car les données sont traitées au fil de l'eau, sans être conservées. Beaucoup ne comprennent pas cette nuance entre les deux approches et essaient de stocker l'ensemble des flux Big Data, pour pouvoir rejouer leurs requêtes à volonté. **[8]**

1.2.5 Architecture de Big data

La plupart des architectures de données volumineuses incluent tout ou partie des éléments suivants :

- Source de données (data Mart, data Warehouse, cloud, base de données hybride).
- Stockage (magasin de données, data Lake).

- Batch processing (traitement par lots).
- Stream processing (traitement de flux de data).
- Préparation de données.
- Data catalogue
- Modélisation de données.
- Technologie d'orchestration. **[9]**

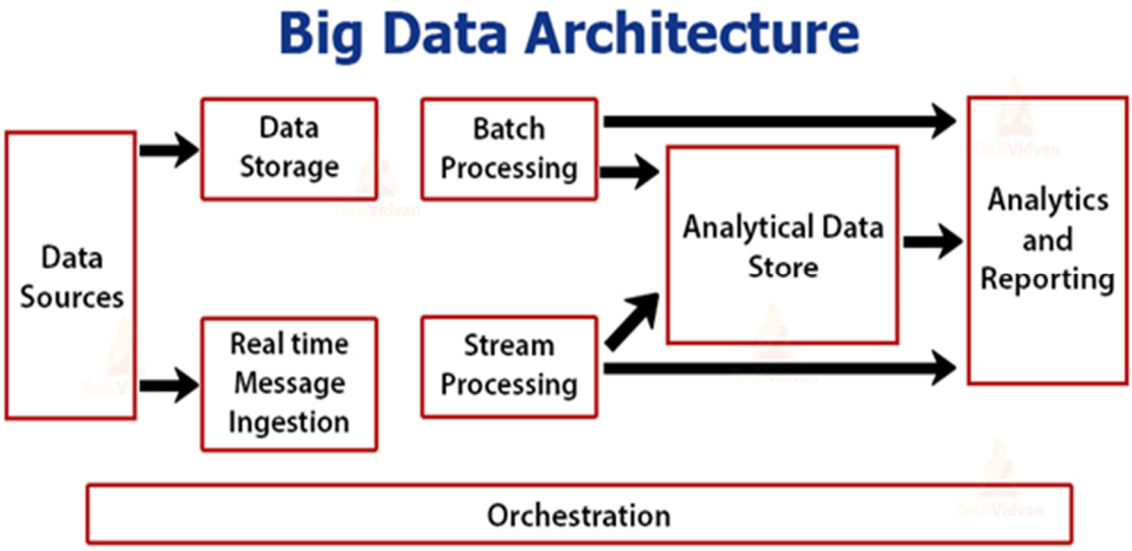

Figure 4 - Architecture de Big Data **[10]**

Il existe deux principaux types d'architecture Big Data : Lambda et Kappa. Chacune de ces architectures permet de répondre à un besoin spécifique. Le choix du modèle architectural le plus adapté à votre stratégie dépend de vos besoins, de vos infrastructures existantes, de vos objectifs et de votre contexte métier.

Dans tous les cas, lorsque l'on souhaite mener des projets Data-Driven (gouverné par la donnée), il faut avoir en tête que c'est une architecture distribuée qui doit être implémentée pour considérer les problèmes de scalabilité, de performance et de synchronisation des différentes couches. On distingue principalement les couches suivantes :

- **Couche matériel (infrastructure Layer)** : peut-être des serveurs virtuels VMware, ou des serveurs lame (blade server).

- **Couche stockage (Storage layer) :** les données seront stockées soit dans une base NoSQL, ou bien directement dans le système de fichier distribué ou les Datawarehouse.

- **Couche management et traitement :** on trouve dans cette couche les outils de traitement et analyse des données comme MapReduce ou Pig. **[11]**

1.2.5.1 À quoi sert la Data Architecture

Une Data Architecture a de nombreux intérêts pour l'entreprise. Elle permet aux organisations de se préparer stratégiquement pour évoluer rapidement et tirer profit des opportunités liées aux technologies émergentes. Son but est aussi de traduire les besoins de l'entreprise en besoins de données et systèmes informatiques. Elle simplifie donc l'alignement du département informatique avec l'activité. L'architecture de données permet aussi de gérer la diffusion d'informations et de données complexes à travers l'entreprise. L'organisation peut donc gagner en agilité. **[12]**

D'un autre coté plusieurs avantages peuvent être associés à une architecture Big Data, nous pouvons citer par exemple :

- Évolutivité (scalabilité) : Quelle est la taille que devra avoir votre infrastructure ? Combien d'espace disque est nécessaire aujourd'hui et à l'avenir ? le concept Big Data nous permet de s'affranchir de ces questions, car il apporte une architecture scalable.
- Performance : Grâce au traitement parallèle des données et à son système de fichiers distribués, le concept Big Data est hautement performant en diminuant la latence des requêtes.
- Coût faible : Le principal outil Big Data à savoir Hadoop est un Open Source, en plus on n'aura plus besoin de centraliser les données dans des baies de stockage souvent excessivement chère, avec le Big Data et grâce au système de fichiers distribués les disques internes des serveurs suffiront.
- Disponibilité : On a plus besoin des RAID disques, souvent coûteux. L'architecture Big Data apporte ses propres mécanismes de haute disponibilité.

1.2.6 Les applications concrètes du Big Data

1.2.6.1 Domaine de la santé

Par exemple, le Big Data favorise une médecine préventive et personnalisée. Ainsi, l'analyse des recherches des internautes sur un moteur de recherche a déjà permis de détecter plus rapidement l'arrivée d'une épidémie de grippe. Dans un futur proche, les

appareils connectés devraient permettre l'analyse en continu des données biométriques des patients.

Prenons l'exemple de la génomique, considérée comme la « reine du Big data ». L'obtention de la première séquence du génome humain (nos 23 000 gènes, soit un « texte » de 3 milliards de lettres), révélée en 2003, a pris près de 15 ans et coûté près de 3 milliards de dollars. Aujourd'hui, on peut obtenir la séquence du génome d'un individu en une journée, grâce aux techniques de séquençage « à haut débit », pour moins de 1000 dollars.**[13]**

1.2.6.2 Domaine des transports

L'analyse des données du Big Data (données provenant des lignes de transport en commun, géolocalisation des personnes et des voitures, etc.) permet de modéliser les déplacements des populations afin d'adapter les infrastructures et les services (horaires et fréquence des trains, par exemple).

1.2.6.3 Domaine de la gestion énergétique

L'analyse des données issues du Big Data intervient dans la gestion de réseaux énergétiques complexes via les réseaux électriques intelligents (smart grids) qui utilisent des technologies informatiques pour optimiser la production, la distribution et la consommation de l'électricité.

1.2.6.4 Domaine d'aviation

L'analyse des données provenant de capteurs sur les avions (données de vol) associées à des données météo permet de modifier les couloirs aériens afin de réaliser des économies de carburant et d'améliorer la conception et la maintenance des avions. Il existe des utilisations concrètes du Big Data dans de nombreux autres domaines : recherche scientifique, marketing, développement durable, commerce, éducation, loisirs, sécurité, etc. **[14]**

1.2.7 Technologies du Big Data

1.2.7.1 Map Reduce

Au départ, il y eut "Map Reduce", une méthode et une technologie de traitement massivement parallèle issues des laboratoires Google Corp ® avec gestion de la tolérance aux

pannes et système de gestion de fichiers spécifiques (Google File System). On parle là de traitement sur des milliers de machines réparties en grappes (clusters).

1.2.7.2 Hadoop

Ensuite, il y eut "Hadoop", un framework mis au point par l'Apache Software Fondation afin de mieux généraliser l'usage du stockage et traitement massivement parallèle de Map Reduce et de Google File System. Bien entendu, Hadoop possède ses limites. Quoi qu'il en soit, c'est une solution de Big Data très largement utilisée pour effectuer des analyses sur de très grandes quantités de données.

1.2.7.3 Bases No SQL

Les bases de données non relationnelles ont une philosophie d'organisation des données bien spécifiques, avec notamment le langage d'interrogation SQL, le principe d'intégrité des transactions (ACID), et les lois de normalisation. Bien utiles pour gérer les données qualifiées de l'entreprise, elles ne sont pas du tout adaptées au stockage de très grandes dimensions et au traitement ultra rapide. Les bases NoSQL autorisent la redondance pour mieux servir les besoins en matière de flexibilité, de tolérance aux pannes et d'évolutivité

1.2.7.4 Stockage "In-Memory"

Pour des analyses encore plus rapides, les traitements directement en mémoire sont une solution. Une technologie bien qu'encore trop coûteuse il est vrai qu'elle se popularise.

1.2.7.5 Cloud Computing

Le Big Data exige une capacité matérielle hors du commun, que ce soit pour le stockage comme pour les ressources processeurs nécessaires au traitement. Nul besoin de s'équiper outre mesure, le "Cloud" est là pour cela. Encore faut-il avoir bien compris le concept pour différencier, le Cloud privé du Cloud public, l'interne de l'externe et les hybrides combinant plusieurs types de solutions.**[15]**

1.2.8 Base de données NOSQL

Le terme « NoSQL » désigne les différents types de bases de données non relationnelles. Ces bases de données stockent les données dans un format différent. Toutefois, les bases de données NoSQL peuvent être interrogées à l'aide d'API en langage idiomatique,

de langages déclaratifs et de langages de requête par exemple, ce qui explique pourquoi elles sont également considérées comme des bases de données « pas seulement SQL ».

1.2.8.1 À quoi sert une base de données NoSQL

Les bases de données NoSQL :

- Utilisées dans les applications Web et le Big data en temps réel, car elles présentent le principal avantage de proposer une évolutivité élevée et une haute disponibilité.
- Préférées par les développeurs, car elles se prêtent naturellement à un paradigme de développement agile en s'adaptant rapidement à l'évolution des exigences.
- Permettent de stocker les données de manière plus intuitive et plus facile à comprendre, ou plus proche de la façon dont elles sont utilisées par les applications, avec moins de transformations requises lors du stockage ou de l'extraction à l'aide d'API de type NoSQL.
- Peuvent tirer pleinement parti du cloud pour éviter tout temps d'inactivité.

1.2.8.2 Quand choisir une base de données NoSQL

Alors que les entreprises et les organisations ont besoin d'innover rapidement, il est essentiel pour elles de pouvoir rester agiles et continuer à travailler à n'importe quelle échelle. Les bases de données NoSQL proposent des schémas flexibles et prennent également en charge une variété de modèles de données idéaux pour créer des applications qui nécessitent de grands volumes de données et des temps de latence ou de réponse faibles (par exemple, des jeux en ligne et des applications Web d'e-commerce).

1.2.8.3 Les types de bases de données NoSQL

Il existe quatre principaux types de base de données NoSQL :

- **Valeur clé :** Il s'agit du type de base de données NoSQL le plus flexible, car l'application dispose d'un contrôle total sur ce qui est stocké dans le champ de valeur sans aucune restriction.
- **Document :** Également appelées bases de données orientées documents ou répertoires de documents, ces bases de données sont utilisées pour stocker, extraire et gérer des données semi-structurées. Il n'est pas nécessaire de spécifier les champs qu'un document contiendra.

- **Graphe :** Ce type de bases de données organise les données en tant que nœuds et relations, qui indiquent les connexions entre ces nœuds. Il prend en charge une représentation plus riche et plus complète des données. Les bases de données de graphes sont utilisées pour les réseaux sociaux, les systèmes de réservation et la détection des fraudes.
- **Colonne large :** Ces bases de données stockent et gèrent les données sous forme de tables, de lignes et de colonnes. Elles sont largement déployées dans des applications qui nécessitent un format de colonne pour capturer des données sans schéma.**[16]**

1.2.8.4 TOP 11 des Meilleures bases de données NoSQL en 2022

- **MongoDB :** Il s'agit d'une base de données NoSQL open source orientée document. MongoDB utilise des documents de type JSON pour stocker toutes les données. Il est écrit en C++.
- **Cassandra :** Il a été développé sur Facebook pour la recherche dans les boîtes de réception. Cassandra est un système de stockage de données distribué pour le traitement de très grandes quantités de données structurées.
- **Redis :** Redis est la plus célèbre base clé-valeur. Redis est composé en langage C. Il est autorisé sous BSD.
- **Hbase** : Il s'agit d'une base de données distribuée et non relationnelle qui est conçue pour la base de données BigTable par Google.
- **Neo4j :** Neo4j est considéré comme une base de données de graphes native car il implémente efficacement le modèle de graphes de propriétés jusqu'au niveau de stockage.
- **RAVENDB :** RavenDB est la base de données documentaire NoSQL originale qui offre une intégrité des données entièrement transactionnelle (ACID) sur plusieurs documents de votre base de données et sur l'ensemble de votre cluster de bases de données.
- **Oracle NoSQL :** Oracle NoSQL Database implémente une carte allant des clés définies par l'utilisateur aux éléments de données non structurées.
- **DynamoDB :** DynamoDB utilise un modèle de base de données NoSQL qui n'est pas relationnel, ce qui permet d'avoir des documents, des graphiques et des colonnes parmi ses modèles de données.

- **Couchbase :** Couchbase Server est une base de données de documents NoSQL pour les applications Web interactives. Il dispose d'un modèle de données flexible, est facilement évolutif et offre des performances élevées et constantes.
- **Memcached :** Il s'agit d'un système de mise en cache de mémoire distribuée de haute performance, son code source est ouvert, destiné à accélérer les applications Web dynamiques en réduisant la charge de la base de données.
- **CouchDB :** C'est une base de données NoSQL Open Source qui utilise JSON pour stocker les informations et utilise JavaScript comme langage de requête. **[17]**

1.3 Cloud Computing

L'idée du Cloud Computing a pris naissance en 1990 et surtout en 1991 avec la naissance d'Internet et la mise sur le marché du logiciel CERN qui a été le premier logiciel accessible par le Web. Le Cloud Computing était créé par Joseph Carl Robnett Licklider, moins connu, il apporta pourtant lui aussi un fort coup de pouce à la « démarche Internet » fondatrice de l'informatique en nuages. **[18]**

Son but est de pouvoir exploiter Internet, de manière à y stocker tous ses outils informatiques, logiciels et systèmes, ainsi que ses données, de manière à pouvoir bénéficier d'un accès permanent à ceux-ci, où que l'on se trouve, pourvu évidemment que l'on soit connecté au web.

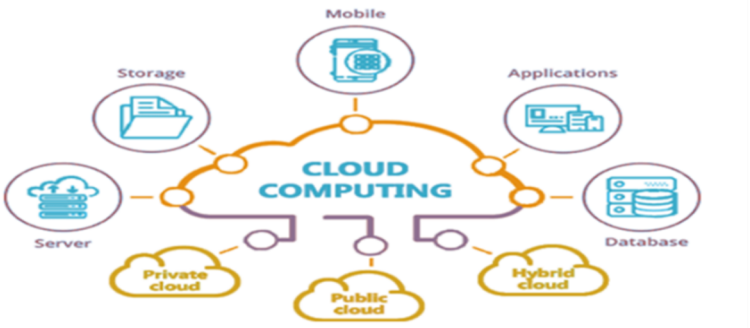

Figure 5 - Cloud Computing **[19]**

1.3.1 Définition

En termes simples, le Cloud Computing nous permet de louer plutôt que d'acheter votre informatique. Plutôt que d'investir massivement dans des bases de données, des logiciels et du matériel, les entreprises choisissent d'accéder à leur puissance de calcul via l'Internet, ou le Cloud, et de la payer au fur et à mesure de son utilisation. Ces services Cloud incluent désormais, mais sans s'y limiter, les serveurs, le stockage, les bases de données, le réseau, les

logiciels, les analyses et la Business Intelligence. Le Cloud Computing offre la vitesse, l'évolutivité et la flexibilité nécessaires pour permettre aux entreprises de développer, d'innover et de prendre en charge des solutions informatiques.**[20]**

1.3.2 Avantages du Cloud Computing

Le Cloud Computing constitue une alternative supérieure aux technologies de l'information traditionnelles, notamment dans les points suivants :

- Coût : Élimination des dépenses d'investissement
- Vitesse : Provisionnement rapide de l'espace pour le développement et les tests
- Portée mondiale : Évolutivité globale élastique
- Productivité : Collaboration accrue pour la productivité, performances prévisibles et isolement client
- Performance : Meilleur rapport prix/performances pour les charges de travail Cloud native
- Fiabilité : Systèmes distribués évolutifs et tolérants aux pannes de fiabilité pour tous les services.**[21]**

1.3.3 Types de Cloud Computing

Il existe trois types de Cloud : public, privé et hybride. Chaque type requiert un niveau de gestion différent du client et fournit un niveau de sécurité différent.

1.3.3.1 Cloud public

Dans un Cloud public, l'ensemble de l'infrastructure informatique est situé dans les locaux du fournisseur de Cloud, et ce dernier fournit des services au client via Internet. Les clients n'ont pas à maintenir leur propre système informatique et peuvent ajouter rapidement plus d'utilisateurs ou de puissance de calcul selon leurs besoins. Dans ce modèle, plusieurs locataires partagent l'infrastructure informatique du fournisseur de Cloud.

1.3.3.2 Cloud privé

Un Cloud privé est utilisé exclusivement par une entreprise. Il peut être hébergé dans les locaux de l'entreprise ou dans le Datacenter du fournisseur de Cloud. Un Cloud privé fournit le niveau de sécurité et de contrôle le plus élevé.

1.3.3.3 Cloud hybride

Comme son nom l'indique, un Cloud hybride est une combinaison de Cloud public et privé. En général, les clients du Cloud hybride hébergent leurs applications critiques sur leurs propres serveurs pour plus de sécurité et de contrôle, et stockent leurs applications secondaires chez le fournisseur de Cloud.

1.3.3.4 Multi-Cloud

La principale différence entre le Cloud hybride et le Multi-Cloud est que ce dernier utilise plusieurs ordinateurs et terminaux de stockage Cloud dans une seule architecture.**[22]**

1.3.4 Services de Cloud Computing

Il existe trois principaux types de service Cloud : logiciel en tant que service (SaaS), plateforme en tant que service (PaaS) et infrastructure en tant que service (IaaS). Il n'existe pas d'approche unique du Cloud, il s'agit plutôt de rechercher la solution adaptée à vos besoins.

1.3.4.1 SaaS

Le SaaS est un modèle de diffusion de logiciel dans lequel le fournisseur de Cloud héberge les applications du client sur l'emplacement du fournisseur de Cloud. Le client accède à ses applications via Internet. Plutôt que de payer et d'entretenir leur propre infrastructure informatique, les clients SaaS bénéficient d'un abonnement au service sur la base d'un paiement à l'utilisation.

1.3.4.2 PaaS

Le PaaS (plateforme en tant que service) offre aux clients l'avantage d'accéder aux outils de développement dont ils ont besoin pour créer et gérer des applications web et mobiles sans avoir à investir dans (ou entretenir) l'infrastructure sous-jacente. Le fournisseur héberge l'infrastructure et les composants middleware, et le client accède à ces services via un navigateur web.

1.3.4.3 IaaS

Le IaaS permet aux clients d'accéder à des services d'infrastructure à la demande via Internet. Le principal avantage est que le fournisseur Cloud héberge les composants d'infrastructure fournissant la capacité de calcul, de stockage et de réseau, de sorte que les abonnés puissent exécuter leurs charges de travail dans le Cloud. L'abonné au Cloud

est généralement responsable de l'installation, de la configuration, de la sécurisation et de la maintenance de tout logiciel sur les solutions Cloud native, comme les bases de données, le middleware et les logiciels d'application.**[23]**

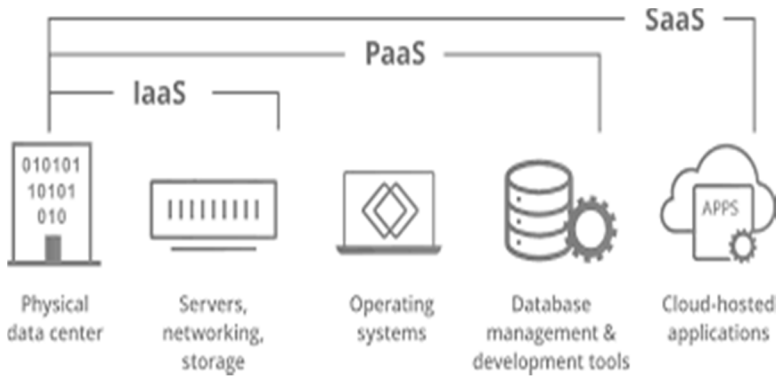

Figure 6 - Différentes Couches De Cloud. **[24]**

1.4 Conclusion

Ce chapitre a permis de cerner ce terme de **« Big Data »** ou mégadonnées, actuellement au centre des préoccupations des acteurs de tous les domaines d'activité, en évoquant les principaux enjeux économiques et sociétaux associés.

Nous avons aussi introduit les différentes grandes méthodes et techniques qui s'y rattachent, tant en ce qui concerne leur stockage, que leur exploitation et leur analyse. Ces méthodes et techniques, tout comme les outils logiciels qui y sont rattachés sont encore en devenir.

Chapitre 2 :

Apache Hadoop et Apache Spark

2 Chapitre 2 : Apache Hadoop et Apache Spark

2.1 Introduction

Au cours de la dernière décennie, les systèmes répartis à large échelle ont connu une popularité importante en raison de leurs caractéristiques particulières, comme le passage à l'échelle et la tolérance aux fautes. Certains systèmes orientés données (BigData), comme les systèmes Pairà-Pairs et MapReduce, ont atteint des millions d'utilisateurs et des pétaoctets de données traitées : Hadoop, Spark, Strom, HPCC etc.

Dans ce chapitre nous allons présenter les Framework : Hadoop et Apache Spark qui opèrent sur les Cloud Computing, leurs objectifs, et leurs fonctionnalités. Ces deux outils sont parfois considérés comme des concurrents, il est souvent admis qu'ils fonctionnent encore mieux quand ils sont ensemble. On va faire un aperçu de leurs caractéristiques et de leurs différences.

2.2 Framework

Un Framework est un ensemble d'outils et de compostant logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme. Il est souvent fourni sous la forme d'une bibliothèque logicielle, et accompagné du plan de l'architecture cible du Framework.

Le Framework va permettre de mettre en place une application en effectuant des séries d'opérations de manière simplifiée en utilisant des méthodes qui auront été conçues au préalable au sein des librairies du Framework

Nous pouvons trouver des Frameworks gratuits ou payants, open-source ou non. Les communautés qui les entourent sont généralement très présentes et réactives. Ce qui permet de fournir des Frameworks gratuits d'excellente qualité.**[25]**

2.3 Apache Hadoop

2.3.1 Définition

Hadoop est un Framework logiciel open source permettant de stocker des données, et de lancer des applications sur des grappes de machines standards. Cette solution offre un espace de stockage massif pour tous les types de données, une immense puissance de traitement et la possibilité de prendre en charge une quantité de tâches virtuellement illimitée. Basé sur Java, ce Framework fait partie du projet Apache, sponsorisé par Apache Software Fondation.

Grâce au Framework MapReduce, il permet de traiter les immenses quantités de données. Plutôt que de devoir déplacer les données vers un réseau pour procéder au traitement, MapReduce permet de déplacer directement le logiciel de traitement vers les données. **[26]**

Il est utilisé pour le stockage et traitement des Big Data. Les données sont stockées sur des serveurs standards peu coûteux configurés en clusters. Le système de fichiers distribué Hadoop supporte des fonctionnalités de traitement concurrent et de tolérance aux incidents. **[27]**

2.3.2 Historique

Hadoop trouve ses racines dans les technologies propriétaires d'analyse de données de Google. En 2004, le moteur de recherche a publié un article de recherche présentant son algorithme MapReduce, conçu pour réaliser des opérations analytiques à grande échelle sur un grand cluster de serveurs, et sur son système de fichier en cluster, Google File_System (GFS).

Doug Cutting, qui travaillait alors sur le développement du moteur de recherche libre Apache Lucene et butait sur les mêmes problèmes de volumétrie de données qu'avait rencontré Google, s'est alors emparé des concepts décrits dans l'article du géant de la recherche et a décidé de répliquer en open source les outils développés par Google pour ses besoins. Employé chez Yahoo, il s'est alors lancé dans le développement de ce qui est aujourd'hui le projet Apache Hadoop – pour la petite histoire, Hadoop est le nom de l'éléphant qui servait de doudou à son jeune fils. **[28]**

2.3.3 Caractéristique d'Hadoop

Robuste : si un nœud de calcul tombe, ses tâches sont automatiquement réparties sur d'autres nœuds. Les blocs de données sont également répliqués.

Coût : il optimise les coûts via une meilleure utilisation des ressources présentées.

Souple : car il répond à la caractéristique de variété des données en étant capable de traiter différents types de données.

Virtualisation : ne plus se reposer directement sur l'infrastructure physique (baie de stockage coûteuse), mais choisir la virtualisation de ses clusters Hadoop.

2.3.4 Architecture

Le Framework Hadoop principal comprend quatre modules qui fonctionnent ensemble pour former l'écosystème Hadoop :

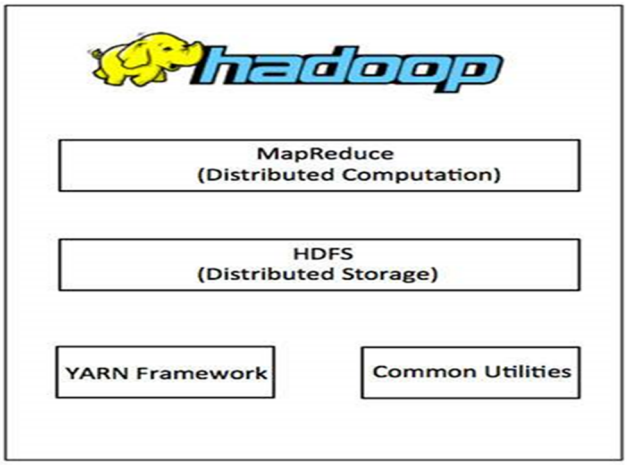

Figure 7 - Architecture de Hadoop. **[29]**

2.3.4.1 Hadoop Distributed File System (HDFS)

Composant principal de l'écosystème Hadoop, HDFS est un système de fichiers distribué qui fournit un accès haut débit aux données d'application, sans avoir à définir de schémas au départ.

HDFS contient une très grande quantité de données et offre un accès plus facile. Pour stocker des données aussi énormes, les fichiers sont stockés sur plusieurs machines. Ces fichiers sont stockés de manière redondante pour sauver le système d'éventuelles pertes de données en cas de panne. HDFS rend également les applications disponibles pour le traitement parallèle.

2.3.4.1.1 Caractéristiques de HDFS

- Il convient au stockage et traitement distribués.
- Hadoop fournit une interface de commande pour interagir avec HDFS.
- Les serveurs intégrés de namenode et de datanode aident les utilisateurs à vérifier facilement l'état du cluster.
- Accès en continu aux données du système de fichiers.
- HDFS fournit des autorisations de fichiers et une authentification.

2.3.4.1.2 Composants HDFS

HDFS suit l'architecture maître-esclave et comporte les éléments suivants :

- **Namenode**

  Le namenode est le matériel de base qui contient le système d'exploitation GNU / Linux et le logiciel namenode. C'est un logiciel qui peut être exécuté sur du matériel de base. Le système ayant le Namenode agit en tant que serveur maître et effectue les tâches suivantes :

  - Gère l'espace de noms du système de fichiers.
  - Régule l'accès du client aux fichiers.
  - Il exécute également les opérations du système de fichiers telles que le changement de nom, la fermeture et l'ouverture de fichiers et de répertoires.

- **Datanode**

  Le Datanode est un matériel de base ayant le système d'exploitation GNU / Linux et un logiciel de Datanode. Pour chaque nœud (matériel / système de base) dans un cluster, il y aura un Datanode. Ces nœuds gèrent le stockage des données de leur système.

  - Les Datanodes effectuent des opérations de lecture-écriture sur les systèmes de fichiers, conformément à la demande du client.
  - Ils effectuent également des opérations telles que la création de blocs, la suppression et la réplication selon les instructions du Namenode.

- **Bloquer**

  En général, les données utilisateur sont stockées dans les fichiers de HDFS. Le fichier dans un système de fichiers sera divisé en un ou plusieurs segments et / ou stocké dans des nœuds de données individuels. Ces segments de fichier sont appelés blocs. En d'autres termes, la quantité minimale de données que HDFS peut lire ou écrire est appelée un bloc. La taille de bloc par défaut est de 64 Mo, mais elle peut être augmentée selon la nécessité de modifier la configuration HDFS. **[30]**

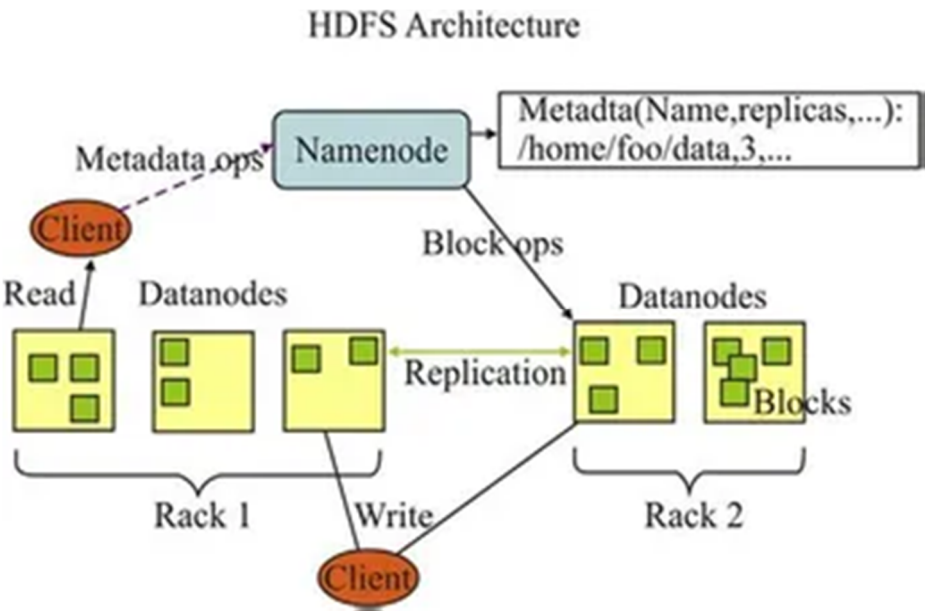

Figure 8 - Architecture de HDFS **[29]**

2.3.4.1.3 Lecture d'un fichier HDFS

Pour lire un fichier au sein de HDFS, il faut suivre les étapes suivantes :

Etape 1 : Le client indique au NameNode qu'il souhaite lire le fichier data.txt

Etape 2 : Le NameNode lui indiquera la taille de fichier (nombre de blocs) ainsi que les différents Data Node hébergeant les n blocs.

Etape 3 : Le client récupère chacun des blocs à un des DataNodes.

Etape 4 : En cas d'erreur/non réponse d'un des DataNode, il passe au suivant dans la liste fournie par le NameNode.

2.3.4.1.4 Écriture dans un fichier ou volume HDFS

Pour écrire un fichier au sein de HDFS :

Étape 1 : On va utiliser la commande principale de gestion de Hadoop : Hadoop, avec l'option fs. Admettons qu'on souhaite stocker le fichier data.txt sur HDFS.

Étape 2 : Le programme va diviser le fichier en blocs de 64KB (ou autre, selon la configuration) –supposons qu'on ait ici 3 blocs.

Étape 3 : Le NameNode lui indique les DataNodes à contacter.

Étape 4 : Le client contacte directement le DataNode concerné et lui demande de stocker le bloc.

Etape 5 : les DataNodes s'occuperont – en informant le NameNode – de répliquer les données entre eux pour éviter toute perte de données.

Etape 6 : Le cycle se répète pour le bloc suivant.**[31]**

2.3.4.2 MapReduce

Est un modèle de programmation qui permet de traiter de grands volumes de données. Grâce à des algorithmes de calcul parallèles et distribués, MapReduce permet de transférer une logique de traitement et d'écrire des applications qui transforment de grands volumes de données en un ensemble gérable. **[32]**

L'algorithme MapReduce contient deux tâches importantes, à savoir Map et Reduce. Map prend un ensemble de données et le convertit en un autre ensemble de données, où les éléments individuels sont décomposés en tuples (paires clé / valeur). Reduce réduit la tâche qui prend la sortie d'une carte comme entrée et combine ces tuples de données en un ensemble plus petit de tuples. Comme la séquence du nom MapReduce l'indique, la tâche de réduction est toujours effectuée après la tâche de carte.

À un niveau élevé, MapReduce divise les données d'entrée en fragments et les distribue sur différentes machines. Les fragments d'entrée sont constitués de paires clé-valeur. Les tâches de mappage parallèle traitent les données fragmentées sur les machines d'un cluster. La sortie de mappage sert ensuite d'entrée pour l'étape de réduction. La tâche de réduction combine le résultat dans une sortie de pair clé-valeur particulière et écrit les données dans HDFS.

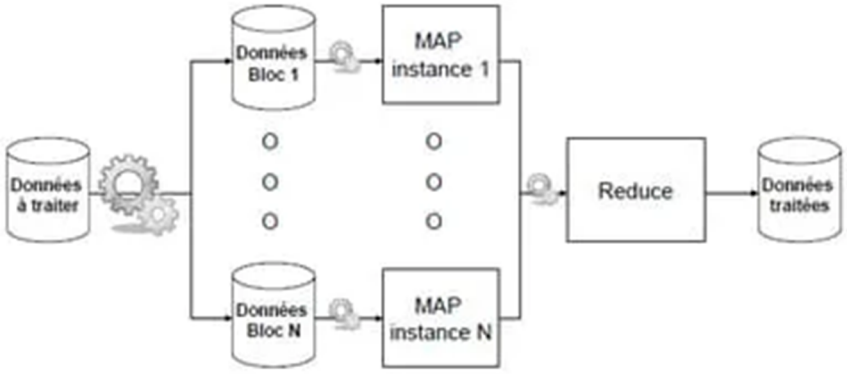

Figure 9 - Le Traitement de MapReduce **[33]**

Le principal avantage de MapReduce est qu'il est facile de faire évoluer le traitement des données sur plusieurs nœuds de calcul. Dans le modèle MapReduce, les primitives de traitement des données sont appelées mappeurs et réducteurs. Décomposer une application de traitement de données en mappeurs et réducteurs n'est parfois pas trivial. Mais, une fois que nous écrivons une application sous la forme MapReduce, la mise à l'échelle de l'application pour qu'elle s'exécute sur des centaines, des milliers, voire des dizaines de milliers de machines dans un cluster n'est qu'un changement de configuration. Cette évolutivité simple est ce qui a incité de nombreux programmeurs à utiliser le modèle MapReduce. **[34]**

Le système de fichiers distribué Hadoop s'exécute généralement sur le même ensemble de machines que le logiciel MapReduce. Lorsque l'infrastructure exécute une tâche sur les nœuds qui stockent également les données, le temps nécessaire pour effectuer les tâches est considérablement réduit.

2.3.4.2.1 Terminologie de base de Hadoop MapReduce

Comme nous l'avons mentionné ci-dessus, MapReduce est une couche de traitement dans un environnement Hadoop. MapReduce travaille sur des tâches liées à une tâche. L'idée est de s'attaquer à une grande demande en la découpant en unités plus petites.

- **JobTracker et TaskTracker :**

  Dans les premiers jours d'Hadoop (version 1), les démons JobTracker et TaskTracker exécutaient des opérations dans MapReduce. À l'époque, un cluster Hadoop ne pouvait prendre en charge que les applications MapReduce.

  Un JobTracker contrôlait la distribution des demandes d'application aux ressources de calcul d'un cluster. Comme il surveillait l'exécution et l'état de MapReduce, il résidait sur un nœud maître.

  Un TaskTracker traitait les demandes provenant du JobTracker. Tous les trackers de tâches étaient répartis sur les nœuds esclaves d'un cluster Hadoop.

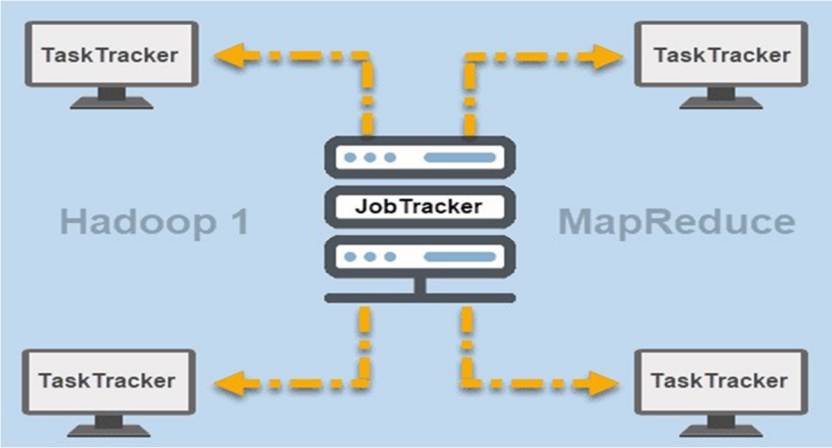

Figure 10 - Terminologie de base de Hadoop MapReduce. **[33]**

- **FIL :**

Plus tard dans Hadoop version 2 et supérieure, YARN est devenu le principal gestionnaire de ressources et de planification. D'où le nom de Yet Another Resource Manager. Yarn a également travaillé avec d'autres frameworks pour le traitement distribué dans un cluster Hadoop.

- **MapReduce Job**

Une tâche MapReduce est l'unité de travail principale dans le processus MapReduce. Il s'agit d'une tâche que les processus Map and Reduce doivent effectuer. Une tâche est divisée en tâches plus petites sur un cluster de machines pour une exécution plus rapide. Les tâches doivent être suffisamment grandes pour justifier le temps de traitement des tâches. Si vous divisez une tâche en segments exceptionnellement petits, le temps total nécessaire pour préparer les fractionnements et créer des tâches peut l'emporter sur le temps nécessaire pour produire la sortie réelle de la tâche

- **Tâche MapReduce**

Les tâches MapReduce comportent deux types de tâches. Une tâche de carte est une instance unique d'une application MapReduce. Ces tâches déterminent les enregistrements à traiter à partir d'un bloc de données. Les données d'entrée sont fractionnées et analysées, en parallèle, sur les ressources de calcul affectées dans un cluster Hadoop. Cette étape d'une tâche MapReduce prépare la sortie de pair < clé, valeur> pour l'étape de réduction.

Une tâche réduite traite une sortie d'une tâche de mappage. Semblable à l'étape de la carte, toutes les tâches de réduction se produisent en même temps et fonctionnent indépendamment. Les données sont agrégées et combinées pour fournir le résultat souhaité. Le résultat final est un ensemble réduit de paires <clé, valeur> que MapReduce, par défaut, stocke dans HDFS. **[32]**

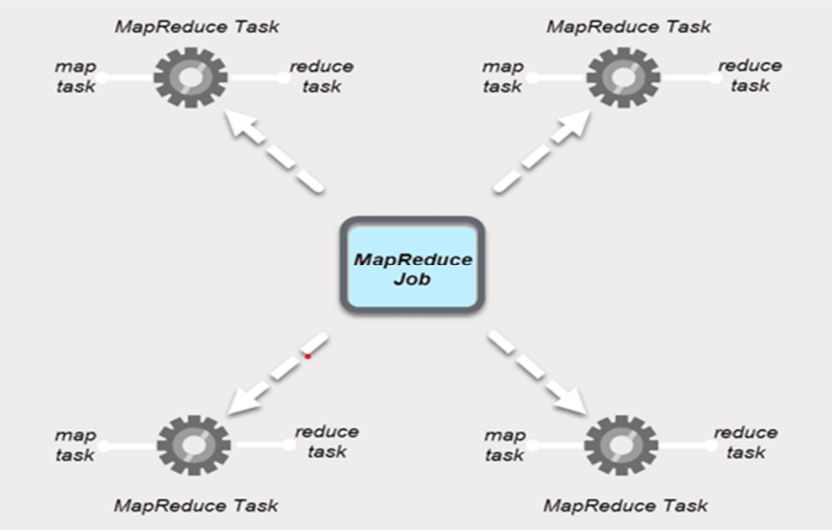

Figure 11 -La Tâche MapReduce **[33]**

2.3.4.2.2 Comment les partitions Hadoop mappent les données d'entrée ?

Le partitionneur est responsable du traitement de la sortie de la carte. Une fois que MapReduce divise les données en morceaux et les affecte aux tâches de mappage, l'infrastructure partitionne les données clé-valeur. Ce processus a lieu avant que la sortie finale de la tâche de mappeur ne soit produite.

MapReduce partitionne et trie la sortie en fonction de la clé. Ici, toutes les valeurs des clés individuelles sont regroupées et le partitionneur crée une liste contenant les valeurs associées à chaque clé. En envoyant toutes les valeurs d'une seule clé au même réducteur, le partitionneur assure une distribution égale de la sortie de carte au réducteur.

Le partitionneur par défaut est bien configuré pour de nombreux cas d'utilisation, mais vous pouvez reconfigurer la façon dont MapReduce partitionne les données.

Si vous utilisez un partitionneur personnalisé, assurez-vous que la taille des données préparées pour chaque réducteur est à peu près la même. Lorsque vous partitionnez les données de

manière inégale, une tâche de réduction peut prendre beaucoup plus de temps. Cela ralentirait l'ensemble du travail MapReduce.**[35]**

2.3.4.3 YARN Et Ses Composants

YARN (Yet Another Resource Negotiator) est la ressource de gestion de cluster par défaut pour Hadoop 2 et Hadoop 3. Dans les versions précédentes d'Hadoop, MapReduce effectuait à la fois le traitement des données et l'allocation des ressources. Au fil du temps, la nécessité de diviser le traitement et la gestion des ressources a conduit au développement de YARN. Le rôle d'allocation des ressources de YARN le place entre la couche de stockage, représentée par HDFS, et le moteur de traitement MapReduce. YARN fournit également une interface générique qui vous permet d'implémenter de nouveaux moteurs de traitement pour différents types de données.

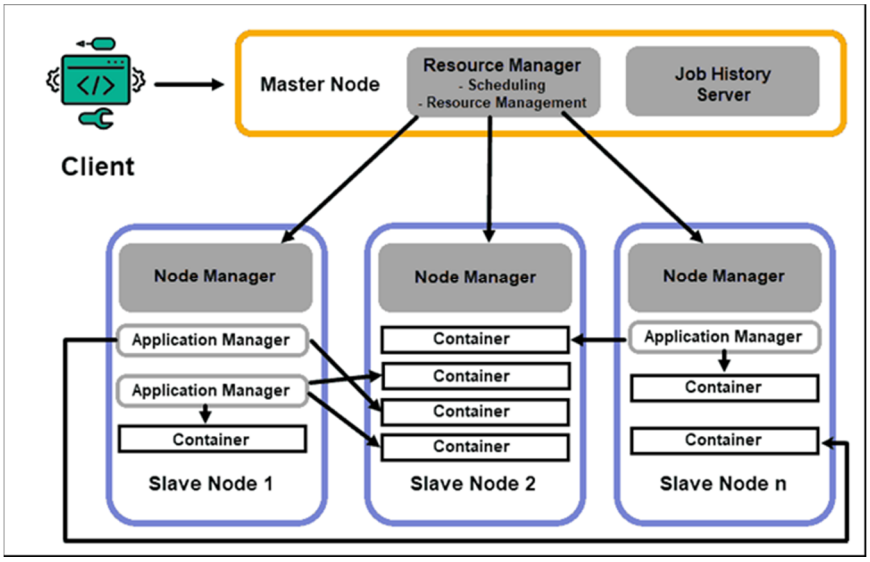

Figure 12 - Gestionnaire de Ressources.**[36]**

- **Gestionnaire de ressources :**

Le démon ResourceManager contrôle toutes les ressources de traitement dans un cluster Hadoop. Son objectif principal est de désigner des ressources pour des applications individuelles situées sur les nœuds esclaves. Il maintient une vue d'ensemble globale des processus en cours et planifiés, gère les demandes de ressources, planifie et affecte les ressources en conséquence. ResourceManager est essentiel au framework Hadoop et doit s'exécuter sur un nœud maître dédié.

Le RM se concentre uniquement sur la planification des charges de travail. Contrairement à MapReduce, il n'a aucun intérêt pour les basculements ou les tâches de traitement individuelles. Cette séparation des tâches dans YARN est ce qui rend Hadoop intrinsèquement évolutif et le transforme en une plate-forme informatique entièrement développée.

- **NodeManager**

  Chaque nœud esclave dispose d'un service de traitement NodeManager et d'un service de stockage DataNode. Ensemble, ils forment l'épine dorsale d'un système distribué Hadoop.

  Le DataNode, comme mentionné précédemment, est un élément de HDFS et est contrôlé par le NameNode. Le NodeManager, de la même manière, agit comme un esclave du ResourceManager. La fonction principale du démon NodeManager est de suivre les données de ressources de traitement sur son nœud esclave et d'envoyer des rapports réguliers au ResourceManager.

- **Conteneurs**

  Les ressources de traitement dans un cluster Hadoop sont toujours déployées dans des conteneurs. Un conteneur contient de la mémoire, des fichiers système et de l'espace de traitement. Un déploiement de conteneur est générique et peut exécuter n'importe quelle ressource personnalisée demandée sur n'importe quel système. Si une quantité demandée de ressources de cluster se situe dans les limites de ce qui est acceptable, le RM approuve et planifie le déploiement de ce conteneur. Les processus de conteneur sur un nœud esclave sont initialement provisionnés, surveillés et suivis par le NodeManager sur ce nœud esclave spécifique.

- **Maître d'application**

  Chaque conteneur d'un nœud esclave possède son maître d'application dédié. Les maîtres d'application sont également déployés dans un conteneur. Même MapReduce dispose d'un maître d'application qui exécute la carte et réduit les tâches.

  Tant qu'il est actif, un maître d'application envoie des messages au gestionnaire de ressources sur son état actuel et l'état de l'application qu'il surveille. Sur la base des informations fournies, Resource Manager planifie des ressources supplémentaires ou les affecte ailleurs dans le cluster si elles ne sont plus nécessaires.

Le maître d'application supervise le cycle de vie complet d'une application, de la demande des conteneurs nécessaires au RM à l'envoi des demandes de location de conteneur au NodeManager.

- **Serveur JobHistory**

  JobHistory Server permet aux utilisateurs de récupérer des informations sur les applications qui ont terminé leur activité. L'API REST assure l'interopérabilité et peut informer dynamiquement les utilisateurs sur les tâches actuelles et terminées servies par le serveur en question.

2.3.4.3.1 Comment fonctionne YARN

Un flux de travail de base pour le déploiement dans YARN démarre lorsqu'une application cliente soumet une demande au ResourceManager.

- ResourceManager demande à un NodeManager de démarrer un maître d'application pour cette demande, qui est ensuite démarrée dans un conteneur.
- Le maître d'application nouvellement créé s'enregistre auprès du RM. Le maître d'application contacte le NameNode HDFS et détermine l'emplacement des blocs de données nécessaires et calcule la quantité de carte et réduit les tâches nécessaires au traitement des données.
- Le maître d'application demande ensuite les ressources nécessaires au RM et continue à communiquer les besoins en ressources tout au long du cycle de vie du conteneur.
- Le RM planifie les ressources avec les demandes de tous les autres maîtres d'application et met en file d'attente leurs demandes. Au fur et à mesure que les ressources deviennent disponibles, le RM les met à la disposition du maître d'application sur un nœud esclave spécifique.
- Application Manager contacte le NodeManager de ce nœud esclave et lui demande de créer un conteneur en fournissant des variables, des jetons d'authentification et la chaîne de commande pour le processus. Sur la base de cette demande, NodeManager crée et démarre le conteneur.
- Le gestionnaire d'applications surveille ensuite le processus et réagit en cas d'échec en redémarrant le processus sur le prochain emplacement disponible. S'il échoue après quatre tentatives différentes, l'ensemble du travail échoue.

Tout au long de ce processus, application Manager répond aux demandes d'état du client.

Une fois toutes les tâches terminées, le maître d'application envoie le résultat à l'application cliente, informe le RM que l'application a terminé sa tâche, se désinscrit du Gestionnaire de ressources et s'arrête et le RM peut également demander au NameNode de mettre fin à un conteneur spécifique pendant le processus en cas de changement de priorité de traitement. **[37]**

2.3.4.4 Hadoop Common

Hadoop Common fait référence à la collection d'utilitaires et de bibliothèques courants qui prennent en charge d'autres modules Hadoop. Il s'agit d'une partie ou d'un module essentiel d'Apache Hadoop Framework, avec Hadoop Distributed File System, Hadoop YARN et Hadoop MapReduce. Comme tous les autres modules, Hadoop Common suppose que les défaillances matérielles sont courantes et qu'elles doivent être automatiquement gérées dans le logiciel par Hadoop Framework.

Hadoop Common est également connu sous le nom de Hadoop Core. Le package Hadoop Common est considéré comme la base/le cœur du framework car il fournit des services essentiels et des processus de base tels que l'abstraction du système d'exploitation sous-jacent et de son système de fichiers. Hadoop Common contient également les fichiers et scripts JAR nécessaires au démarrage de Hadoop. Le package Hadoop Common fournit également le code source et la documentation, ainsi qu'une section de contribution qui inclut différents projets de la communauté Hadoop. **[38]**

2.3.5 Les différents outils de l'écosystème Hadoop

Les entreprises qui souhaitent exploiter leurs données utilisent aujourd'hui Hadoop d'une manière ou d'une autre. Cependant, la valorisation des données a entraîné un foisonnement de problématiques qui nécessitent des réponses technologiques aussi différentes les unes que les autres. Hadoop a beau être le socle technique du Big Data, il n'est pas capable à lui seul de répondre à toutes ces problématiques. À chaque nouvelle problématique, les développeurs sont obligés de coder de nouveaux modules ce qui, en plus d'être frustrant à mettre en production (à cause généralement de l'incompatibilité entre les environnements et la complexité liée aux versions des composants d'Hadoop), réduit la productivité des équipes de

développement, qui passent désormais leur temps plus au débogage du code qu'au développement.

C'est pour résoudre tous ces problèmes qu'un ensemble de technologies, regroupées sous le nom d'écosystème Hadoop, a été développé. L'écosystème Hadoop fournit une collection d'outils et technologies spécialement conçus pour faciliter le développement, le déploiement et le support des solutions Big Data. Actuellement, pas mal de développeurs sont en train de travailler sur l'écosystème Hadoop et offrent leurs travaux en Open Source à la fondation Apache. Apache est aujourd'hui le dépositaire de la majorité de ces technologies. Après incubation des projets qu'elle reçoit des développeurs, elle a inclus dans Hadoop une série spécifique de logiciels et d'outils pour favoriser la productivité des développeurs et pour éviter que les entreprises aient à chaque fois besoin de développer elles-mêmes des outils compatibles avec Hadoop afin de déployer ses solutions sur le cluster. La définition de cet écosystème est importante, car il facilite l'adoption d'Hadoop et permet aux entreprises de surmonter les défis de la valorisation de leurs données.

La figure suivante présente de façon globale l'écosystème Hadoop.

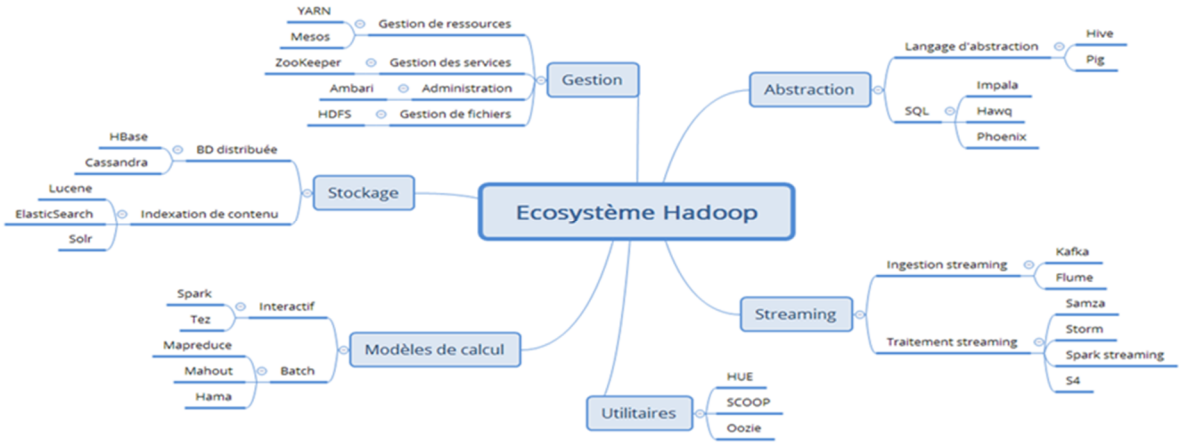

Figure 13 -Ecosystème de Hadoop **[39]**

La configuration de base de l'écosystème Hadoop contient les technologies suivantes : Oozie, Hive, PIG, HBase, Sqoop, Storm, ZooKeeper et Spark. **[40]**

2.4 Apache Spark

2.4.1 Définition

Apache Spark est une infrastructure de traitement parallèle open source qui prend en charge le traitement en mémoire pour améliorer les performances des applications qui analysent le Big Data. Les solutions Big Data sont conçues pour gérer les données trop volumineuses ou complexes pour les bases de données traditionnelles. Spark traite de grandes quantités de données en mémoire, ce qui est beaucoup plus rapide que les alternatives sur disque.

Il est aussi défini qu'il est un Framework Apache destiné à augmenter la vitesse de calcul de Hadoop. Cela aide Hadoop à réduire le temps d'attente entre les requêtes et à minimiser le temps d'attente pour exécuter le programme. **[41]**

2.4.2 Historique

Tout commence en 2009. Spark fut conçu par Matei Zaharia, un informaticien canadien, lors de son doctorat au sein de l'université de Californie à Berkeley. Initialement, son développement est une solution pour accélérer le traitement des systèmes Hadoop. Aujourd'hui il s'agit d'un projet de la fondation Apache. Depuis 2009, plus de 1200 développeurs ont contribué au projet. Certains sont issus de sociétés réputées comme Intel, Facebook, IBM, Netflix…

En 2014, Spark établit officiellement un nouveau record dans le tri à grande échelle. Il remporte le concours Daytona Grey Sort en triant 100 To de données en 23 minutes seulement. Le précédent record du monde était de 72 minutes établi par Yahoo à l'aide d'un cluster Hadoop MapReduce de 2100 nœuds tandis que Spark utilise uniquement 206 nœuds. Cela signifie qu'il a trié les mêmes données trois fois plus rapidement en utilisant dix fois moins de machines.

De plus, bien qu'il n'existe pas de compétition officielle de tri de pétaoctets, Spark va encore plus loin en triant 1 Po de données, ce qui équivaut à 10000 milliards d'enregistrements, sur 190 machines en moins de quatre heures. Il s'agissait de l'un des premiers tris à l'échelle du pétaoctet jamais effectué dans un cloud public. L'obtention de cette référence marque une étape importante pour le projet Spark. Cela prouve que Spark tient

sa promesse de servir de moteur plus rapide et plus évolutif pour le traitement de données de toutes tailles, des Go aux To aux Pb. **[41]**

2.4.3 Scénarios Big Data courants

Vous pouvez envisager une architecture Big Data si vous devez stocker et traiter de grands volumes de données, transformer des données non structurées ou traiter des données de streaming. Spark est un moteur de traitement distribué à usage général qui peut être utilisé pour plusieurs scénarios big data.

2.4.3.1 Extraire, transformer et charger (ETL)

L'extraction, la transformation et le chargement est le processus de collecte de données à partir d'une ou plusieurs sources, de la modification des données et du déplacement des données vers un nouveau magasin de données. Il existe plusieurs façons de transformer des données, notamment :

- Filtrage
- Tri
- Agrégation
- Jonction
- Nettoyage
- Déduplication
- Validation

2.4.3.2 Traitement du flux de données en temps réel

La diffusion en continu, ou en temps réel, des données sont des données en mouvement. Les données de télémétrie provenant d'appareils IoT, de weblogs et de flux de clics sont tous des exemples de données de streaming. Les données en temps réel peuvent être traitées pour fournir des informations utiles, telles que l'analyse géospatiale, la surveillance à distance et la détection des anomalies. Tout comme les données relationnelles, vous pouvez filtrer, agréger et préparer des données de streaming avant de déplacer les données vers un récepteur de sortie. Apache Spark prend en charge le traitement des flux de données en temps réel via Spark Streaming.

2.4.3.3 Traitement par lots

Le traitement par lots est le traitement du Big Data au repos. Vous pouvez filtrer, agréger et préparer des jeux de données très volumineux à l'aide de travaux de longue durée en parallèle.

2.4.3.4 Machine Learning via MLlib

Le Machine Learning est utilisé pour les problèmes analytiques avancés. Votre ordinateur peut utiliser des données existantes pour prévoir ou prédire les comportements futurs, les résultats et les tendances. La bibliothèque Machine Learning d'Apache Spark, MLlib, contient plusieurs algorithmes et utilitaires de Machine Learning.

2.4.3.5 Graph traitement via GraphX

Un graphique est une collection de nœuds connectés par des arêtes. Vous pouvez utiliser une base de données de graphe si vous avez des données hiérarchiques ou des données avec des relations interconnectées. Vous pouvez traiter ces données à l'aide de l'API GraphX d'Apache Spark.

2.4.3.6 SQL et le traitement structuré des données avec Spark SQL

Si vous utilisez des données structurées (mises en forme), vous pouvez utiliser des requêtes SQL dans votre application Spark à l'aide de Spark SQL.

2.4.4 Architecture de Spark

Apache Spark a trois composants principaux : le pilote, les exécuteurs et le gestionnaire de cluster. Les applications Spark s'exécutent en tant qu'ensembles indépendants de processus sur un cluster, coordonnés par le programme pilote.

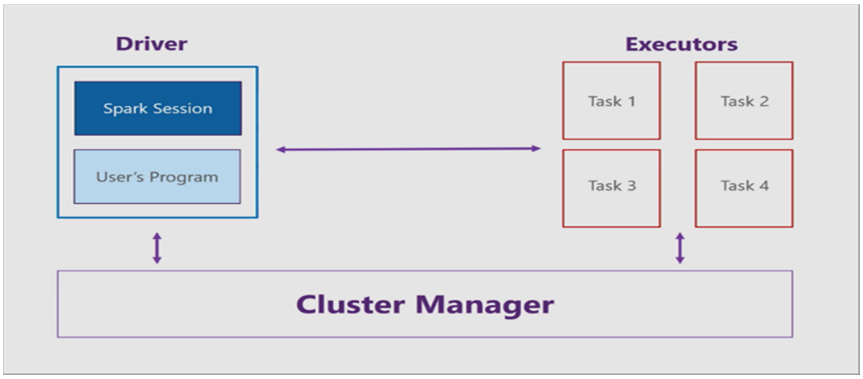

Figure 14 - Architecture d'Apache Spark **[42]**

2.4.4.1 Pilote

Le pilote se compose de votre programme, comme une application console C# et une session Spark. La session Spark prend votre programme et la divise en tâches plus petites gérées par les exécuteurs.

2.4.4.2 Exécuteurs

Chaque exécuteur, ou nœud Worker, reçoit une tâche du pilote et exécute cette tâche. Les exécuteurs résident sur une entité appelée cluster.

2.4.4.3 Gestionnaire de cluster

Le gestionnaire de cluster communique avec le pilote et les exécuteurs pour :

- Gérer l'allocation de ressources
- Gérer la division des programmes
- Gérer l'exécution du programme

2.4.4.4 Support multilingue

Apache Spark prend en charge les langages de programmation suivants :

- Scala
- Python
- Java
- SQL
- R
- Langages .NET (C#/F#)

2.4.4.5 API Spark

Apache Spark prend en charge les API suivantes :

- Spark Scala API
- Spark Java API
- Spark Python API
- Spark R API

- Fonctions intégrées spark SQL

2.4.5 Composants de Spark

2.4.5.1 Spark SQL

Permet d'exécuter des requêtes SQL qui peut être utilisé pour traiter n'importe quelles données, quel que soit leur format d'origine.

2.4.5.2 Spark Streaming

Il offre à son utilisateur un traitement des données en flux.

2.4.5.3 Spark Graph X

Il permet de traiter les informations issues de graphes

2.4.5.4 Spark MLlib

C'est une bibliothèque d'apprentissage automatique, apparue dans la version 1.2 de Spark, qui contient tous les algorithmes et utilitaires d'apprentissage classiques, comme la classification, la régression, le clustering, le filtrage collaboratif et la réduction de dimensions.

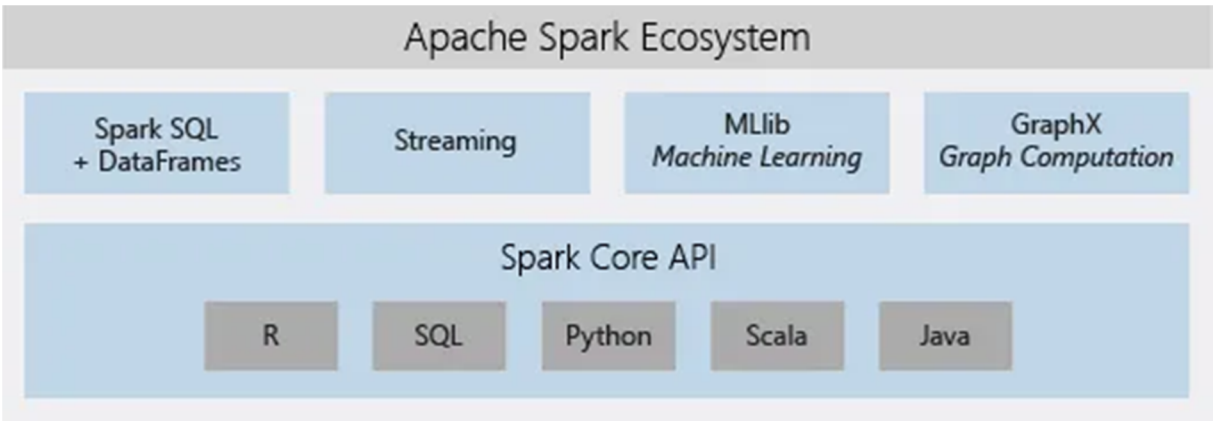

Figure 15 - Ecosysteme de Spark **[43]**

2.4.6 Caractéristiques de Spark

- **Vitesse :** Spark permet aux applications en grappes Hadoop pour exécuter jusqu'à 100x plus rapide dans la mémoire, et 10 fois plus rapide, même lors de l'exécution sur le disque. Spark permet en réduisant nombre de lecture / écriture sur le disque. Il stocke ces données de traitement intermédiaire en mémoire. Il utilise le concept d'un

Dataset Resilient (Distributed RDD), ce qui lui permet de stocker des données de manière transparente sur la mémoire et persistent sur disque seulement est nécessaire. Cela contribue à réduire la majeure partie du disque lire et écrire le principal temps facteurs consommation du traitement des données.

- **Facilité d'utilisation :** Spark permet d'écrire rapidement des applications en Java, Scala ou Python. Cela permet aux développeurs de créer et exécuter leurs applications sur leurs langages de programmation familiers et faciles à construire des applications parallèles. Il est livré avec un ensemble de plus de 80 hauts niveaux intégrés operators Combine SQL, streaming, et des analyses complexes : En plus de simples "carte" et "réduire" les opérations, Spark prend en charge les requêtes SQL, des flux de données, et des analyses complexes telles que l'apprentissage de la machine et des algorithmes de graphes out-of-the-box. Non seulement cela, les utilisateurs peuvent combiner toutes ces capacités de manière transparente dans un seul flux de travail.

- **Exécute Partout :** Spark fonctionne sur Hadoop, Mesos, autonome, ou dans le nuage. Il peut accéder à diverses sources de données, y compris HDFS, Cassandra, HBase, S3.

2.5 Apache Spark vs Hadoop

Depuis plus de 10 ans, Hadoop est considéré comme la principale technologie de traitement de données Big Data. Il s'agit effectivement d'une solution de choix pour le traitement de larges ensembles de données. Pour les calculs « one-pass », MapReduce est effectivement très efficace, mais se retrouve moins pratique pour les cas d'usage nécessitant des calculs multi-pass et des algorithmes. Pour cause, chaque étape du traitement de données est décomposée entre une phase Map et une phase Reduce.

Entre chaque étape, les données doivent être stockées dans le Système de Fichier Distribué avant que la prochaine étape ne puisse débuter. Dans la pratique, cette approche se révèle très lente. De plus, les solutions Hadoop incluent généralement des clusters difficiles à configurer et à gérer. Plusieurs outils doivent également être intégrés pour les différents cas d'usage Big Data. Pour le Machine Learning, il faudra par exemple utiliser Mahout. Pour le traitement de flux de données, il sera nécessaire d'intégrer Storm.

De son côté, Apache Spark permet aux programmeurs de développer des pipelines de données multi-step complexes en utilisant des patterns DAG. Spark prend également en

charge le partage de données in-memory à travers les DAGs, permettant d'effectuer différentes tâches avec les mêmes données. Il est exécuté à partir d'une infrastructure HDFS existante pour fournir des fonctionnalités améliorées et additionnelles. Il permet de déployer des applications sur un cluster Hadoop V1 avec SIMR, un cluster Hadoop V2 YARN ou sur Apache Mesos.

Plutôt qu'un remplacement d'Hadoop, il peut être considéré comme une alternative Spark à Hadoop MapReduce. Spark n'a pas pour vocation de remplacer Hadoop, mais de fournir une solution unifiée et compréhensible pour gérer différents cas d'usage Big Data. **[44]**

2.5.1 Apache Spark peut- il s'exécuter sans Apache Hadoop

Hadoop comprend un composant de stockage, connu sous le nom de HDFS (Hadoop Distributed File System), et un outil de traitement appelé MapReduce. De fait, il n'est pas nécessaire de faire appel à Spark pour traiter ses données Hadoop. Et inversement, il est possible d'utiliser Spark sans faire intervenir Hadoop. Spark n'a pas de système de gestion de fichiers propre, ce qui veut dire qu'il faut lui associer un système de fichiers - soit HDFS, soit celui d'une autre plate-forme de données dans le cloud. Donc il est possible d'utiliser Hadoop indépendamment de Spark et réciproquement. **[44]**

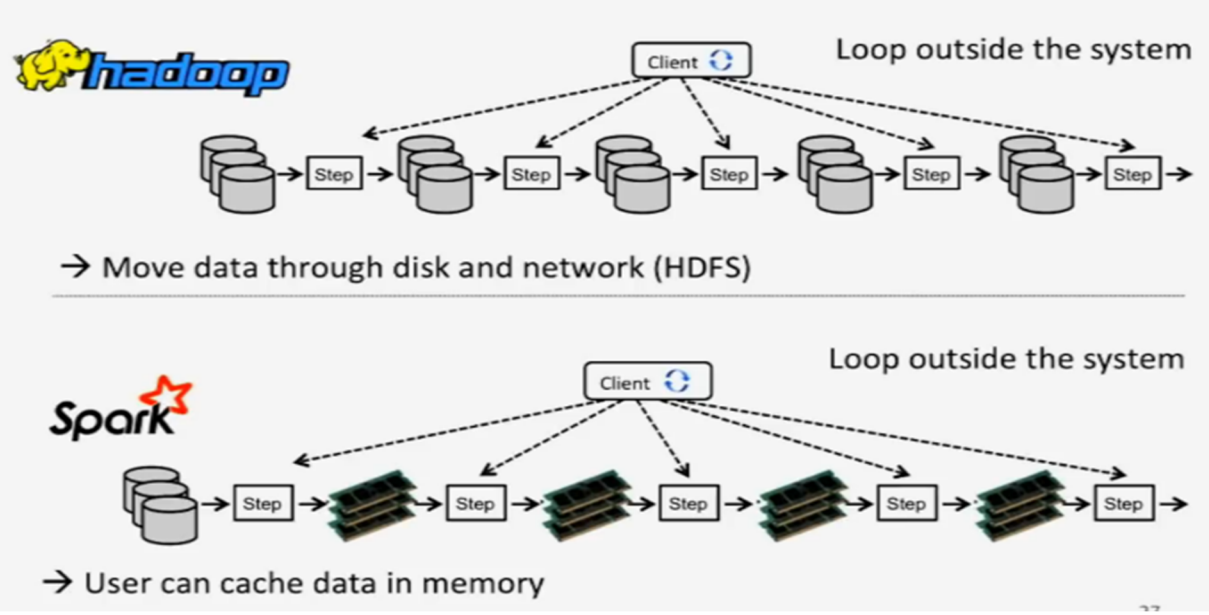

Figure 16 - Hadoop et Spark **[44]**

Néanmoins, Spark a été conçu pour Hadoop, et la plupart des gens s'accordent pour dire qu'ils fonctionnent mieux ensemble.

La figure 16 illustre que Hadoop ne travaille qu'en mode lots avec MapReduce alors que Spark fait du temps réel en in-memory

2.6 Conclusion

Nous avons présenté dans ce chapitre les Framework Hadoop et Spark et leurs objectifs. On a vu les différents composants d'Hadoop, principalement, HDFS et le modèle de programmation MapReduce, nous avons vu aussi comment le Framework Apache Spark, avec son API standard, nous aide en matière de traitement et d'analyse de données. Nous avons aussi vu comment Spark se positionne par rapport aux implémentations MapReduce traditionnelles comme Apache Hadoop. Spark s'appuie sur le même système de stockage de fichiers qu'Hadoop.

Enfin nous avons terminez avec une étude comparative entres les deux Framework concourants Spark et Hadoop, on peut dire qu'ils ont une relation symbiotique avec l'autre. Hadoop fournit des fonctionnalités qui Spark ne possède pas, comme un système de fichiers distribué et Spark fournit en temps réel, le traitement en mémoire pour les ensembles de données qui en ont besoin.

Chapitre 3 :

Mise en oeuvre, Test et Evaluation

3 Chapitre 3 : Mise en œuvre, Test et Evaluation

3.1 Introduction

Dans ce chapitre, nous allons aborder la partie pratique de notre projet qui consiste à annoter et à traiter des données (images médicales) dans un cluster Hadoop ; ainsi la conception et la mise en œuvre complète de notre environnement distribué.

Nous définirons les différentes étapes d'installation et de configuration de notre cluster Hadoop et les différents tests effectués sur des tâches MapReduce.

3.2 Qu'allons-nous installer pour créer le cluster Hadoop Multi-Node

- Java 8.
- SSH.
- PDSH.
- Utilisation de trois machines virtuelles (VM) avec un système d'exploitation Linux Ubuntu 20.04.
- Installation d'Hadoop à partir de la distribution Apache qui est considérée actuellement comme la distribution la plus utilisée dans les entreprises, du fait qu'elle propose une version open source complète qui utilise les principaux composants d'Hadoop (HDFS, MapReduce, Hbase, Pig, Zookeeper).
- Installation de Spark à partir de la distribution Apache qui est considérée actuellement comme la distribution la plus utilisée dans les entreprises, du fait qu'elle propose une version open source complète qui utilise les principaux composants de Spark (Spark Core, Spark SQL, Spark Streaming, Spark MLLib, GraphX, SparkR).
- PySpark.
- Python3.
- OpenCv.

3.3 L'environnement de travail

Dans la section suivante, on va décrire les outils et systèmes composant notre environnement de travail.

3.3.1 VMware Workstation

3.3.1.1 Définition

VMware Workstation est un logiciel de machine virtuelle utilisé par les ordinateurs x86 et x86-64 pour exécuter plusieurs systèmes d'exploitation sur un seul ordinateur hôte physique. Chaque machine virtuelle peut exécuter simultanément une seule instance de n'importe quel système d'exploitation (Microsoft, Linux, etc.). **[45]**

3.3.1.2 Historique

VMware a été créé en 1998 et a produit de nombreux produits pour la virtualisation. VMware Workstation a été lancé par VMware en 2001.

3.3.1.3 Avantages

VMware Workstation prend fortement en charge la compatibilité matérielle et fonctionne comme un pont entre l'hôte et la machine virtuelle pour toutes sortes de ressources matérielles, y compris les disques durs, les périphériques USB et les CD-ROM. Tous les pilotes de périphériques sont installés via la machine hôte. VMware Workstation permet l'installation de plusieurs instances de différents systèmes d'exploitation, y compris les systèmes d'exploitation client et serveur. Il aide les administrateurs réseau ou système à vérifier, tester et vérifier l'environnement client-serveur. L'administrateur peut également basculer entre différentes machines virtuelles en même temps.

VMware Workstation a ses limites, notamment la prise en charge matérielle, les problèmes de système d'exploitation et les obstacles liés aux protocoles réseau.

3.3.2 Ubuntu 20.04

3.3.2.1 Définition

Ubuntu est probablement la plus populaire des distributions Linux avec Debian. C'est un système d'exploitation libre et open-source. Il fonctionne sur un ordinateur physique ou dans une machine virtualisée et se divise en 3 éditions : core, server et desktop. Le système d'exploitation se trouve donc sur une multitude d'appareils allant de l'ordinateur personnel au serveur internet.

Figure 17 – Ubuntu **[46]**

3.3.2.2 Historique

La distribution de base a eu trois périodes de diffusion : La période 1993-2004 : stabilisation du noyau Linux et apparition des variantes (Arch, Debian...)

Ubuntu voit le jour en 2004. Avant de devenir le système d'exploitation basé sur l'entraide que l'on connait aujourd'hui, le projet est d'abord celui d'un seul homme : Mark Shuttleworth. Cet entrepreneur sud-africain fonde l'entreprise Canonical en 2004 et se base sur Debian pour créer un système d'exploitation libre plus accessible aux utilisateurs novices. La première version d'Ubuntu verra le jour en octobre de la même année. Un an plus tard, en 2005, Shuttleworth créera la Ubuntu Foundation, une organisation à but non lucratif ayant pour but d'assurer le développement du système d'exploitation et ses mises à jour dans le futur. La période 2005-2017 : Debian sert de base à sa variante : Ubuntu.Unity est développée en vue de la convergence desktop-phone dans sa version 8. La période 2017-actuelle : Unity est abandonnée au profit de Gnome.

Son nom trouve son origine en Afrique du Sud et il est issu du mot bantoue « ubuntu » qui pourrait être traduit par « je suis ce que je suis grâce à ce que nous sommes tous ». D'après les développeurs, ce nom décrit parfaitement l'esprit collaboratif de la communauté autour du projet.**[47]**

3.3.2.3 Avantages

Le système d'exploitation a pour base le kernel Linux et est compatible avec les architectures IA-32, x86-64, ARM64, ARMhf, ppc64le, s390x. Son développement, bien que basé sur une grande communauté de développeurs bénévoles, est assuré par la compagnie britannique Canonical. C'est notamment elle qui offre le support sur les versions d'Ubuntu pour les entreprises et les autres services Premium sur lesquelles l'entreprise fait un bénéfice, ce qui lui permet donc de continuer ses opérations.

Aujourd'hui, Ubuntu réunit des millions de développeurs et d'utilisateurs qui y trouvent tous leur compte. L'OS est en effet très « customisable » par sa nature open-source et offre en plus de cela des mises à jour de sécurité quasi quotidiennement. Il présente aussi l'avantage d'être peu gourmand en ressources.

3.3.2.4 Choix d'Ubuntu

Le système d'exploitation open source Ubuntu offre plusieurs opportunités, parmi lesquelles, on trouve :

- Facile à installer : Un environnement graphique clair vous guide durant cette phase délicate et tout est prêt en 40 mn environ, y compris l'installation des dernières mises à jour de sécurité et corrections de paquets logiciels.
- Moins gourmand et plus léger : Après installation du système et des applications, Ubuntu n'occupe que 6 Go sur mon disque dur, contre le double ou le triple pour un système tel que Windows Seven. Ubuntu est également moins exigeant en termes de puissance et fonctionnera donc parfaitement sur un modèle d'ordinateur un peu ancien.
- Est un système multi utilisateur.
- Pas de menaces et n'a pas besoin d'antivirus, il est sécurisé.
- Est multitâche, gère plusieurs tâches en même temps comme Windows.
- Ubuntu vous propose de nombreux logiciels – Firefox, OpenOffice, logiciels de jeux, de formations, bureautiques, de sciences, d'éducations etc...
- Il facilite la communication en ligne – Yahoo, Google, Skype.
- Gérer parfaitement comme tout système d'exploitation les images et vidéos. **[47]**

3.4 Filtre de Canny

3.4.1 Définition

Le filtre de Canny (ou détecteur de Canny) est utilisé en traitement d'images pour la détection des contours. L'algorithme a été conçu par John Canny en 1986 pour être optimal suivant trois critères clairement explicités : **[48]**

- Premier critère : Bonne détection : faible taux d'erreur dans la signalisation des contours,
- Deuxième critère : Bonne localisation : minimisation des distances entre les contours détectés et les contours réels,

- Troisième critère : Clarté de la réponse : une seule réponse par contour et pas de faux positifs

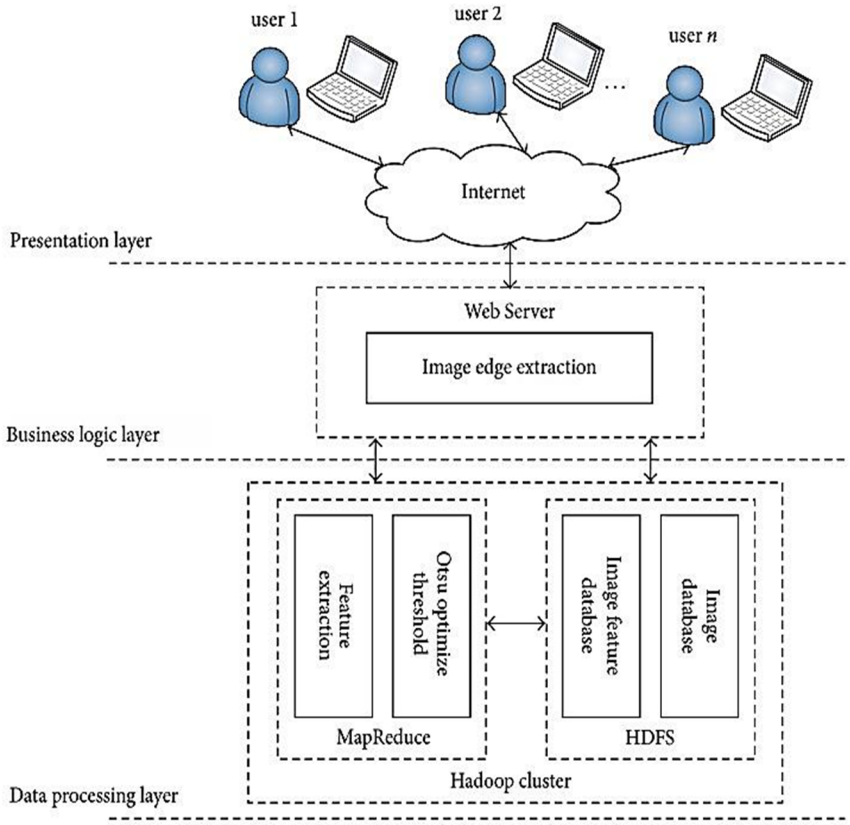

Figure 18 - Architecture pour l'extraction massive des bords d'images. **[49]**

3.4.2 Développement du filtre de Canny

L'Algorithme de filtre de Canny peut être décomposé en 5 étapes :

3.4.2.1 Réduction du bruit

La première étape est de réduire le bruit de l'image originale avant d'en détecter les contours. Ceci permet d'éliminer les pixels isolés qui pourraient induire de fortes réponses lors du calcul du gradient, conduisant ainsi à de faux positifs.

Un filtrage gaussien 2D est utilisé, dont voici l'opérateur de convolution :

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{.....(1)}$$

Usuellement, un filtre est de taille plus réduite que l'image filtrée. Plus le masque est grand, moins le détecteur est sensible au bruit et plus l'erreur de localisation grandit.

3.4.2.2 Gradient d'intensité

Après le filtrage, l'étape suivante est d'appliquer un gradient qui retourne l'intensité des contours. L'opérateur utilisé permet de calculer le gradient suivant les directions X et Y, il est composé de deux masques de convolution, un de dimension 3 × 1 et l'autre 1 × 3 :

$$G_x = [-1 \quad 0 \quad 1] \quad ; \quad G_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad \text{.....(2)}$$

La valeur du gradient en un point est approximée par la formule :

$$|G| = |G_x| + |G_y| \quad \text{.....(3)}$$

Et la valeur exacte est :

$$|G| = \sqrt{G_x^2 + G_y^2} \quad \text{.....(4)}$$

3.4.2.3 Direction des contours

Les orientations des contours sont déterminées par la formule :

$$\theta = \pm \arctan\left(\frac{G_y}{G_x}\right) \quad \text{.....(5)}$$

Nous obtenons finalement une carte des gradients d'intensité en chaque point de l'image accompagnée des directions des contours.

3.4.2.4 Suppression des non-maxima

La carte des gradients obtenue précédemment fournit une intensité en chaque point de l'image. Une forte intensité indique une forte probabilité de présence d'un contour. Toutefois, cette intensité ne suffit pas à décider si un point correspond à un contour ou non. Seuls les points

correspondant à des maximas locaux sont considérés comme correspondants à des contours, et sont conservés pour la prochaine étape de la détection.

Un maximum local est présent sur les extrema du gradient, c'est-à-dire là où sa dérivée selon les lignes de champs du gradient s'annule

3.4.2.5 Seuillage des contours

La différenciation des contours sur la carte générée se fait par seuillage à hystérésis.

Cela nécessite deux seuils, un haut et un bas ; qui seront comparés à l'intensité du gradient de chaque point. Le critère de décision est le suivant. Pour chaque point, si l'intensité de son gradient est :

-Inférieur au seuil bas, le point est rejeté ;

-Supérieur au seuil haut, le point est accepté comme formant un contour ;

-Entre le seuil bas et le seuil haut, le point est accepté s'il est connecté à un point déjà accepté.

Une fois ceci réalisé, l'image obtenue est binaire avec d'un côté les pixels appartenant aux contours et les autres. **[48]**

3.4.3 Conception et réalisation de l'algorithme

Le flux de travail des fonctions mapper() et reducer() est illustré dans la figure 20 MapReduce transforme les fractionnements en paires clé-valeur (<key1, value1>) à l'aide de la méthode recordReader de SequenceFileInputFormat, dans laquelle key1 est le nom du chemin de l'image et value1 est un pointeur vers les données d'image. La fonction mapper() renvoie une autre paire clé-valeur (<clé2, valeur2>). La fonctionnalité MapReduce fusionne toutes les valeurs qui ont la même clé pour générer <key2, (value2 list)>, qui sert d'entrée aux fonctions reducer(). Après le traitement de reducer(), la paire clé-valeur de sortie (<key3, value3>) est écrite dans le système de fichiers HDFS par les fonctions RecordWriter dans la catégorie personnalisée ImageOutputFormat. **[50]**

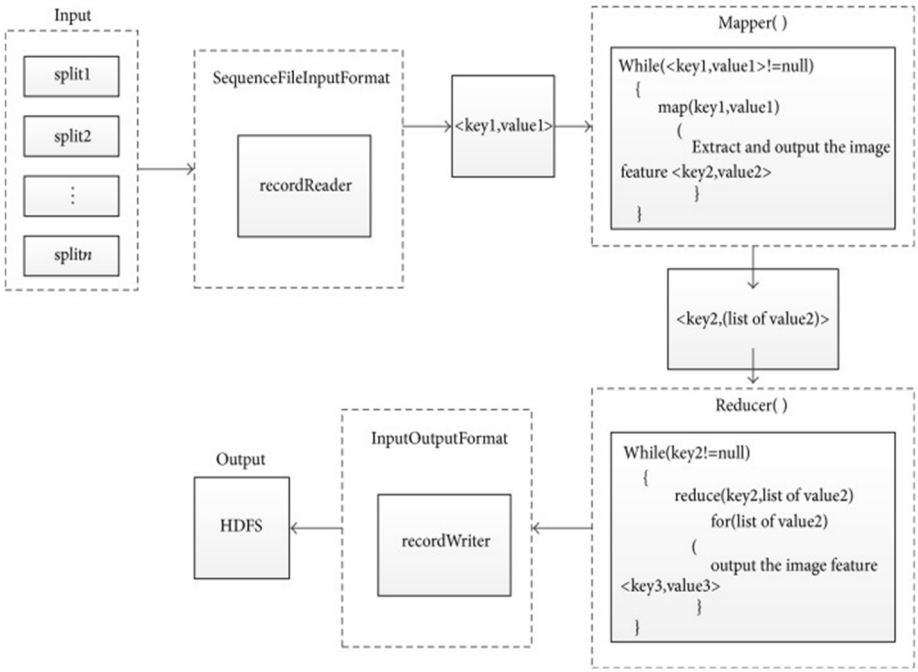

Figure 19 - Le workflow des fonctions mapper() et reducer(). **[50]**

3.5 Les étapes d'installation et de configuration de notre système

3.5.1 La configuration de notre réseau

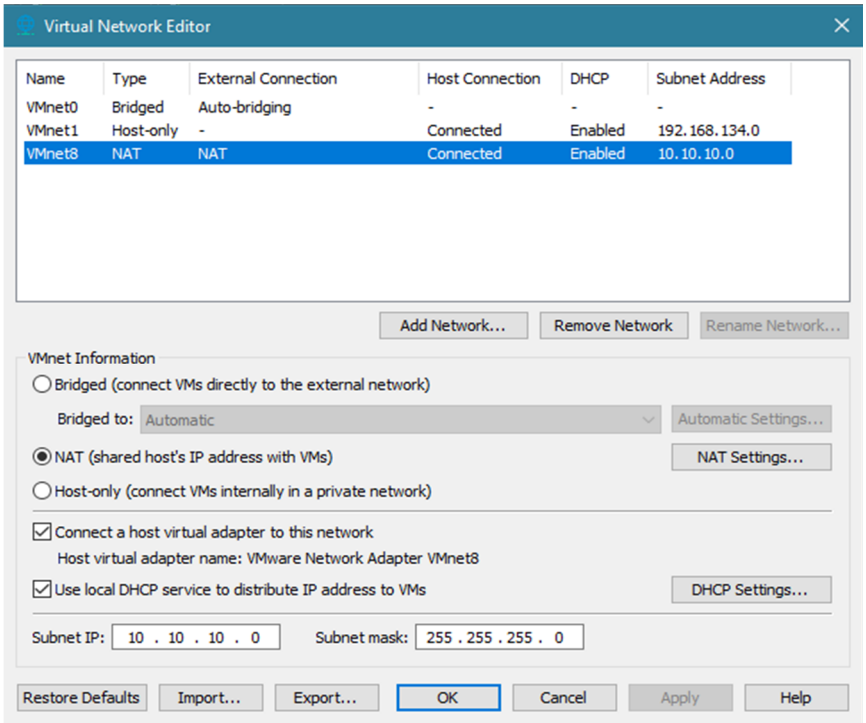

Figure 20 - Configuration des adresses IP et DHCP

3.5.2 Installation de ssh et pdsh

Avec la commande : sudo apt install ssh

Avec la commande : sudo apt install pdsh

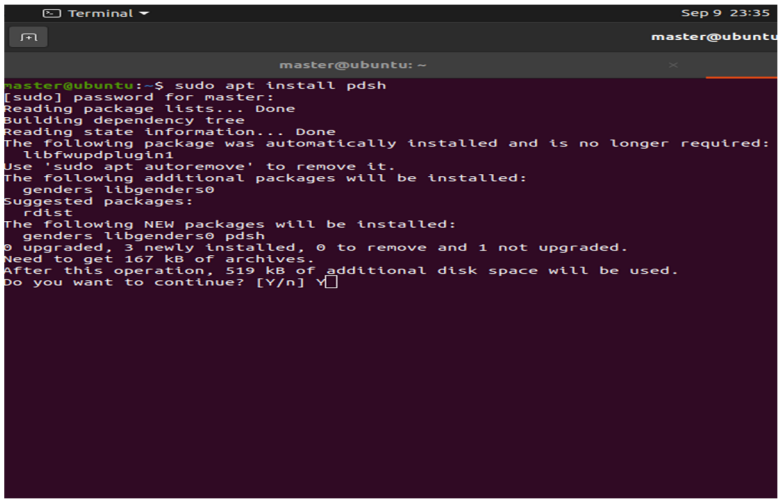

Figure 21 - Installation du ssh et pdsh

3.5.3 Définir l'environnement pdsh sur ssh

Ouvrir le fichier Bashrc avec nano : sudo nano .bashrc

Ajouter à la fin du fichier : export PDSH_RCMD_TYPE=ssh

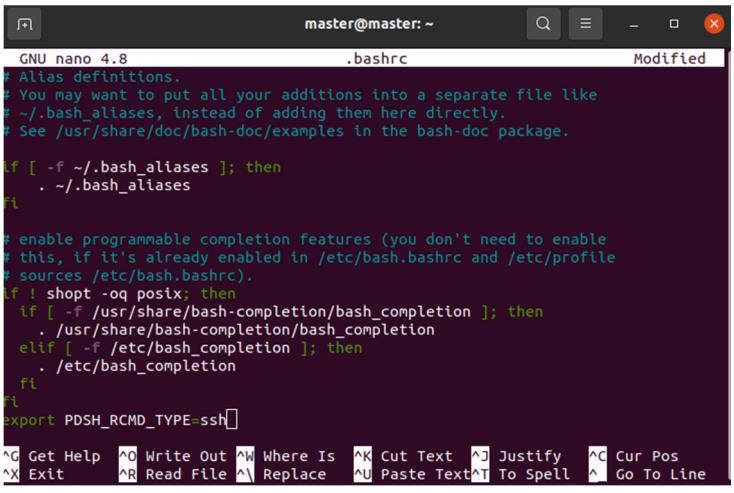

Figure 22 - Configuration de l'environnement

3.5.4 Générer la clé ssh

Avec la commande : ssh-keygen -t rsa -P ""

Appuyons sur Entrée lorsque nous sommes invités à choisir le fichier de stockage.

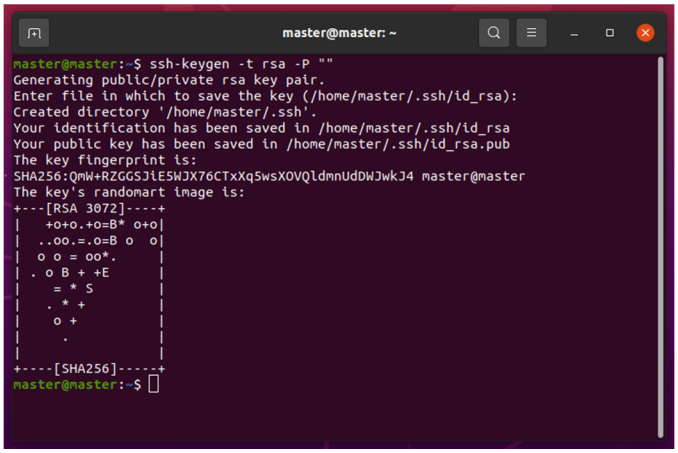

Figure 23 - Génération de la clé ssh.

3.5.5 Cloner la clé dans les fichiers authorized_keys

Pour donner les bonnes autorisations à notre clé ssh, vous devons créer une copie sur les fichiers authorized_keys :

Avec la commande : cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

Assurons-nous que tout est bien configuré, en faisant un ssh à notre machine.

Avec la commande : ssh localhost

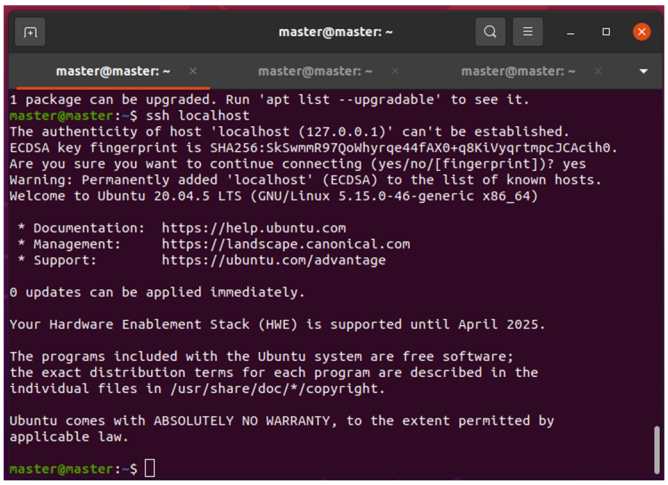

Figure 24 - Connexion avec ssh.

3.5.6 Installation de Java 8

Pour exécuter Hadoop, nous devons installer Java 8 sur notre machine. Pour ce faire, utilisons la commande suivante : sudo apt install openjdk-8-jdk

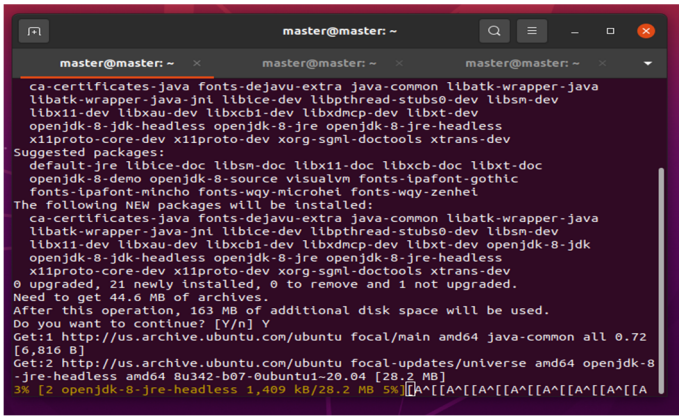

Figure 25 - Installation du JAVA.

Nous vérifions si Java est installé avec la commande suivante :

java -version

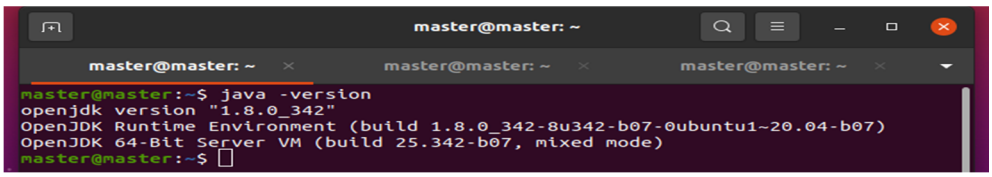

Figure 26 - La version de JAVA.

3.5.7 Installer Hadoop

Téléchargeons d'abord le fichier tar qui contient Hadoop avec la commande suivante :
sudo wget -P ~ https://mirrors.sonic.net/apache/hadoop/common/hadoop-3.2.3/hadoop-3.2.3.tar.gz

Figure 27 - Téléchargement de hadoop

Une fois le téléchargement terminé, décompressons-le : tar xzf hadoop-3.2.3.tar.gz
Et déplaçons-le dans un dossier qui s'appellera hadoop (c'est une solution pratique, ne changeons pas la mécanique du reste) :

mv hadoop-3.2.3 hadoop

3.5.8 Configurer Hadoop

Commençons à configurer le chemin Java sur l'environnement virtuel de Hadoop :

sudo nano ~/hadoop/etc/hadoop/hadoop-env.sh

Recherchons ensuite la ligne de Java_Home et remplaçons-la par :

Export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/

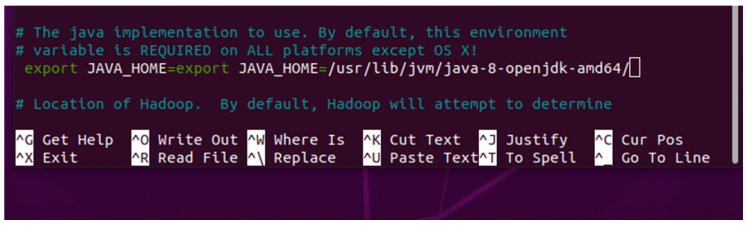

Figure 28 - Configuration du chemin de JAVA

Enregistrons le fichier avec Ctrl + O puis quittez avec Ctrl + X.

3.5.9 Déplacez le répertoire hadoop vers notre fichier utilisateur local

Déplaçons-le avec la commande suivante : sudo mv hadoop /usr/local/hadoop

3.5.10 Configurer le chemin Hadoop

Pour configurer le chemin hadoop sur l'environnement de la machine, ouvrons le fichier d'environnement avec :

sudo nano /etc/environment

Et puis remplaçons tout par :

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/local/hadoop/bin:/usr/local/hadoop/sbin" JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"

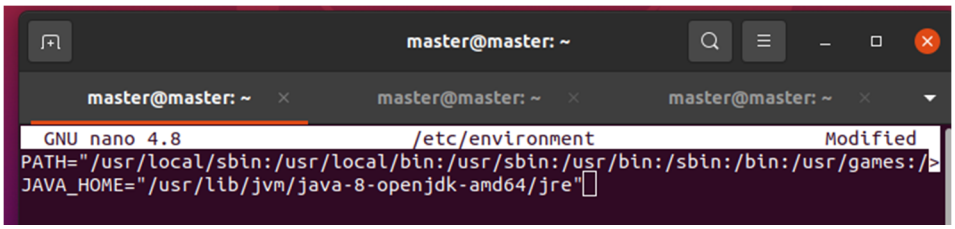

Figure 29 - Configuration du chemin HADOOP

Enregistreons le fichier avec Ctrl + O puis quittez avec Ctrl + X.

3.5.11 Créer un utilisateur spécifique pour Hadoop

Commençons par créer un nouvel utilisateur :

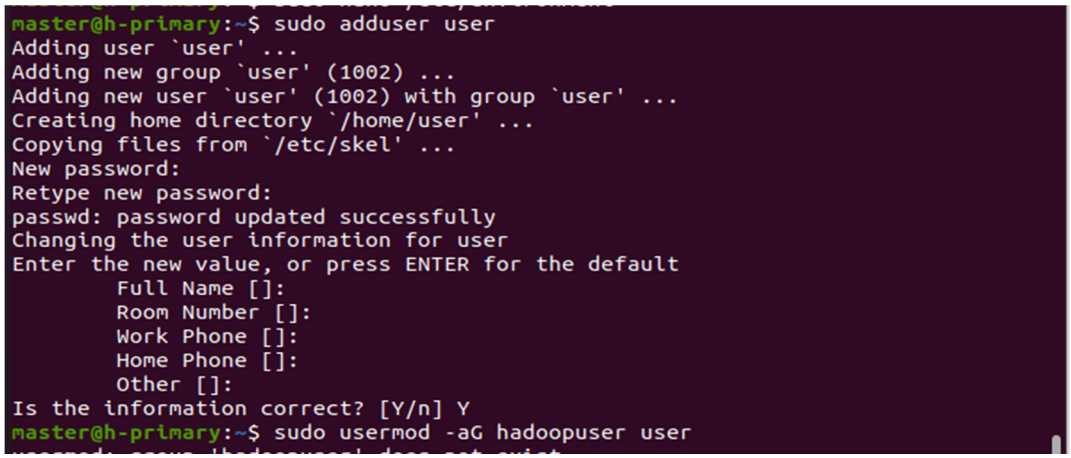

Figure 30 - Creation d'un utilisateur.

Remarque : Avant de faire cette étape j'ai changé le nom de ma machine du master au h-primary par la commande : sudo nano /etc/hostname. Après la commande sudo reboot pour que la machine se redémarre et avoir le nouveau nom.

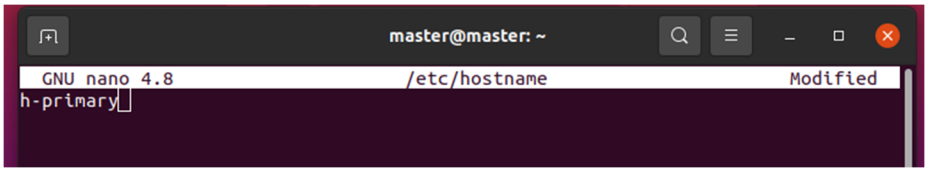

Figure 31 - Changement du nom

Nous devons maintenant donner à cet utilisateur les autorisations nécessaires pour travailler dans le dossier de hadoop.

sudo usermod -aG user user

sudo chown user:root -R /usr/local/hadoop/

sudo chmod g+rwx -R /usr/local/hadoop/

sudo adduser h-user sudo

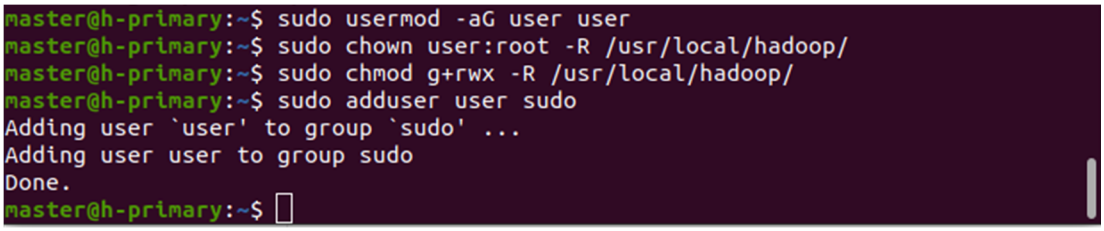

Figure 32 - configuration des autorisations

3.5.12 Cloner la machine principale afin de créer deux machines secondaires

Créons deux clones complets de la machine principale :

Assurons-nous maintenant que toutes les machines ont des adresses Mac différentes afin d'avoir des adresses IP différentes :

3.5.13 Modifier les hostname

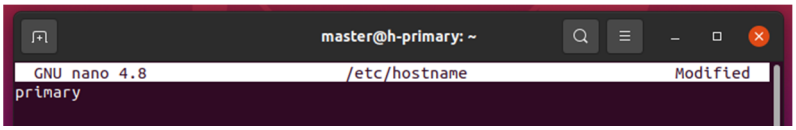

Figure 33 - Modification du hostname (master)

Faisons de même pour les machines secondaires :

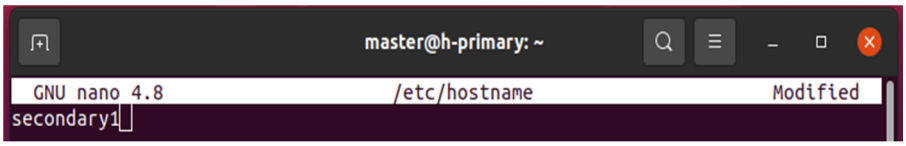

Figure 34 - Modification du hostname (slave1)

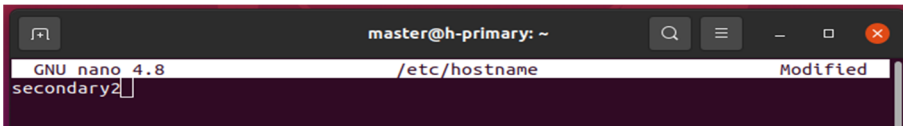

Figure 35 - Modification du hostname (slave2)

Maintenant, redémarrons toutes les machines : reboot

3.5.14 Identifier l'IP de la machine

Pour connaître l'utilisation IP de la machine : ip addr

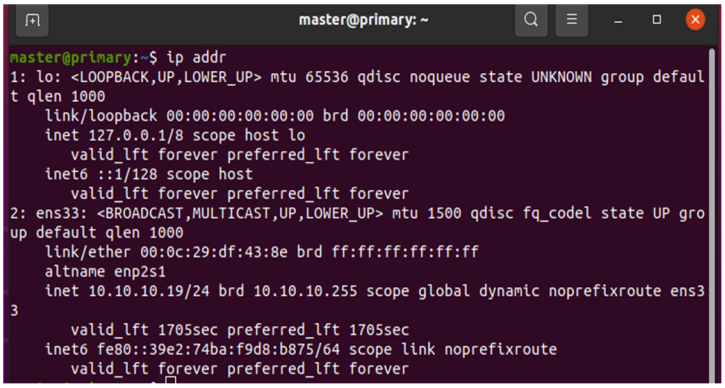

Figure 36 - Adresse IP master

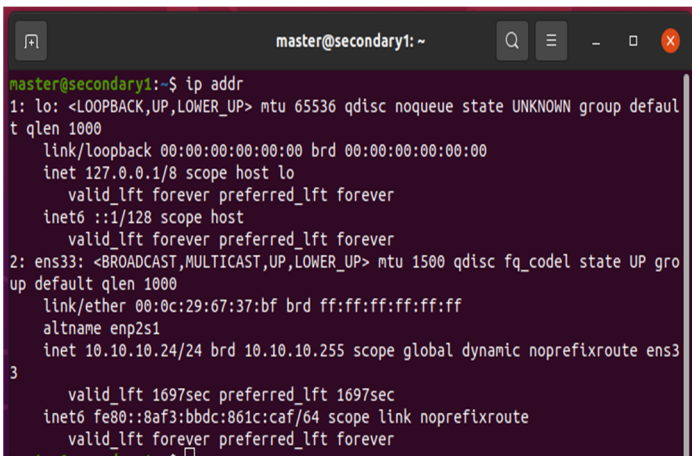

Figure 38 - Adresse IP slave1

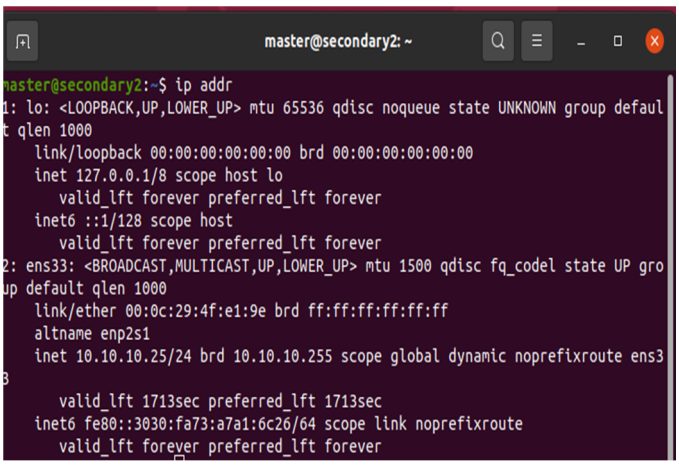

Figure 37 - Adresse IP slave2

Modifions maintenant le fichier hosts sur toutes les machines :

Avec la commande : sudo nano /etc/hosts

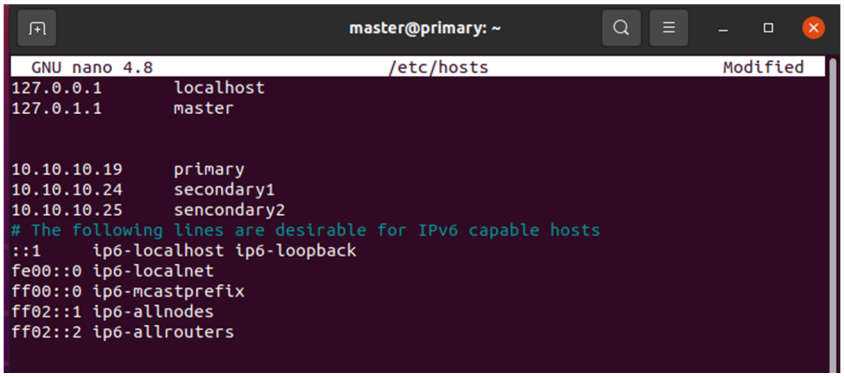

Figure 39 - Modification du fichier hosts

3.5.15 Configurer ssh sur Primary avec notre utilisateur

Démarrer pour changer d'utilisateur : su - user

Figure 40 - Connection avec user

Nous devons maintenant générer une clé ssh pour cet utilisateur avec la commande : ssh-keygen -t rsa

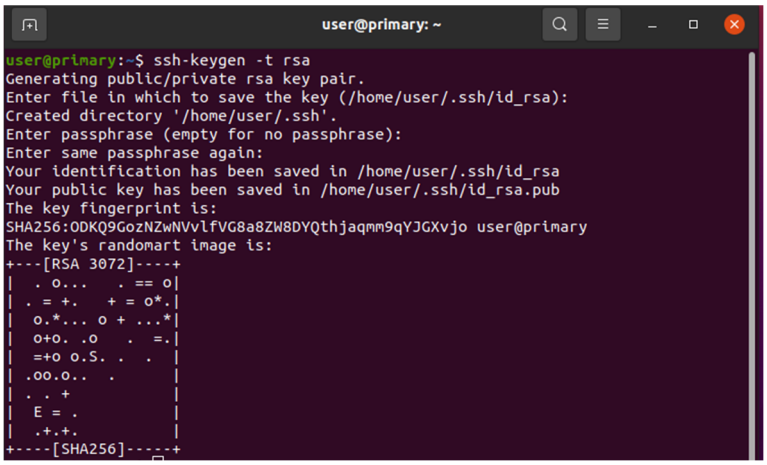

Figure 41 - Generation du cle ssh

3.5.16 Copier la clé ssh sur nos machines secondaires

Copions la clé déjà générée sur toutes les machines :

ssh-copy-id user@primary

ssh-copy-id user@secondary1

ssh-copy-id user@secondary2

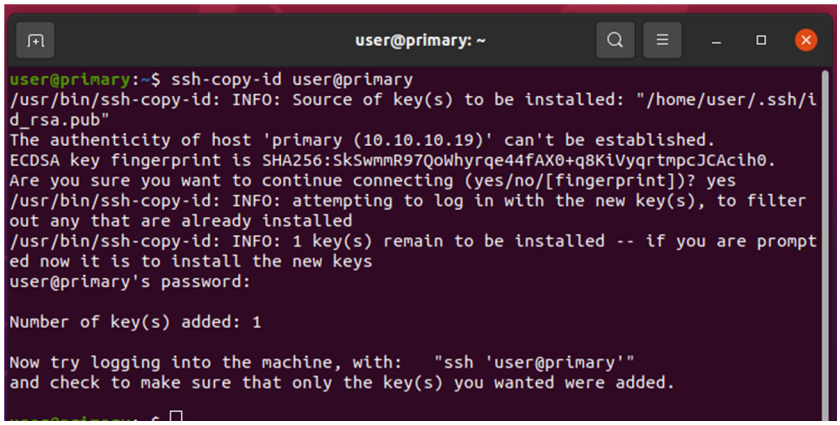

Figure 42 - clone du cle ssh (master)

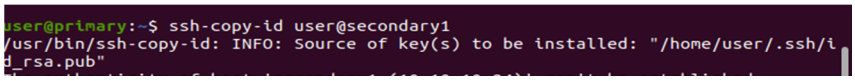

Figure 43 - clone du cle ssh (slave1)

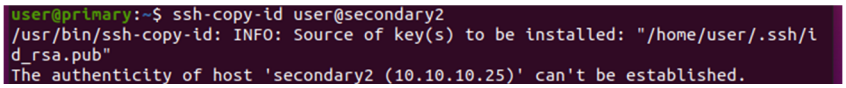

Figure 44 - clonage du cle ssh (slave2)

3.5.17 Configurer le port de service Hadoop

Modifier les configurations de port hadoop : (uniquement sur la machine principale)

sudo nano /usr/local/hadoop/etc/hadoop/core-site.xml

Et puis ajoutons à la configuration du fichier :

<property>

<name>fs.defaultFS</name>

<value>hdfs://primary:9000</value>

</property>

Figure 45 - Configuration de port de service Hadoop

3.5.18 Configuration du système HDFS

Modifier les configurations HDFS : (uniquement sur la machine principale).

sudo nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml

Et puis ajoutons à la configuration du fichier :

<property>

<name>dfs.namenode.name.dir</name><value>/usr/local/hadoop/data/nameNode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name><value>/usr/local/hadoop/data/dataNode</value>

</property>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

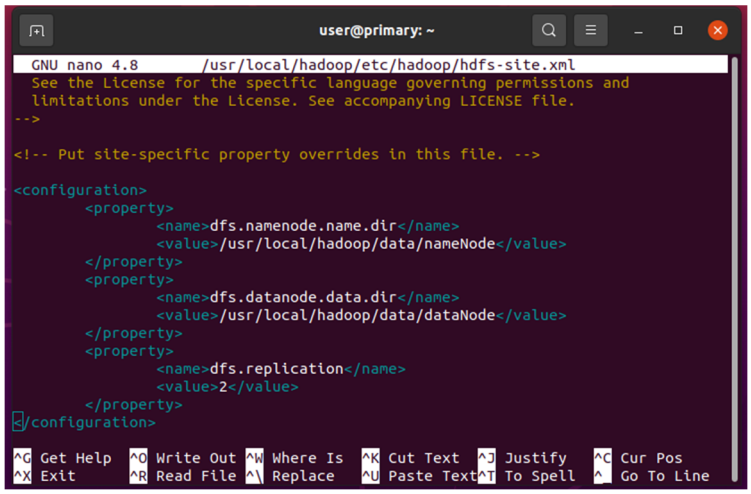

Figure 46 - Configuration du système HDFS

3.5.19 Identifier les workers

Ajouter les machines secondaires au fichier workers : (uniquement sur la machine principale).

sudo nano /usr/local/hadoop/etc/hadoop/workers

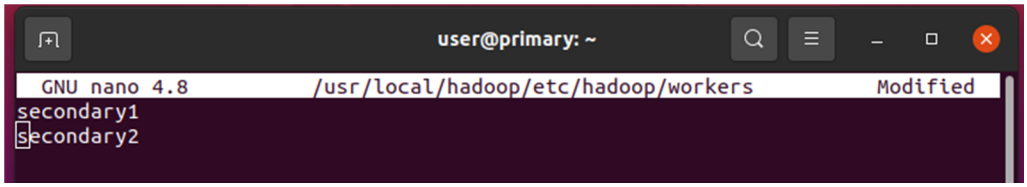

Figure 47 - Workers

3.5.20 Copier les configurations dans des machines secondaires

Nous devons nous assurer que toutes les configurations que nous venons de modifier vont sur toutes les machines, pour ce faire, exécutons les commandes suivantes :

scp /usr/local/hadoop/etc/hadoop/* secondary1:/usr/local/hadoop/etc/hadoop/

scp /usr/local/hadoop/etc/hadoop/* secondary2:/usr/local/hadoop/etc/hadoop/

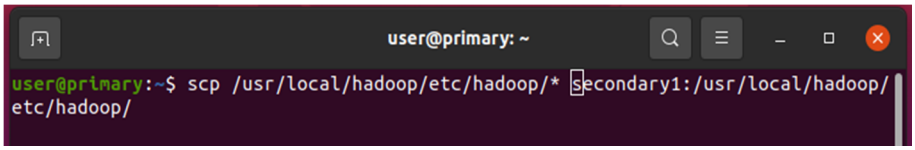

Figure 48 - Copier les configurations dans slave1

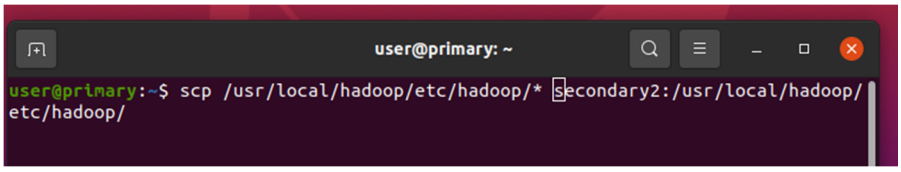

Figure 49 - Copier les configurations dans slave2

3.5.21 Formatage et démarrage du système HDFS (uniquement la machine principale)

Commençons pour nous assurons que toutes les modifications sont appliquées :

Avec la commande : source /etc/environment

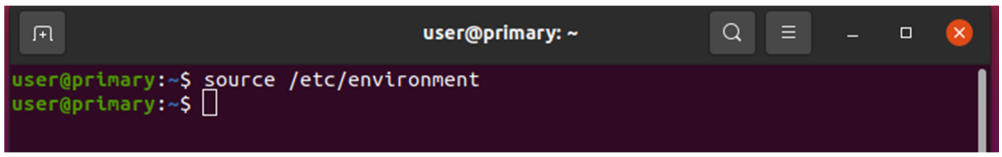

Figure 50 - formatter HDFS

Formatons ensuite le système hdfs avec :

Avec la commande : hdfs namenode -format

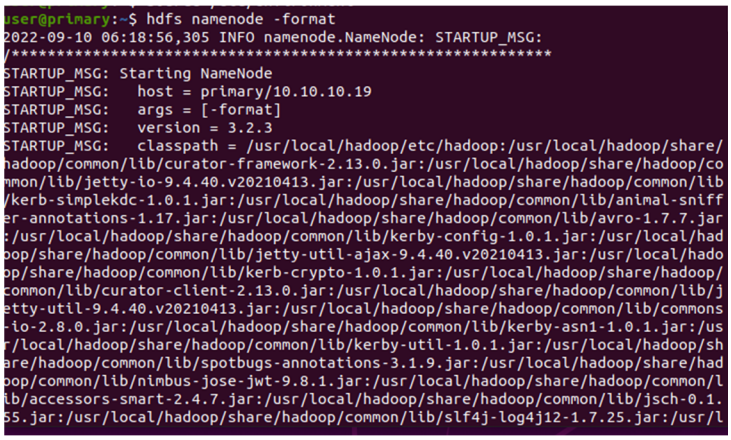

Figure 52 - Formatter HDFS 1

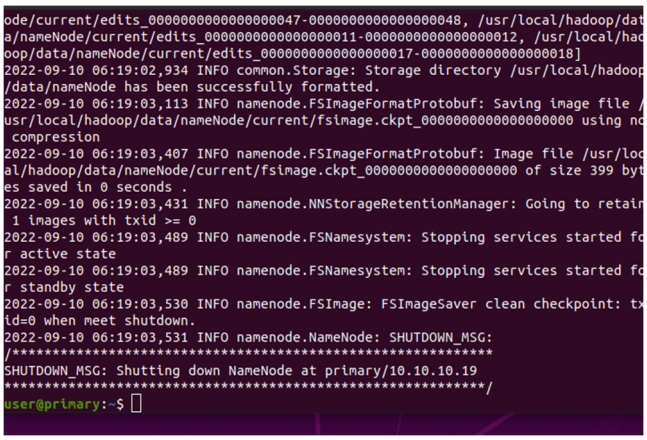

Figure 51 - Formatter HDFS 2

Assurons-nous que notre fichier .bashrc est configuré :

Ouvrons le fichier avec la commande : sudo nano .bashrc

Et vérifions si à la fin du fichier ont le chemin suivant [PATH] : (machine principale et secondaire)

export PDSH_RCMD_TYPE=ssh

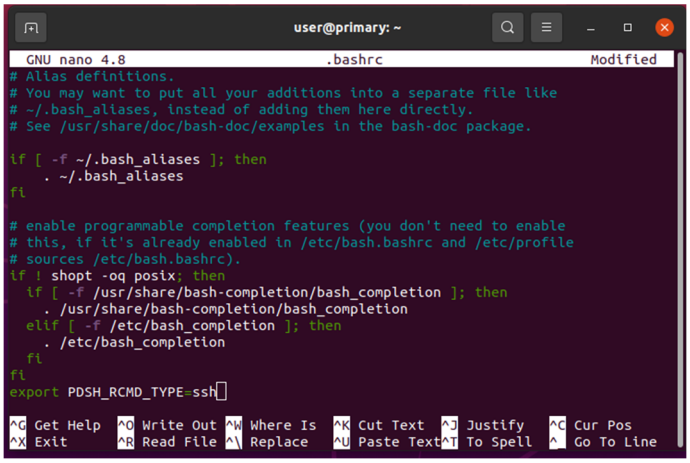

Figure 53 - Configuration du chemin pdsh

Mettez à jour les modifications :

Avec la commande : source ~/.bashrc

Lorsque ces opérations sont terminées, démarrez le service :

Avec la commande : start-dfs.sh

Pour vérifier si toutes les machines utilisent les bonnes ressources, utilisez : jps

- Sortie primaire :

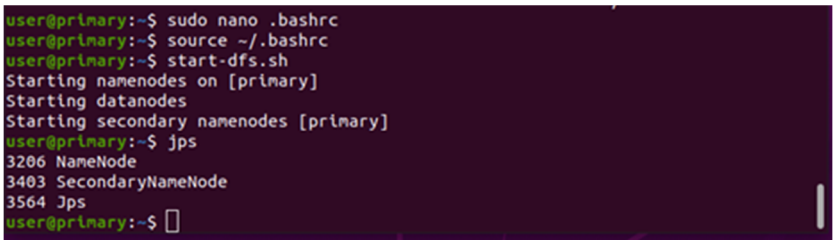

Figure 54 - Sortie jps (master)

- Sortie secondary:

Figure 55 - Sortie jps (slave1)

Figure 56 - Sortie jps (slave2)

3.5.22 Outil de gestion des nœuds

Il est temps de vérifier si tout fonctionne bien. Écrivons l'adresse IP principale sur notre navigateur en utilisant le port 9870. (Ex: primary:9870).

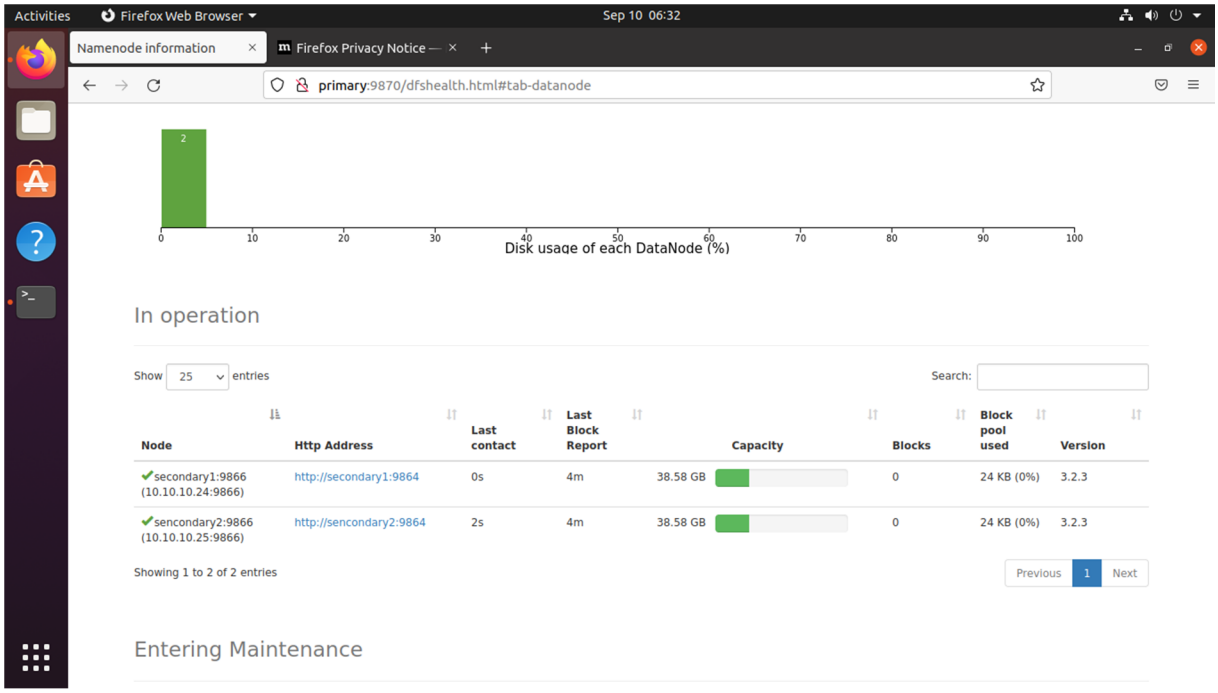

Figure 57 - Gestion des nœuds en Hadoop

3.5.23 YARN configuration

Pour configurer le YARN, nous devons commencer à exporter tous les chemins : (sur la machine principale)

export HADOOP_HOME="/usr/local/hadoop"

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop

export HADOOP_HDFS_HOME=$HADOOP_HOME

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_YARN_HOME=$HADOOP_HOME

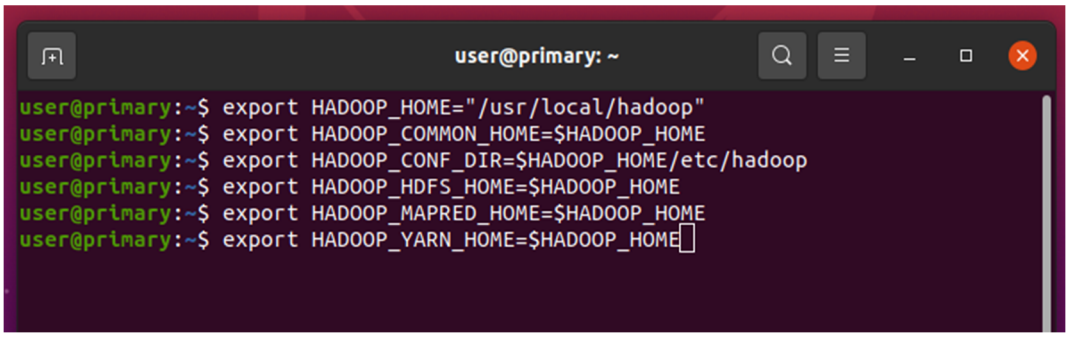

Figure 58 - Configuration YARN

Maintenant, changeons simplement la configuration du YARN sur les deux secondaires :

sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml

Figure 59 - Changement de configuration Yarn (slaves)

3.5.24 Démarrer Yarn

Pour démarrer le service Yarn, utilisons :

start-yarn.sh

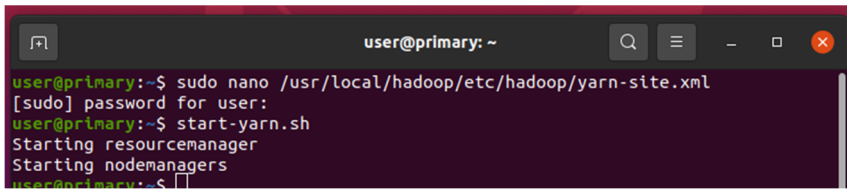

Figure 60 - Demarrage de YARN

Figure 61 - Sortie jps

Pour avoir accès à l'outil de gestion de Yarn, utilisons notre navigateur pour accéder à l'IP primaire sur le port 8088 :

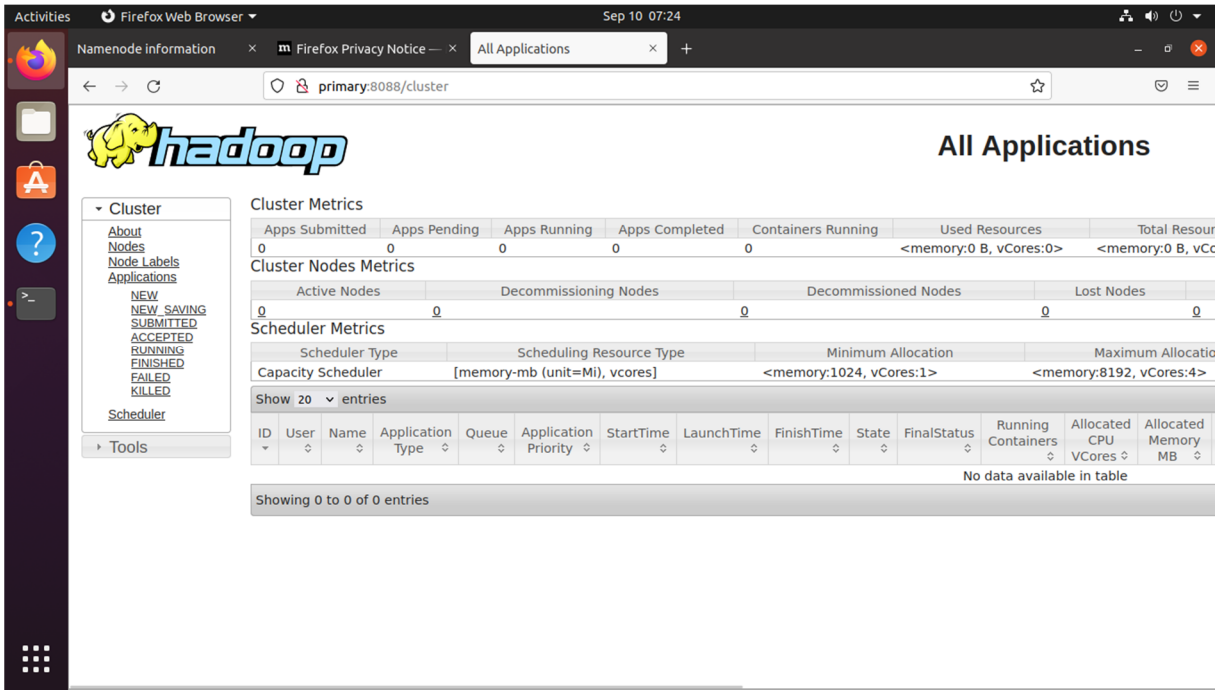

Figure 62 - Accès à l'outil de gestion de YARN

3.5.25 Télécharger Apache Spark

Avec la commande : wget https://dlcdn.apache.org/spark/spark-3.3.0/spark-3.3.0-bin-hadoop3.tgz

Une fois le téléchargement terminé, décompressons-le : tar xzf spark-3.3.0-bin-hadoop3.tgz

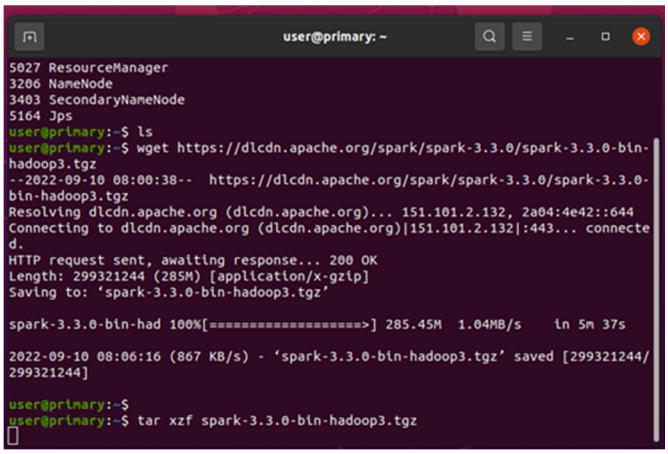

Figure 63 - Telechargement et decompression de fichier Spark

Renommer le fichier avec la commande : mv spark-3.3.0-bin-hadoop3.tgz spark-3

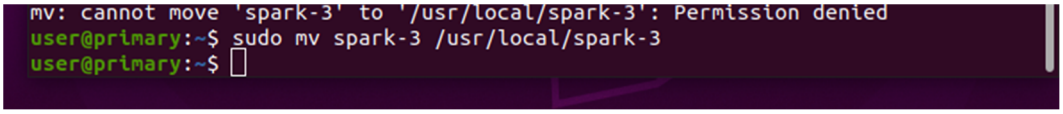

Figure 64 - Renommer le fichier spark-3

Ajoutons les configurations dans le fichier ./.bashrc

Ouvrons le fichier avec la commande : sudo nano ./.bashrc

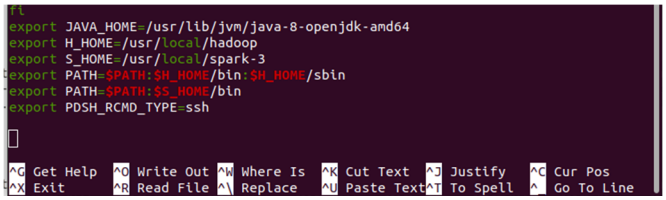

Figure 65 - Ajout des configurations

Mettons à jour les modifications :

Avec la commande : source ~/.bashrc

3.5.26 Vérifier l'installation de Spark

En appliquant la commande : spark-shell

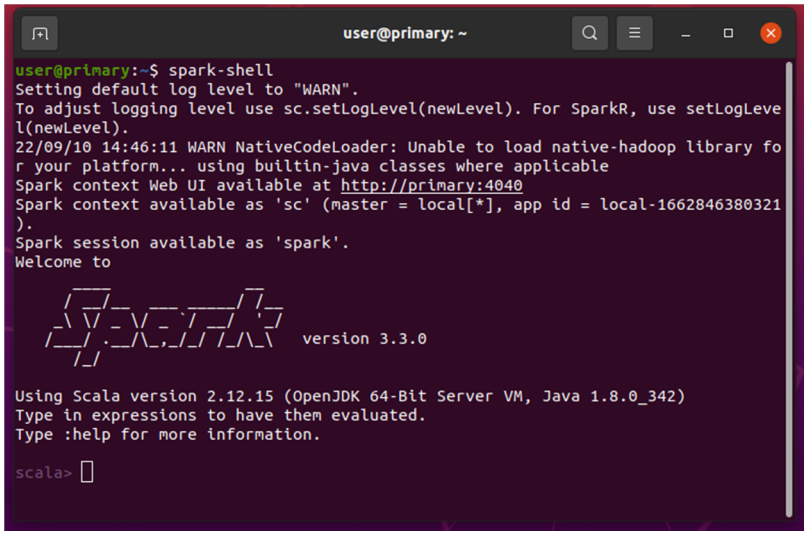

Figure 66 - Vérification de l'installation de Spark

Il est le temps de vérifier si tout fonctionne bien. Écrivons l'adresse IP principale sur votre navigateur en utilisant le port 4040. (Ex : primary:4040).

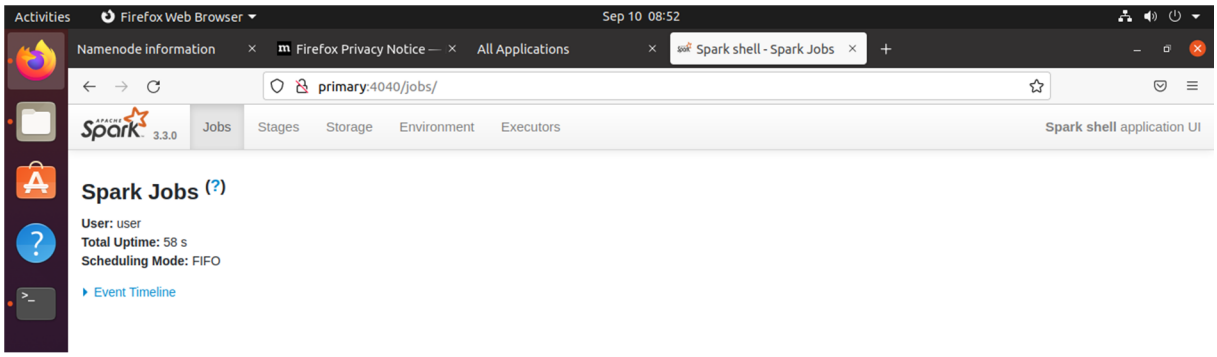

Figure 67 - Vérification de Spark sur le navigateur

3.5.27 Installer Python3-pip

Avec la commande : sudo apt install python3-pip

Figure 68 - Installation de Python3-pip

3.5.28 Installer OpenCv

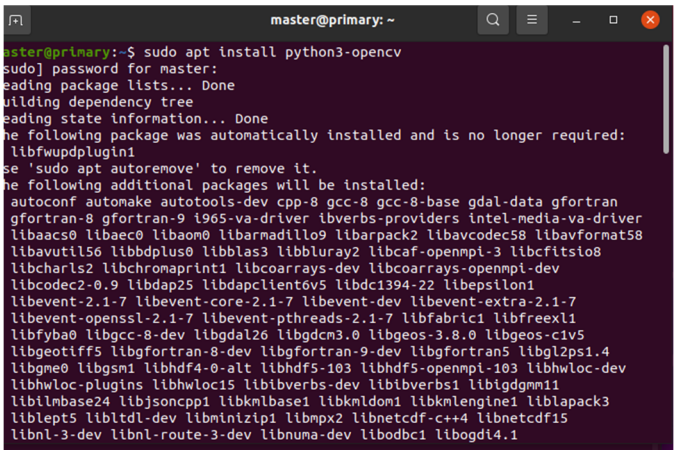

Figure 69 - Installation de OpenCv

3.5.29 Chargement des données

```
user@primary:~$ hdfs dfs -mkdir /Images
user@primary:~$ hdfs dfs -ls /
Found 1 items
drwxr-xr-x   - user supergroup          0 2022-09-10 09:23 /Images
```

Figure 70 - Chargement des images

```
user@primary:~/images$ hdfs dfs -ls /Images/*
Found 22 items
-rw-r--r--   2 user supergroup   12798899 2022-09-10 09:26 /Images/images/Copie de A_1302_1.LEFT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   14784329 2022-09-10 09:27 /Images/images/Copie de A_1302_1.LEFT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   16011459 2022-09-10 09:27 /Images/images/Copie de A_1305_1.RIGHT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   14055594 2022-09-10 09:27 /Images/images/Copie de A_1316_1.LEFT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   16892206 2022-09-10 09:26 /Images/images/Copie de A_1316_1.LEFT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   33666068 2022-09-10 09:27 /Images/images/Copie de A_1389_1.LEFT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   29406098 2022-09-10 09:27 /Images/images/Copie de A_1412_1.RIGHT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   13815255 2022-09-10 09:27 /Images/images/Copie de A_1458_1.RIGHT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   16452008 2022-09-10 09:26 /Images/images/Copie de A_1506_1.LEFT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   18159345 2022-09-10 09:26 /Images/images/Copie de A_1559_1.LEFT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   19000473 2022-09-10 09:26 /Images/images/Copie de A_1559_1.LEFT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   34386611 2022-09-10 09:26 /Images/images/Copie de A_1560_1.LEFT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   34934144 2022-09-10 09:26 /Images/images/Copie de A_1560_1.LEFT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   17253920 2022-09-10 09:26 /Images/images/Copie de A_1561_1.RIGHT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   19226152 2022-09-10 09:26 /Images/images/Copie de A_1562_1.LEFT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   20021358 2022-09-10 09:27 /Images/images/Copie de A_1562_1.LEFT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   18489779 2022-09-10 09:26 /Images/images/Copie de A_1565_1.LEFT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   17168362 2022-09-10 09:27 /Images/images/Copie de A_1568_1.LEFT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   18720882 2022-09-10 09:27 /Images/images/Copie de A_1584_1.LEFT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   19359094 2022-09-10 09:26 /Images/images/Copie de A_1584_1.LEFT_MLO.LJPEG.png
-rw-r--r--   2 user supergroup   17450412 2022-09-10 09:26 /Images/images/Copie de A_1599_1.RIGHT_CC.LJPEG.png
-rw-r--r--   2 user supergroup   20817996 2022-09-10 09:26 /Images/images/Copie de A_1740_1.LEFT_CC.LJPEG.png
```

Figure 71 - Stockage des images

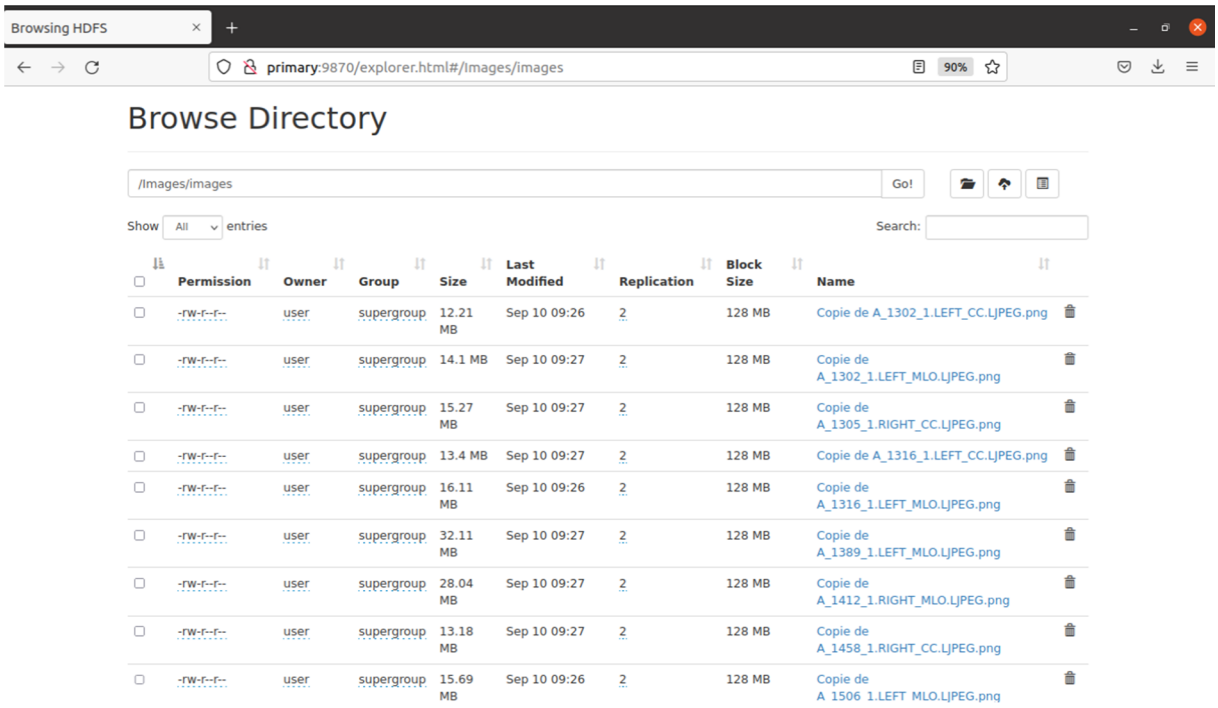

Figure 72 - Vérification de stockage des images on Hadoop

3.5.30 Résultats Obtenus :

Après on applique le filtre de Canny sur ces images et voici quelques exemples :

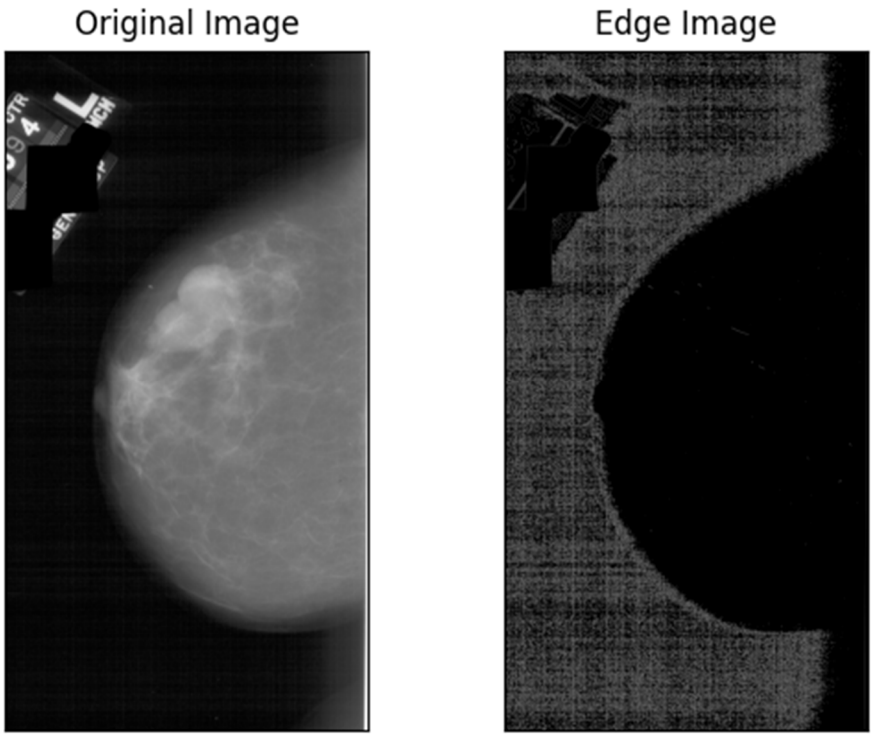

Figure 73 - Exemple 1

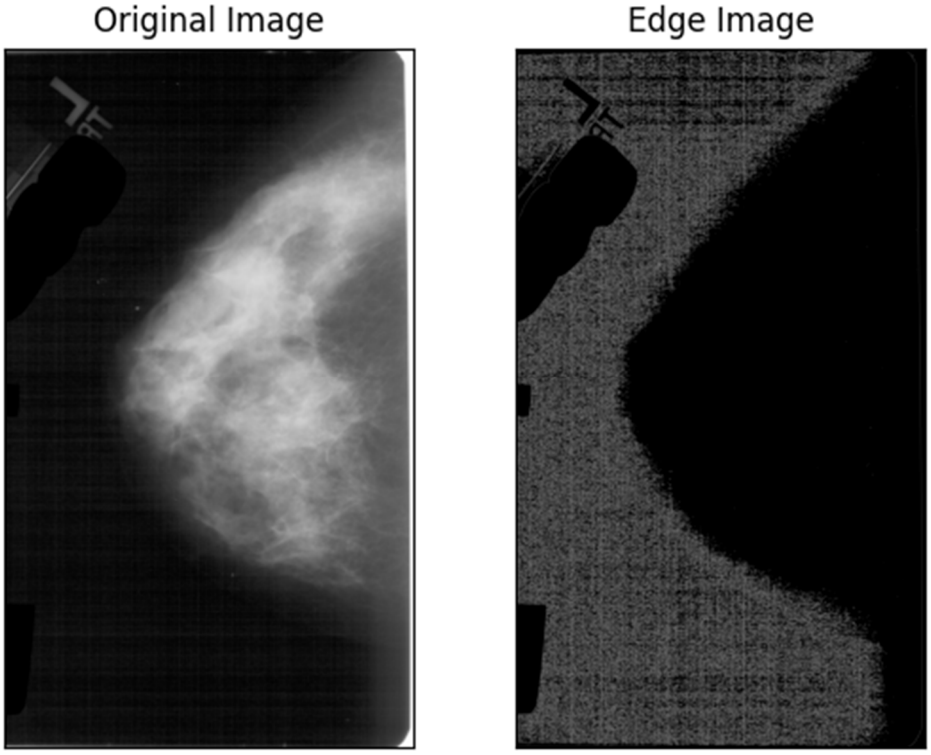

Figure 74 - Exemple 2

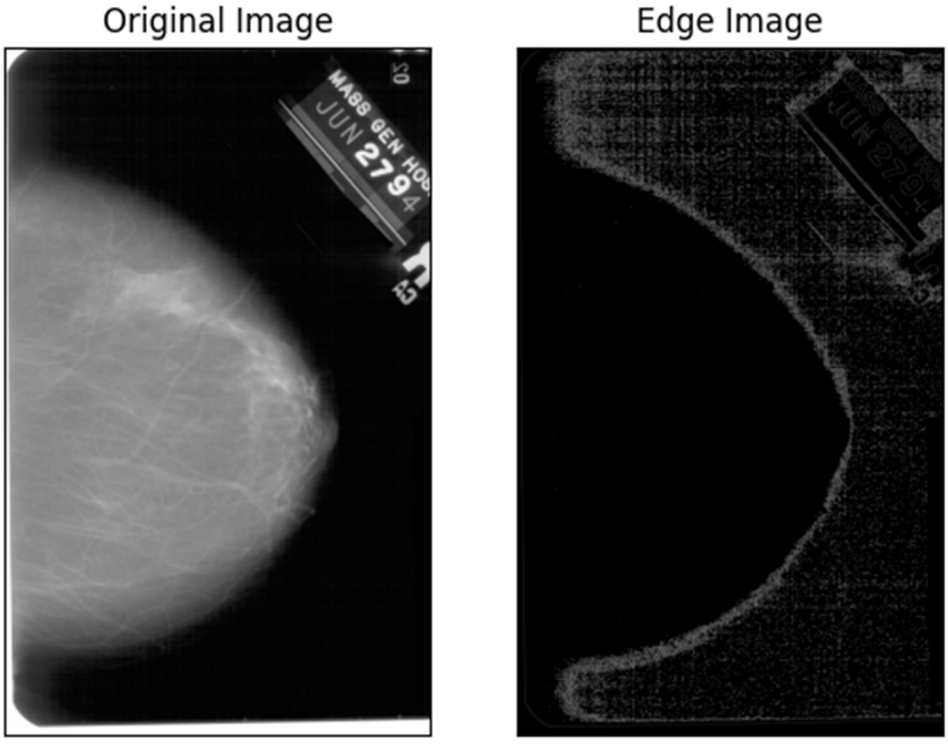

Figure 75 - Exemple 3

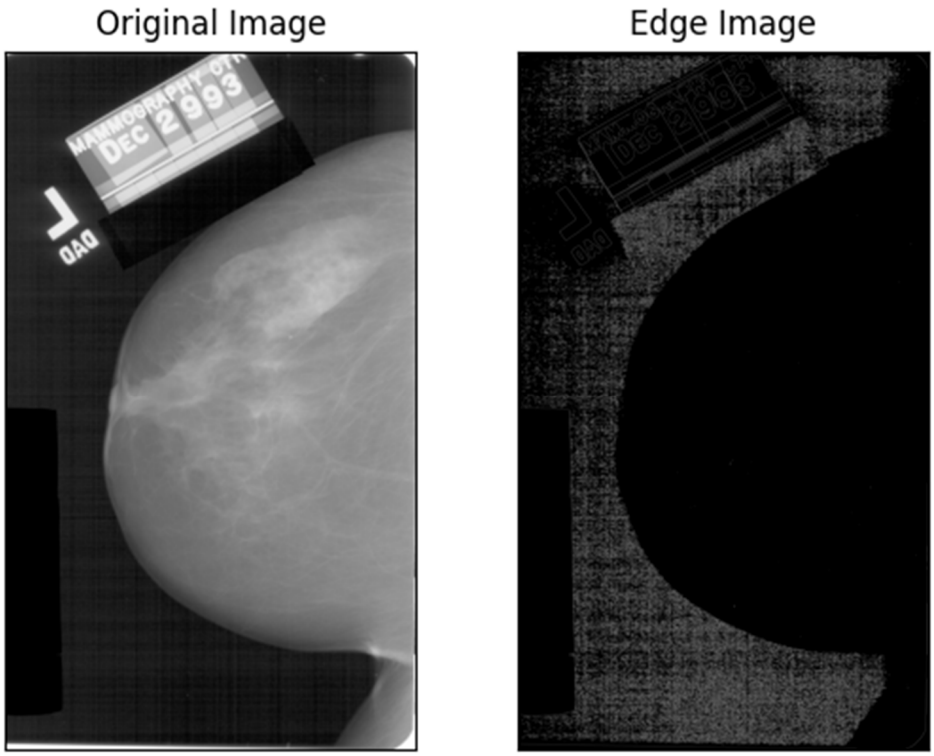

Figure 76 - Exemple 4

Conclusion Générale

Conclusion Générale

Dans ce travail, nous avons mené une étude sur le Big Data et ses techniques dans le domaine de la santé. Nous avons réalisé une application permettant aux laboratoires de faire un traitement parallèle sur les images médicales du diagnostic de cancer, en appliquant un ensemble de critères aux données stockées.

Pour résoudre le problème de l'augmentation de la taille des données et leur impact sur le processus de diagnostic, nous avons réalisé une application Big Data. Nous avons utilisé les deux Framework Hadoop et Spark. Le noyau d'Hadoop est constitué d'une partie de stockage : HDFS (Hadoop Distributed File System), et d'une partie de traitement appelée MapReduce. Hadoop fractionne les fichiers en gros blocs et les distribue à travers les nœuds du cluster. Pour traiter les données, Hadoop 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 qui repose sur un système de fichiers parallèle où les calculs et les données sont distribués via les réseaux à grande vitesse.

- **Perspectives**

Le domaine de recherche reste très ouvert aux différents travaux et différents tests dans des architectures fortement distribuées, néanmoins le futur projet qui se voit très intéressant, est la mise en œuvre d'un cluster Hadoop sur des machines physiques puissantes dans une plateforme High-Performance-Computing (HPC) pour pallier les problèmes du Hardware qu'on a rencontré.

Références

Webographie

[2] https://www.futura-sciences.com/tech/definitions/informatique-big-data-15028/ Mars2022

[3]https://www.google.com/search?q=3v+du+big+data&sxsrf=ALiCzsYKc-IwZ0S7tTKecCu6ON8E7L95Mw:1663806006776&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjG8IiwkKf6AhVxhv0HHXmLC94Q_AUoAXoECAEQAw&biw=1366&bih=657&dpr=1#imgrc=x271QVi94hmJiM

[4] https://www.oracle.com/dz/big-data/what-is-big-data/ Mars2022

[5]https://www.google.com/search?q=5v+du+big+data&tbm=isch&ved=2ahUKEwjBnYKxkKf6AhUXPhoKHau3ANkQ2-cCegQIABAA&oq=5v+du+big+data&gs_lcp=CgNpbWcQAzIECAAQQzIICAAQHhAHEAU6BAgjECc6BggAEB4QB1DLDFihEmCTHWgAcAB4AIABdogB4QKSAQMwLjOYAQCgAQGqAQtnd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=OKorY8GhLpf8aKvvgsgN&bih=657&biw=1366#imgrc=mR_WKuX-9x5c1M

[6]https://www.lemagit.fr/definition/Collecte-de-donneesMars2022

[7]https://www.google.com/search?q=les+couches+de+big+data&biw=1024&bih=625&sxsrf=ALiCzsZo1o443iL6Hlr3x9TZ0NLe9CZuyA%3A1663842527260&ei=3zgsY6a9D9qUxc8PzJ6i-Ak&oq=les+couches+de+Big&gs_lcp=Cgdnd3Mtd2l6EAEYADIFCCEQoAEyBQghEKABMgUIIRCgATIICCEQHhAWEB06BggAEB4QBzoICAAQHhAHEAo6CAgAEB4QDxAHOggIABAeEAgQBzoICCEQwwQQoAFKBAhBGABKBAhGGABQAFjfIWDqMmgAcAF4AIABiwKIAaMWkgEGMC4xLjEymAEAoAEBwAEB&sclient=gws-wiz

[8]https://www.silicon.fr/hub/hpe-intel-hub/ne-confondez-pas-le-big-data-avec-un-data-warehouse-geant Mars2022

[9]https://www.talend.com/fr/resources/architecture-big-data/Mars2022

[10]https://www.google.com/search?q=Architecture+de+Big+Data&tbm=isch&ved=2ahUKEwj0wOOpkaf6AhWWgc4BHesvADoQ2-cCegQIABAA&oq=Architecture+de+Big+Data&gs_lcp=CgNpbWcQAzIECCMQJzoECAAQHjoGCAAQHhAIOgQIABBDOgUIABCABDoICAAQgAQQsQM6BwgjEOoCECdQnQZY1TVgtj5oBXAAeACAAbMBiAGkE5IBBDAuMjGYAQCgAQGqAQtnd3Mtd2l6LWltZ7ABCsABAQ&sclient=img&ei=NqsrY_RNloO6vg_r34DQAw&bih=657&biw=1366#imgrc=484-4Us6XJDLhM le 15/04/2022

[12] https://datascientest.com/data-architecture Mars2022

[13] https://www.pasteur.fr/fr/journal-recherche/dossiers/comment-big-data-revolutionne-recherche-sante

[14] https://www.lcl.fr/mag/tendances/big-data-definition-enjeux-et-applications Mars2022

[15] Big Data et ses technologies ETS 2017.07 - Philippe Laflamme

[16]https://www.oracle.com/fr/database/nosql/what-is-nosql/#:~:text=sur%20NoSQL%20Cloud-,%C3%80%20quoi%20sert%20une%20base%20de%20donn%C3%A9es%20NoSQL%20%3F,%C3%A9lev%C3%A9e%20et%20une%20haute%20disponibilit%C3%A9. le 05/08/2022

[17] https://www.ambient-it.net/top-meilleures-db-nosql/ Mars2022

[18]https://www.cigref.fr/archives/histoire-cigref/blog/cloud-computing-histoire-de-linformatique-en-nuages/ Mars2022

[19]https://www.google.com/search?q=cloud+computing&tbm=isch&ved=2ahUKEwiJrOn3kaf6AhVDYBoKHb2wBfgQ2-cCegQIABAA&oq=clou&gs_lcp=CgNpbWcQARgAMgQIIxAnMgQIIxAnMgQIABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDMgQIABBDOgcIIxDqAhAnOgUIABCABDoICAAQsQMQgwE6CAgAEIAEELEDUJADWLIfYOotaAFwAHgCgAGWAYgB5RqSAQQwLjI4mAEAoAEBqgELZ3dzLXdpei1pbWewAQrAAQE&sclient=img&ei=2asrY8nbKcPAab3hlsAP&bih=657&biw=1366#imgrc=sforFXRSWRe2qM

[21]https://www.ideematic.com/actualites/2017/12/avantages-cloud-computing-entreprises/#:~:text=Le%20cloud%20computing%20est%20en,aux%20entreprises%20et%20aux%20utilisateurs. le 15/08/2022

[22] https://www.redhat.com/fr/topics/cloud-computing/public-cloud-vs-private-cloud-and-hybrid-cloud#:~:text=Il%20existe%20quatre%20principaux%20types,as%2Da%2DService).

[23]https://www.oracle.com/dz/cloud/what-is-cloud-computing/#:~:text=une%20seule%20architecture.-,Services%20de%20Cloud%20computing,solution%20adapt%C3%A9e%20%C3%A0%20vos%20besoins.Mars2022

[24]https://www.google.com/search?q=differentes+couches+du+cloud+computing&tbm=isch&ved=2ahUKEwj9_7Tzkqf6AhVXwoUKHRGUCgUQ2-cCegQIABAA&oq=differentes+couches+du+cloud+computing&gs_lcp=CgNpbWcQAzoECCMQJzoECAAQQzoFCAAQgAQ6BggAEB4QBzoICAAQHhAIEAdQ3hNY8kxg91JoAHAAeACAAZgBiAGEF5IBBDAuMjSYAQCgAQGqAQtnd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=3KwrY_2hNdeElwSRqKoo&bih=657&biw=1366#imgrc=1qY_lsMaZ9tM8M
le 05/03/2022

[25] https://www.techno-science.net/glossaire-definition/Framework.html le 05/08/2022

[26] https://www.lebigdata.fr/hadoop le 04/08/2022

[27] https://www.talend.com/fr/resources/what-is-hadoop/ le 05/08/2022

[28] https://www.lemagit.fr/tutoriel/A-la-decouverte-dHadoop le 05/08/2022

[29]https://www.google.com/search?q=Architecture+de+Hadoop&tbm=isch&ved=2ahUKEwiopLTDk6f6AhUQ04UKHUJLDCMQ2-cCegQIABAA&oq=Architecture+de+Hadoop&gs_lcp=CgNpbWcQAzIFCAAQgAQ6BAgjECc6CAgAEIAEELEDOgQIABBDOgsIABCABBCxAxCDAToHCCMQ6gIQJ1CfB1iMJ2COLmgBcAB4AIABmwGIAYwkkgEEMC4zOZgBAKABAaoBC2d3cy13aXotaW1nsAEKwAEB&sclient=img&ei=hK0rY6jSJpCmlwTClrGYAg&bih=657&biw=1366

[30] https://tutoriels.edu.lat/pub/hadoop/hadoop-hdfs-overview/hadoop-presentation-de-hdfs

[31] https://fr.acervolima.com/anatomie-de-la-lecture-et-de-l-ecriture-de-fichiers-dans-hdfs/

[32]https://cloud.google.com/learn/what-is-hadoop le 04/08/2022

[33]https://www.google.com/search?q=terminologie+de+base+de+hadoop+mapreduce&tbm=isch&ved=2ahUKEwjZ98bqlaf6AhWIgs4BHWnwCAsQ2-cCegQIABAA&oq=Terminologie+de+base+de+Hadoop+MapReduce&gs_lcp=CgNpbWcQARgAMgQIIxAnUABYAGCVhQdoAHAAeACAAXSIAecBkgEDMC4ymAEAqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=768rY5nEJIiFur4P6eCjWA&bih=657&biw=1366

[34] https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm le 04/08/2022

[35] http://b3d.bdpedia.fr/calculdistr.html le 05/08/2022

[36]https://www.google.com/search?q=ressource+manager&tbm=isch&ved=2ahUKEwiciMLRlqf6AhUR1IUKHTcuCt8Q2-cCegQIABAA&oq=ressource+manager&gs_lcp=CgNpbWcQAzIFCAAQgAQyBggAEAoQGDIGCAAQChAYMgYIABAKEBg6BAgjECc6BggAEB4QCDoECAAQHjoECAAQEzoGCAAQHhATOgYIAB
AeEAVQuwRYxWRgr2ZoA3AAeACAAaIBiAH1LpIBBDAuNTCYAQCgAQGqAQtnd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=x7ArY9yIIJGolwS33Kj4DQ&bih=657&biw=1366 le 05/08/2022

[37] https://www.lebigdata.fr/yarn-apache-hadoop

[38]https://www.techopedia.com/definition/30427/hadoop-commonle05/08/2022

[39]https://www.google.com/search?q=ecosystem+de+hadoop&tbm=isch&ved=2ahUKEwit6_3flqf6AhUP8IUKHd02At8Q2-cCegQIABAA&oq=ecosystem+de+hadoop&gs_lcp=CgNpbWcQAzoECCMQJzoGCAAQChAYOgUIABCABDoHCCMQ6gIQJzoICAAQgAQQsQM6BAgAEEM6BAgAEB46BggAEB

4QCFD_B1jsXWDyY2gCcAB4AoABsQGIAbYdkgEEMC4zMZgBAKABAaoBC2d3cy13a
XotaW1nsAEKwAEB&sclient=img&ei=5bArY-3kNI_glwTd7Yj4DQ&bih=657&biw=1366
[40] https://www.lcl.fr/mag/tendances/big-data-definition-enjeux-et-applications
le 04/08/2022
[41] https://www.snowflake.com/guides/what-spark le 14/08/2022
[42]https://www.google.com/search?q=Architecture+d%E2%80%99Apache+Spark&tbm=isc
h&ved=2ahUKEwj708Ggl6f6AhUFgc4BHSlDCSEQ2-
cCegQIABAA&oq=Architecture+d%E2%80%99Apache+Spark&gs_lcp=CgNpbWcQAzoE
CCMQJzoHCCMQ6gIQJ1DuBlj_GGCnIWgBcAB4AoABigGIAcARkgEEMS4xOZgBAKA
BAaoBC2d3cy13aXotaW1nsAEKwAEB&sclient=img&ei=bbErY7vpC4WCur4PqYaliAI&b
ih=657&biw=1366 le 05/08/2022
[43]https://www.google.com/search?q=Ecosysteme+de+Spark&tbm=isch&ved=2ahUKEwiw
y8zNl6f6AhUMRhoKHRhnDtQQ2-
cCegQIABAA&oq=Ecosysteme+de+Spark&gs_lcp=CgNpbWcQAzoECCMQJzoHCCMQ6g
IQJ1DSA1iBGGCvHGgBcAB4A4AB1wKIAfsMkgEHMC44LjAuMZgBAKABAaoBC2d3c
y13aXotaW1nsAEKwAEB&sclient=img&ei=y7ErY_C5LYyMaZjOuaAN&bih=657&biw=1
366
[44]https://www.google.com/search?q=hadoop+vs+spark&tbm=isch&ved=2ahUKEwj0wKO
DmKf6AhW0hc4BHXQyBMgQ2-
cCegQIABAA&oq=hadoop+&gs_lcp=CgNpbWcQARgBMgQIIxAnMgQIABBDMgQIABB
DMgQIABBDMgQIABBDMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCA
AQgAQ6BwgjEOoCECc6CAgAEIAEELEDUIoEWNctYPk6aAJwAHgCgAGlAYgBoxmS
AQQwLjI2mAEAoAEBqgELZ3dzLXdpei1pbWewAQrAAQE&sclient=img&ei=PLIrY_SyE
7SLur4P9OSQwAw&bih=657&biw=1366 Mars2022
[45] https://www.techopedia.com/definition/25690/vmware-workstation 15/04/2022
[46]https://www.google.com/search?q=ubuntu&tbm=isch&ved=2ahUKEwiO7duwmKf6AhU
FdBoKHR8GB7oQ2-
cCegQIABAA&oq=ubun&gs_lcp=CgNpbWcQARgAMgQIABBDMgUIABCABDIFCAAQ
gAQyBQgAEIAEMgUIABCABDIFCAAQgAQyBQgAEIAEMgUIABCABDIFCAAQgAQy
BQgAEIAEOgQIIxAnOgYIABAeEAc6BAgAEB46BggAEB4QCDoGCAAQHhAFOgcIIxD
qAhAnOggIABCABBCxAzoLCAAQgAQQsQMQgwFQ1gVYsx1grCloAXAAeAKAAZkBi
AG5EpIBBDAuMTmYAQCgAQGqAQtnd3Mtd2l6LWltZ7ABCsABAQ&sclient=img&ei=
m7IrY86zJYXoaZ-MnNAL&bih=657&biw=1366 le 05/08/2022
[47] https://www.futura-sciences.com/tech/definitions/technologie-ubuntu-19676/15/04/2022

[48] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5971336/

[49]https://www.google.com/search?q=Architecture++filtre+de+canny&tbm=isch&ved=2ahUKEwi4tpXnmaf6AhUygHMKHUw6D7wQ2-cCegQIABAA&oq=Architecture++filtre+de+canny&gs_lcp=CgNpbWcQAzoECCMQJ1CIEliIEmCEFWgAcAB4AIABcogB4gGSAQMwLjKYAQCgAQGqAQtnd3Mtd2l6LWltZ8ABAQ&sclient=img&ei=GrQrY7jBDrKAzgPM9LzgCw&bih=657&biw=1366 Mars2022

[50]https://www.google.com/search?q=Le+Traitement+de+MapReduce&tbm=isch&ved=2ahUKEwiMlb-NlKf6AhUJ4RoKHW-mDkwQ2-cCegQIABAA&oq=Le+Traitement+de+MapReduce&gs_lcp=CgNpbWcQAzoECCMQJzoHCCMQ6gIQJ1DMBVjEGGCgImgBcAB4AoABkgGIAZ0UkgEEMC4yM5gBAKABAaoBC2d3cy13aXotaW1nsAEKwAEB&sclient=img&ei=IK4rY4wFicJr78y64AQ&bih=657&biw=1366 le 05/08/2022

Bibliographies

[1] Gil Press, " A Very Short History Of Big Data "December 21, 2013

[11] Livre électronique intitulé : Big data application and architecture; de himanshu et soumendra mohanty, Springer; 2015th edition (July 10, 2015)

[20] Badger, L., Grance, T., Patt-Corner, R., & Voas, J. (2012). Draft cloud computing synopsisand recommendations. Gaithersburg: NIST,