MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIEENTIFIQUE

Université ABBES LAGHROUR Khenchela

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

DEPARTEMENT DE MATHEMATIQUE ET INFORMATIQUE

MEMOIRE

PRESENTE POUR L'OBTENTIR DU DIPLOME DE MASTRE EN INFORMATIQUE (L M D)

PAR :

ATTARI Yassine

BARECHE Mohamed Ramzi

OPTION : SECURITE ET TECHNOLOGIE WEB (STW)

UNE IMPLEMENTATION MULTI-AGENTS DE LA TECHNIQUE MAP-REDUCE POUR LA RECHERCHES DES DONNÉES EN BIG DATA.

Encadré par :

- Dr : SIAM Abderrahime

2021/2022

Remerciements

Nous remercions Dieu pour nous avoir donné la santé, la patience et le courage pour faire et finaliser ce travail.

Nous tenons à remercier vivement **Dr : SIAM Abderrahime** pour tous les efforts pour le succès de ce travail ; merci aux jurys pour l'intérêt qu'ils ont montré en acceptant d'examiner et d'évaluer notre mémoire de fin d'études.

Bien entendu, Nous nous n'oublions pas de remercier tous nos collègues de la promotion 2021/2022 pour leurs encouragements et toute la famille et tous ceux qui nous ont aidé de près ou de loin durant la réalisation de ce travail.

BARECHE Med Ramzi

ATTARI Yassine

Résumé:

Dans ce mémoire on parlera du big data, l'écosystème hadoop, les systèmes multi-agents, on propose un système multi-agent pour les traitements des données de ressources différentes et de différentes types et une implémentation d'un exemple Map-Rduce en python sur Hadoop.

Mots clés: Big Data, Système multi agents, Hadoop, HDFS, Map-Reduce.

Abstract:

In this dissertation we will talk about big data, the hadoop ecosystem, multi-agent systems, we propose a multi-agent system for processing data from different resources and of different types and an implementation of a Map-Rduce example in python on Hadoop.

Key words: Big Data, multi-agent systems, Hadoop, HDFS, Map-Reduce.

الملخص

في هذه المذكرة سنتحدث عن البيانات الضخمة، والنظام البيئي hadoop، والأنظمة متعددة العوامل، ونقترح نظامًا متعدد العوامل لمعالجة البيانات من مصادر مختلفة وأنواع مختلفة وتنفيذ مثال Map-Rduce في Python على Hadoop.

الكلمات المفتاحية البيانات الضخمة، الأنظمة متعددة العوامل ,Hadoop, HDFS, Map-reduce.

Table des matières

Listes des figures :

Introduction générale

De nos jours, des très grandes quantités de données générées de manière massive, à partir de diverses sources de réseaux sociaux, internet, Google, appareils mobiles, système GPS… etc. Ces données générées d'une manière rapide et en temps réel et à grande échelle. Elles sont de différentes types et structures ; On trouve des textes, des audio, des images et des vidéo, le big data est un terme apparu avec l'augmentation de ces données.

Le défi confronté dans ce contexte les systèmes classiques et les entrepôts de données rencontrent le problème de la dégradation des performances face à une quantité de données aussi importante en termes d'analyse et de traitement.

Pour simplifier notre problème de recherche et traitement du big data on a utilisé les systèmes multi agents. Les systèmes multi-agents ont des applications dans le domaine de l'intelligence artificielle où ils permettent de réduire la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-ensembles, en associant un agent intelligent indépendant à chacun de ces sous-ensembles et en coordonnant l'activité de ces agents.

Plusieurs modèles de programmation ont apparu, et l'une des techniques les plus puissantes de traitement et l'analyse des données est le framework Hadoop. Hadoop est un environnement d'exécution distribuée et performant, il propose un système de stockage distribué via son système de fichier HDFS (Hadoop Distributed File System) et un système d'analyse et de traitement de données basé sur le modèle de programmation MapReduce pour réaliser des traitements parallèles et distribués sur des gros volumes de données.

Le présent mémoire est organisé comme suit :

Le premier chapitre : nous allons présenter une introduction du concept du Big Data avec ses définitions, ses caractéristiques, ses avantages et son importance, ses fonctionnements, ses technologies et les domaines dans lesquels on l'utilise.

Le deuxième chapitre: est consacré à la présentation du Framework Hadoop, les caractéristiques de ses principaux composants HDFS et de MapReduce, les écosystèmes de Hadoop.

Le troisième chapitre: nous présenterons les systèmes multi agents, le concept d'agent, les types d'agents et les communications entre les agents.

Le quatrième chapitre : présent la conception du système en utilisant l'AUML, et une implémentation de l'agent du traitement de données textuelles.

Enfin, Nous concluons par une conclusion générale.

Chapitre I Big Data

I.1 Introduction

Depuis longtemps, les données générées n'ont fait qu'augmenter : à l'heure actuelle, la quantité de données générée chaque année est très importante, estimée à près de 3 trillions ($3 * 10^{18}$) octets **[1]**. La croissance des données affecte tous les secteurs de la science et de l''économie, ainsi que le développement d'applications Web et de réseaux sociaux **[2]**, ce qui a conduit à l'apparition du terme Big Data. Le mot anglo-saxon signifie littéralement « big data », et sa traduction officielle française recommandée est le big data, même quand on parle parfois de big data. Aujourd'hui, ces mégas donnés sont devenues le centre d'attention des participants dans tous les domaines d'activité.

I.2 Origine du Big data

Le Big Data est un nouveau contexte et une grande quantité de données qui ne peuvent être traitées avec les technologies traditionnelles. Le premier projet Big Data concerne les participants qui effectuent la recherche d'informations sur les moteurs de recherche internet (tels que Google et Yahoo). En fait, ces participants sont tous confrontés au problème de la mise à l'échelle du système et de la réponse aux demandes des utilisateurs.

Le Big Data est devenu une tendance de base pour de nombreux acteurs de l'industrie car il contribue à la qualité du stockage, du traitement et de l'analyse des données. **[3]**

I.3 Définition du Big Data

Contrairement aux données traditionnelles, le terme Big Data fait référence à de grands ensembles de données en croissance comprenant des formats hétérogènes: données structurées, non structurées et semi-structurées. Le Big Data a une nature complexe qui nécessite des technologies puissantes et des algorithmes avancés. Ainsi, les outils de Business Intelligence statiques traditionnels ne peuvent plus être efficaces dans le cas d'applications Big Data.**[4]**

Big data, littérairement les grosses données, est une expression anglophone utilisée pour désigner des ensembles de données qui deviennent tellement volumineux qu'ils en deviennent difficiles à travailler avec des outils classiques de gestion de base de données. Il s'agit donc d'un ensemble de technologies, d'architecture, d'outils et de procédures permettant à une organisation très rapidement de capter, traiter et analyser de larges quantités et contenus hétérogènes et changeants, et d' extraire les informations pertinentes à un coût accessible.**[5]**

I.4 Caractéristiques du Big Data

La caractérisation de Big Data est généralement faite selon 3«V», V de Volume, de Variété et de Vélocité. D'autres "V" complémentaires peuvent s'ajouter, comme la valeur et la véracité. **[6]**

Le Volume

Fait référence à la grande quantité de données générées chaque seconde. Pensez simplement à tous les e-mails, tweets, photos, vidéos, données de capteurs que nous générons et partageons chaque seconde. Nous ne parlons plus en téraoctets, mais en zettabytes ou brontobytes.

Rien que sur Facebook, nous envoyons 10 millions de messages chaque jour, 4,5 millions de fois de « j'aime », et nous téléchargeons 350 millions de nouvelles photos chaque jour. Maintenant il y a une grande quantité de données sera générée chaque minute. Or, une telle quantité de données est trop grande pour être stockée ou analysée de manière « traditionnelle » (c'est-`a-dire une base de données). Avec le Big Data, nous pouvons utiliser des systèmes distribués pour stocker et utiliser ces ensembles de données, ou' différentes parties des données sont stockées à différents endroits mais collectées par logiciel. **[7]**

La Vélocité

La vitesse à laquelle les données sont traitées ou reçues. Considérez simplement que les publications sur les réseaux sociaux se répandront en quelques secondes, les transactions bancaires frauduleuses peuvent être détectées en quelques minutes, ou un logiciel qui analyse les réseaux sociaux et saisit l'heure qui a déclenché l'achat. Doit être des millisecondes ! Désormais, le big data nous permet d'analyser le moment ou les données sont générées au lieu d'avoir à les analyser dans la base de données. **[7]**

La Variété

Les types de données traditionnels ont été des données ayant une structure et sont stockées dans une base de données relationnelle, mais avec l'apparence du Big Data, les données ne sont pas nécessairement structurées telles que les données texte, audio et vidéo requièrent un prétraitement supplémentaire pour dégager du sens et prendre en charge les métadonnées.**[8]**

La Véracité

Désigne la fiabilité des données. Avec autant de formes de mega-données, sa qualité et sa précision sont difficiles à vérifier (regardons les tweets avec des balises, des abréviations, des fautes de frappe, la familiarité, la fiabilité et l'exactitude du contenu). Mais ! Le Big Data et l'analyse nous permettent désormais d'utiliser ces données pour la production. Le manque de qualité et de précision est généralement le résultat d'une production de masse. **[7]**

La Valeur

C'est le dernier 'V' à considérer quand on parle de big data. Avoir accès au big data c'est super, mais il faut quand même le convertir en valeur, sinon ce sera inutile ! Par conséquent, dans ce sens, on peut dire que la valeur est très importante ! Il est également important pour les entreprises d''évaluer la rentabilité de la collecte de données. Sans bien comprendre et définir les avantages, nous pouvons facilement tomber dans le piège de la réalisation de projets Big Data. Combien nous coutent-ils ? **[7]**

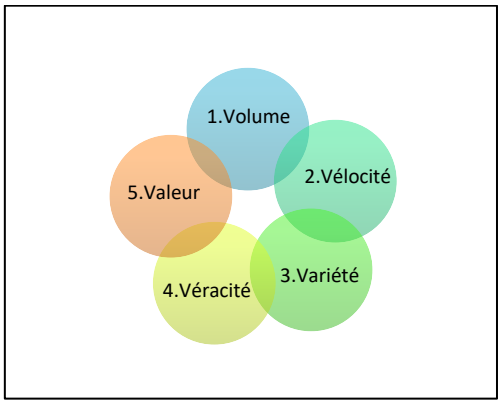


Figure I.1 : Les 5 'V' de Big Data. [9]

I.5 Les avantages de Big Data

Plusieurs avantages peuvent être associés Big Data, nous pouvons citer par exemple :

- **Prendre les bonnes décisions:** toute partie, qu'elle soit gouvernementale ou privée, rentable ou volontaire, peut économiser beaucoup d'argent en exploitant les données. En planifiant des décisions saines et correctes et des plans pour l'avenir.
- **Augmentation des ventes**: les producteurs et les commerçants peuvent bénéficier d'informations stockées sur des réseaux sociaux tels que Facebook et Twitter pour voir la réactivité de leurs offres, campagnes publicitaires et autres éléments les aidant à planifier leurs produits et à augmenter leurs ventes.
- **Meilleurs services de santé:** les données des patients enregistrés à l'hôpital, telles que les dossiers pré-médicaux, peuvent être utilisées pour fournir des services plus rapides. Et mieux pour les patients.
- **Prévention des maladies:** les médecins peuvent éviter les données du génome humain (trois milliards de caractères contenant des informations humaines) Nombreuses maladies et traitement des maladies incurables.
- **Réduire le coût de la publicité**: toute partie peut envoyer aux clients des publicités personnalisées en fonction de leurs intérêts, directement via le système appelé(Microtargeting). **[10]**

I.6 Gestion de Big Data

Gestion de Big Data ou bien Big Data Management est une nouvelle discipline dans laquelle les techniques, outils et plates-formes de gestion de données y compris le stockage, le prétraitement, le traitement et la sécurité peuvent être appliqués. **[11]**

Le rôle de la gestion des données est assure un haut niveau de qualité des données et aide les entreprises à faire face à la quantité des données qui grandit.

a) Stockage des données :

Stocker les données on pétaoctets de façon distribuée utilise les services de Cloud, le stockage consiste trois opérations principales (Regroupement, Réplication, indexation).

b) Prétraitement :

Avant l'analyse de Big Data on a besoin de vérifier la qualité des données et réparer les données au traitement par l'application des étapes suivantes (nettoyage des données, transformation, intégration, transmission, réduction, discrétisation).

c) Traitement :

C'est aptitude de traiter un grand volume de données quel que soit le type ou bien la structure et l'emplacement de ces données, ce traitement peut être classification ou prédiction.

d) Sécurité :

Pour sécuriser un grand volume des données, plusieurs algorithmes de sécurité sont apparus pour la confidentialité, intégrité, disponibilité.

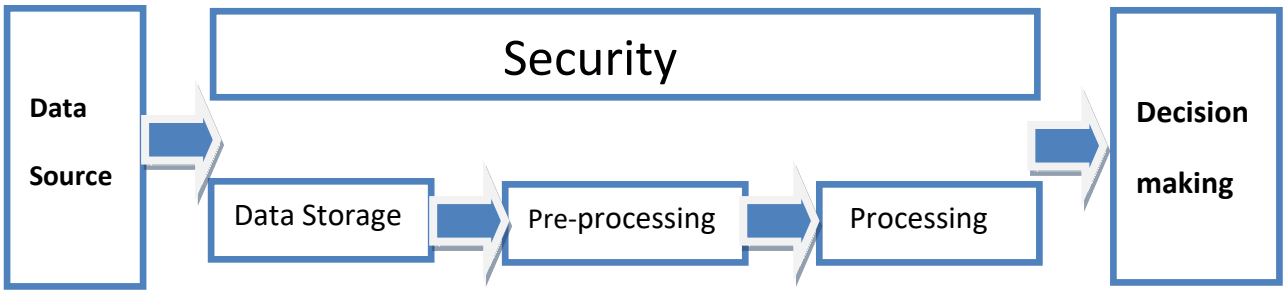


Figure I.2 : Les étapes de gestion de Big Data. [12]

I.7 Quelques domaines d'utilisation du BigData

Le BigData trouve sa place dans de nombreux domaines. Citons quelques domaines d'utilisation du BigData :

- ✓ **La Télécommunication:** l'analyse de l'état du réseau en temps réel.

- ✓ **Les Banques:** la sanctuarisation de données anciennes dues à des contraintes réglementaires.
- ✓ **Les Médias Numériques:** le ciblage publicitaire et l'analyse de sites web.
- ✓ **Les Marchés Financier:** l'analyse des transactions pour la gestion des risques et la gestion des fraudes, ainsi que pour l'analyse des clients.
- ✓ **Les Services Publics:** l'analyse des compteurs (gaz, électricité, etc.) et la gestion des équipements.
- ✓ **Le Marketing:** le ciblage publicitaire et l'analyse de tendance.
- ✓ **La Santé:** l'analyse des dossiers médicaux et l'analyse génomique. **[13]**

I.8 Fonctionnement du Big Data

Le Big Data offre de nouvelles perspectives, qui ouvrent de nouvelles opportunités et favorisent de nouveaux Business Model. Son adoption implique trois actions principales :

- ✓ **Intégrer:**

Le Big Data rassemble des données provenant de sources et d'applications disparates. Les mécanismes d'intégration des données classiques, comme ETL (extraire, transformer et charger) ne sont généralement pas à la hauteur. Pour analyser des jeux de Big Data à l'échelle de téraoctets, voire de pétaoctets, il est nécessaire d'adopter de nouvelles stratégies et technologies.

Lors de la phase d'intégration, vous devez importer les données, les traiter et vous assurer qu'elles sont formatées et disponibles sous une forme que vos analystes peuvent exploiter.

- ✓ **Gérer :**

Le Big Data nécessite du stockage. Votre solution de stockage peut se trouver dans le cloud, sur site, ou les deux à la fois. Vous pouvez stocker vos données sous la forme de votre choix et imposer à ces jeux de données vos exigences de traitement, ainsi que les moteurs de traitement nécessaires, à la demande. Nombreux sont ceux qui choisissent leur solution de stockage en fonction de l'endroit où sont hébergées leurs données. Le cloud est de plus en plus adopté, car il prend en charge vos besoins informatiques actuels et laisse la possibilité d'augmenter les ressources en fonction des besoins.

- ✓ **Analyser :**

Votre investissement dans le Big Data porte ses fruits dès lors que vous êtes en mesure d'analyser vos données et d'agir à partir de l'analyse. Forgez-vous un nouveau point de vue grâce à une analyse visuelle de vos divers jeux de données. Explorez davantage les données afin de faire de nouvelles découvertes. Partagez vos conclusions avec d'autres utilisateurs. Créez des modèles de données avec l'apprentissage automatique et l'intelligence artificielle. Exploitez vos données. **[8]**

I.9 Les Technologies de Big Data

Nous avons cité quelques technologies utilise le Big Data:

HADOOP : Hadoop un framework mis au point par la Apache Software Fundation afin de mieux généraliser l'usage du stockage et traitement massivement parallèle de MapReduce 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 grands nombres de données.

Bases NoSQL: Les bases de données 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.

I.10 Big Data et Data warehouse

Dans le passé, les entrepôts de données étaient des structures structurées stockant les informations dans un format normalisé et liées aux systèmes de leurs processus qui gèrent ces données depuis le stockage, le traitement, l'accès et la mise à jour. Ces entrepôts ensuite sont appelées bases de données et ont utilisé la structure relationnelle pour stocker ces informations et ont adopté les systèmes de gestion de bases de données relationnelles SGBDR. Ces règles et règlements sont apparus au début des années 90.Avec l'augmentation des données stockées et la nécessité pour les entreprises de bénéficier de ces données. Au début des années 80, un système dédié à la prise de décisions pour l'entreprise à partir ses entrepôts de données et le terme Data Warehouse est apparu.

Data Warehouse est une base de données (données structurées) regroupant une partie ou l'ensemble des données fonctionnelles d'une entreprise. Il entre dans le cadre de l'informatique décisionnelle; son but est de fournir un ensemble de données servant de référence unique, utilisées pour la prise de décisions dans l'entreprise par les baies de statistiques et de rapports réalisés via des outils de Reporting. D'un point de vue technique, il sert surtout à 'délester' les bases de données opérationnelles des requêtes pouvant nuire à leurs performances. **[14]**

Les entreprises traditionnellement utilisent ses entrepôts de données pour la gestion des données relationnelles et structurée, donc il suffit d'utiliser les systèmes de gestion des bases de données relationnelles SGBDRs pour manipuler ce type de données.

Cependant, avec l'avènement du Big Data, le défi pour les entrepôts de données est de réfléchir à une approche complémentaire avec le Big Data, on pourrait concevoir un modèle hybride. Dans ce modèle les restes de données optimisées opérationnelles très structurées seront stockées et analysées dans l'entrepôt de données, tandis que les données qui sont fortement distribuées et non structurées seront contrôlées par Big Data (Hadoop ou NoSQL). **[14][15]**

On peut donc interfacer Big Data avec le DataWarehouse(DW), effectivement les données non structurées provenant de différentes sources peuvent être regroupées dans un HDFS avant d'être transformées et chargées à l'aide d'outils spécifiques dans le DataWarehouse et les outils traditionnels de BD. **[15][16]**

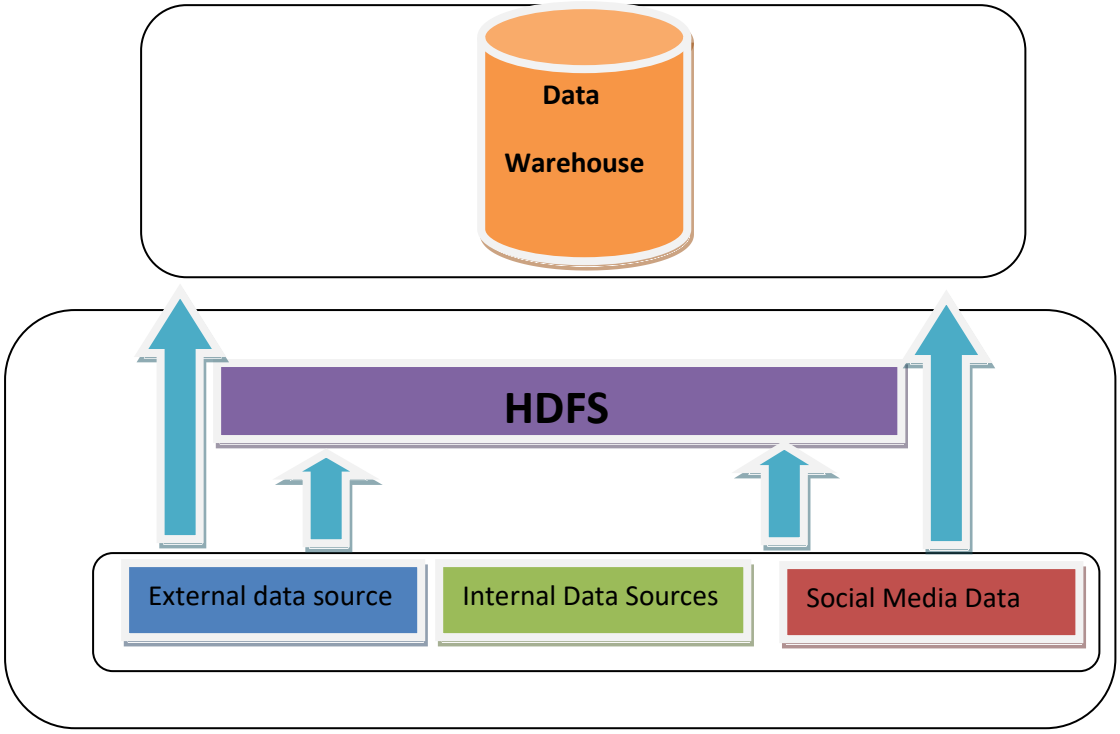


Figure I.3 : Liens de Big data et DataWarehouse. [15]

I.11 Conclusion

Les données devenant de plus en plus volumineuse et complexe, nos bases de données traditionnelles sont limitées face à l'analyse et au traitement de ces données. Dans un souci de gain de temps, de nouvelle technologie sont venu pour soulager les entreprises génératrices d'un grand nombre de données. L'analyse de Big Data est sans aucun doute vouée à gagner une importance, certains parlent même de révolution.

Chapitre II L'écosystème d'Hadoop

II.1 Introduction

La programmation distribuée, sous la forme de processus répartis sur un ensemble (un cluster, un cloud, une grille) de machines, est la seule solution qui permet de traiter en temps raisonnable de gros problèmes et de gros volumes de données. Selon les besoins en échanges de données, et selon les capacités relatives de traitement et de transfert de données, une application distribuée peut être simple ou très complexe à développer.

Pour débarrasser le data scientist d'une partie de ces préoccupations, des intergiciels distribués (ou middleware) spécialisés dans le stockage et l'analyse de données ont émergé, comme Hadoop. Ce chapitre présente les principales caractéristiques de l'architecture logicielle d'Hadoop, destinée finalement à faciliter le stockage distribué des gros volumes de données et à supporter une chaîne de traitements de type Map-Reduce.

II.2 Historique

En 2004, Google publie un article présentant son algorithme basé sur des opérations analytiques à grande échelle sur un grand cluster de serveurs, le MapReduce, ainsi que son système de fichier en cluster, le GoogleFS. Doug Cutting, qui travaille à cette époque sur le développement de l'Apache Lucene et rencontre des problèmes similaires à ceux de la firme de Mountain View, décide alors de reprendre les concepts décrits dans l'article pour développer sa propre version des outils en version open source, qui deviendra le projet Hadoop. **[16]**

Le logo et le nom de ce nouveau framework Java sont inspirés par Doug Cutting du doudou de son fils de cinq ans. **[17]**

En 2006, Doug Cutting a décidé de rejoindre Yahoo avec le projet Nutch et les idées basées sur les premiers travaux de Google en termes de traitement et de stockage de données distribuées. Yahoo proposa Hadoop sous la forme d'un projet open source en 2008. **[18]**

La communauté open source lance Hadoop 2.07 celle-ci fut proposée au public en 2012 dans le cadre du projet Apache, sponsorisé par l'Apache Software Foundation. La révolution majeure a été l'ajout de la couche YARN dans la structure de Hadoop. À partir de septembre 2016, la version 3.0.0 est rendue disponible. **[18]**

II.3 Présentation de Hadoop

Hadoop est une plateforme open source de la fondation Apache, ayant une capacité de gérer des données volumineuses, qui sont structurées et non structurées. Elle est conçue pour trouver une solution aux problèmes liés à la volumétrie et la variété des données en les traitants sur des différents serveurs simultanément. Cette architecture va offrir une puissance, et un stockage importants, les données vont être par la suite répliquées et

réparties sur les machines du cluster, grâce à un système de réplication de façon à garantir une très haute disponibilité des données en cas de défaillance d'un ou de plusieurs serveurs. **[18]**

II.4 Caractéristiques de Hadoop

Hadoop est une architecture logicielle de stockage et d'analyse de données, dont on peut lister les propriétés suivantes :

1. *La plate-forme possède un système de fichiers distribué très facilement extensible*. Hadoop gère seul la distribution et le stockage des données sur ses différents nœuds, et pour augmenter la capacité de stockage il suffit d'ajouter des nœuds de données dans la plate-forme.
2. *Les codes des traitements sont routés jusqu'aux données*. Cette stratégie est la plus efficace pour de grosses volumétries de données stockées sur des machines standard reliées par des réseaux standard. Les nœuds de données se transforment donc en nœuds de calculs le temps des traitements, et par conséquent, augmenter le nombre de nœuds de données pour accroître la capacité de stockage augmente aussi la capacité de traitement.
3. *Des mécanismes de tolérance aux pannes sont intégrés à la plate-forme*. Hadoop étant conçu pour fonctionner sur du matériel standard (bon marché), des pannes fréquentes sont supposées inéluctables, et les données sont répliquées sur plusieurs nœuds afin d'être toujours accessibles. Quand un réplicat disparaît (suite à une panne), ses copies sont à nouveau répliquées pour maintenir un bon taux de réplication. De même, les tâches de traitements exécutées sur les nœuds de données sont monitorées et relancées sur le nœud d'un autre réplicat si une panne survient. L'utilisateur n'a pas à se soucier de la tolérance aux pannes.
4. *Un paradigme de programmation Map-Reduce est intégré à la plate-forme.* Ce paradigme convient à la récupération et au filtrage de données stockées dans l'ensemble des nœuds.**[10]**

II.5 L'architecture Hadoop

Nous allons présenter l'architecture de cluster dans Hadoop:

- **JobTracker:**

Le responsable de lancer des tâches distribuées aux autres machines ou bien les esclaves, ensuite, il contrôle l'état de ces esclaves et fait agréger les résultats de calculs.

- **NameNode:**

Est la machine qui fait la réplication et la répartition des données dans le cluster, possède toutes les informations des données et leurs emplacement, elle facilite donc l'accès des client aux fichiers stockés dans le cluster.

- **Secondary NameNode :**

Durant le traitement d'une opération: Chaque couple de minutes, Secondary NameNode va copier les nouveaux informations stockés dans de NameNode, Normalement, le Secondary NameNode doit être assure par une autre machine physique autre que le master afin d'assurer la continuité du fonctionnement du cluster.

Pour les machines esclaves, nous pouvons leur attribuer ces différents rôles :

- **TaskTracker:**

Permettre à l'esclave d'exécuter une tâche MapReduce sur les données qu'elle contient. JobTracker va envoyer les tâches à exécuter aux TaskTrackers. Nous pouvons remarquer qu'avec les solutions Big Data, les instructions sont amenées vers les données et non pas les données sont amenées aux instructions comme les programmes classiques.

- **DataNode:**

C'est la machine qui contient un bloc des données, en effet, les données sont généralement partitionnées et répliquées sur les différents nœuds du cluster pour garantir la disponibilité des données. Cette machine doit périodiquement informer NameNode par un rapport d'état. [21]

II.6 Les composants de Hadoop

II.6.1 HDFS (Hadoop Distributed File System)

HDFS est un système de fichiers aide au stockage des données structurées ou non sur des machines distribués (cluster). Il s'appuie sur le système de fichier natif de l'OS pour présenter un système de stockage unifié reposant sur un ensemble de disques et de systèmes de fichiers hétérogènes. Ce système est basé sur la redondance dont une donnée est stockée sur au moins N volumes différents. [10]

Les composants de HDFS :

HDFS définit de deux types de nœud:

- **NameNode:** Il se caractérise par :
  - ✓ faire la partition et de la duplication des blocs des données.
  - ✓ Stocker et gérer les informations des blocs (métadonnées).
  - ✓ Sauvegarder la liste des blocs pour chaque fichier (dans le cas de lecture).
  - ✓ Contenir la liste des DataNodes pour chaque bloc (dans le cas de l'écriture).
  - ✓ Tenir les attributs des fichiers (ex : nom, date de création, facteur de réplication).
  - ✓ Logs toute métadonnée et toute transaction sur un support persistant.
  - ✓ Créations/suppressions.[21]

- **DataNode:** Il se caractérise par:
  - ✓ Stocker des blocs de données dans le système de fichier local.
  - ✓ Maintenir des métadonnées sur les blocs possédés.
  - ✓ Heartbeat avec le NameNode : Heartbeat est système permettant sous Linux la mise en clusters de plusieurs serveurs pour effectuer entre eux un processus de tolérance de panne. Le processus Heartbeat se chargera de passer un message-aller vers le NameNode indiquant son identité, sa capacité totale, son espace utilisé, son espace restant. [21]
- **SecondaryNode:** Il caractérise par:
  - ✓ Télécharger régulièrement les logs sur le NameNode.
  - ✓ Crée une nouvelle image en fusionnant les logs avec l'image HDFS.
  - ✓ Renvoie la nouvelle image au NameNode. [21]

Ecriture dans un fichier ou volume HDFS :

Pour écrire un fichier au sein d'HDFS:

- **Etape 1:** On va utiliser la commande principale de gestion de Hadoop: Hadoop, avec l'option fs. Admettons qu'on souhaite stocker un fichier. Ex: data.txt sur HDFS.
- **Etape 2:** Le programme va diviser le fichier en blocs de 64KB (ou autre, selon la configuration) – supposons qu'on ait ici 3 blocs.
- **Etape 3:** Le NameNode lui indique les DataNodes à contacter.
- **Etape 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 chaque bloc. [22]

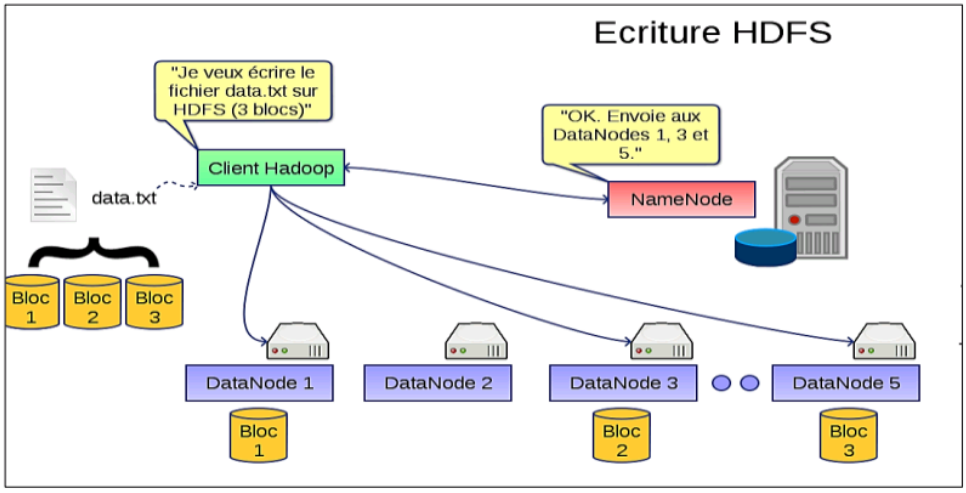


Figure II.1 Processus d'écriture dans un volume ou fichier HDFS [22]

Lecture d'un fichier HDFS :

Pour lire un fichier existe dans HDFS, il faut suivre les étapes suivantes :

- **Etape1:** Le client indique au NameNode qu'il souhaite lire le fichier par exemple: data.txt.
- **Etape2:** Le NameNode lui indiquera la taille de fichier (nombre de blocs) ainsi que les différents DataNode hébergeant les n blocs.
- **Etape3:** Le client récupère chacun des blocs à un des DataNodes.
- **Etape4:** En cas d'erreur/non réponse d'un des DataNode, il passe au suivant dans la liste fournie par le NameNode. [21]

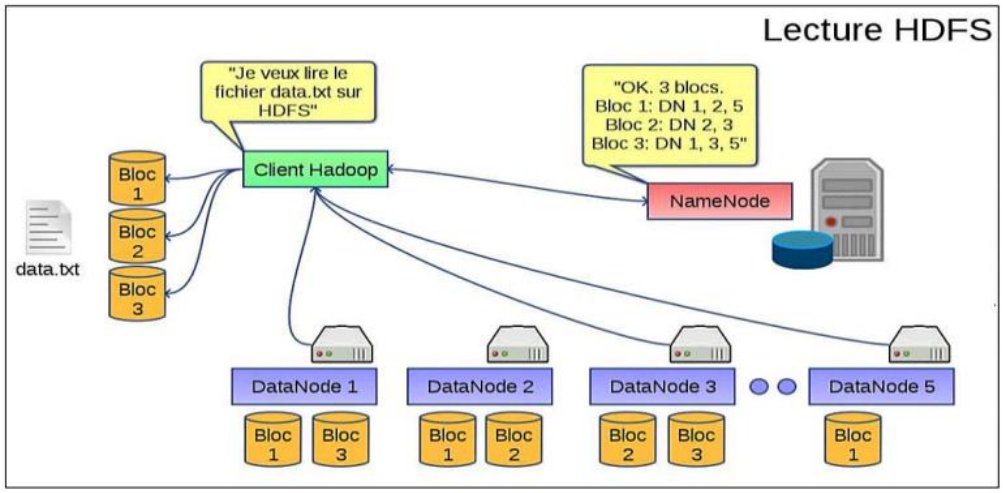


Figure II.2 Lecture d'un fichier HDFS [22]

II.6.2 MapReduce

Le principe de MapReduce est simple, il s'agit de découper une tâche manipulant un gros volume de données en plusieurs tâches traitant chacune un sous-ensemble de ces données. MapReduce a deux étapes Map et Reduce. Dans Map les tâches sont donc dispatchées sur l'ensemble des nœuds. Chaque nœud traite un ensemble de données concernées. Dans Reduce, les résultats sont fusionnés pour former le résultat final du traitement. Nous pouvons aussi distinguer des autres étapes intermédiaires comme suivant:[20]

- ***Map :***

La fonction mapper est pour lire les données stockées et de les découper et générer un autre ensemble de données sous forme de tuples (paire de clé/valeur).

Dans Hadoop, cela se traduit par plusieurs exécutions de la fonction Map, sur les machines esclaves qui contient les données.

- ***Combiner :***

Une étape intermédiaire gérée directement par Hadoop, son rôle est de trier regrouper les paires avec des clés identiques. Elle sert donc d'une part à réduire le résultat à la sortie du mapper et d'autre part à faciliter la vie du Reduce.

- ***Shuffle :***

Une étape intermédiaire aussi permet de regrouper les tuples ayant la même clé dans une seul tuple mais contient la fusion des autres résultats.

- ***Reduce :***

La fonction Reduce prend la sortie de la phase Shuffle pour agréger les données. Chaque tâche de Reduce produit un fichier de sortie qui sera stocké dans le système de fichiers HDFS.[20]

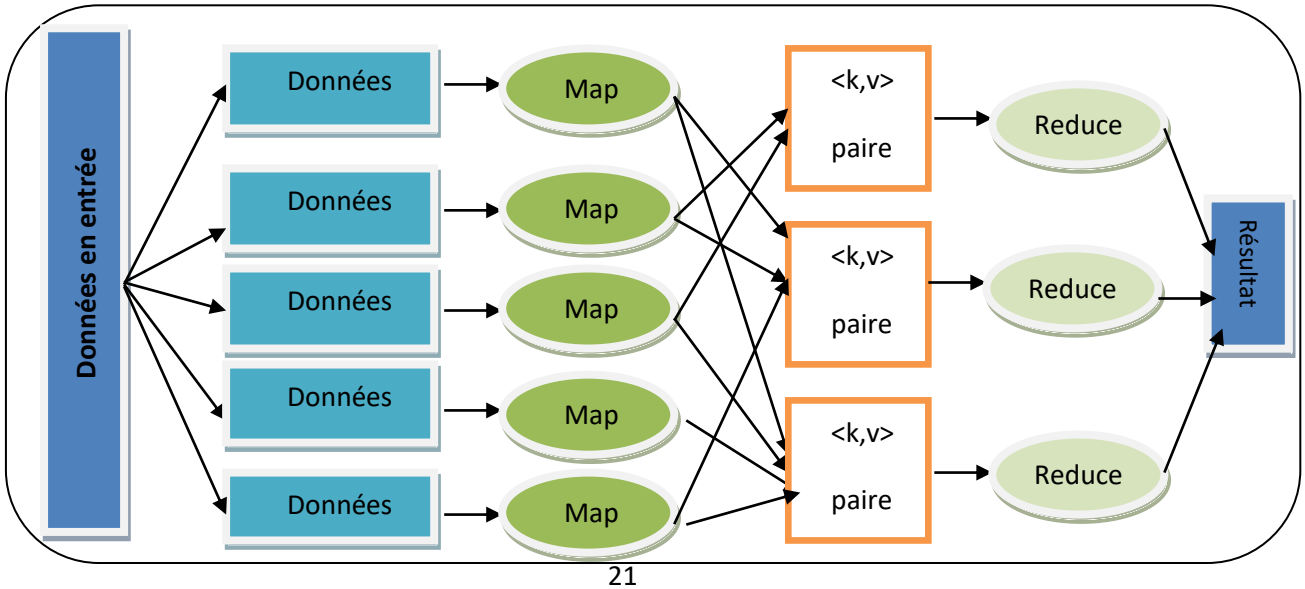


Figure II.3 Exemple des étapes de MapReduce.

L'architecture de MapReduce :

Le MapReduce possède une architecture maître-esclave

- **Le maître MapReduce :** le JobTracker.
- **Les esclaves MapReduce :** les TaskTracker.

Le JobTracker :

- ✓ Gérer l'ensemble des ressources du système.
- ✓ Recevoir les jobs des clients.
- ✓ Ordonnancer les différentes tâches des jobs.
- ✓ Assigner les tâches aux TaskTrackers.
- ✓ Réaffecter les tâches défaillantes.
- ✓ Sauvegarder des informations sur l'état d'avancement des jobs.

Le TaskTracker :

- ✓ Exécute les tâches données par le JobTracker.
- ✓ Exécution des tâches dans une autre JVM (Child).
- ✓ A une capacité en termes de nombres de tâches qu'il peut exécuter.
- ✓ Heartbeat avec le JobTracker. [21]

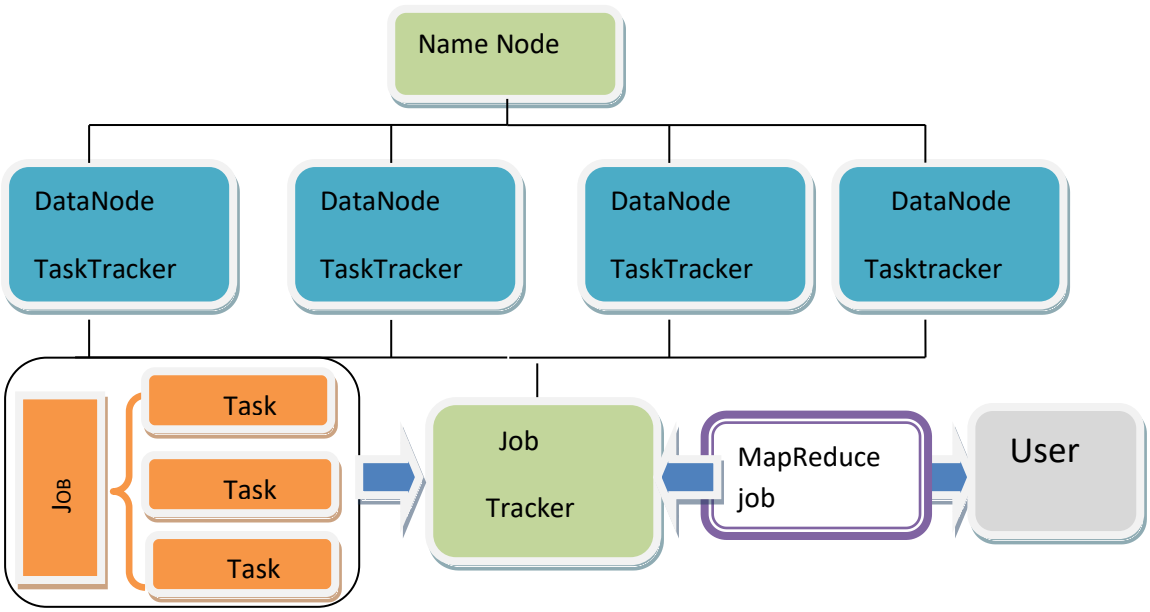


Figure II.4 L'architecture de MapReduce.

II.7 L'écosystème Hadoop

Nous allons identifier dans ce schéma qui suit les composants de l'écosystème Hadoop :

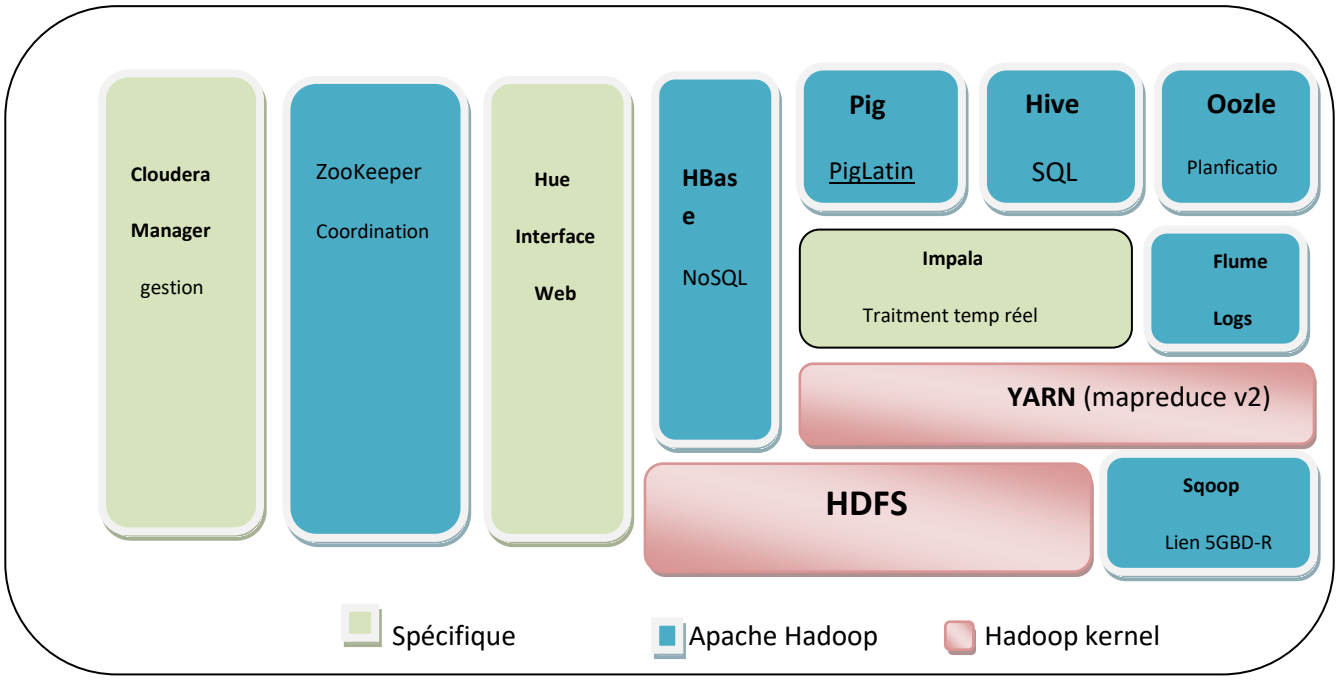


Figure II.5 Outils composant 1e noyau HADOOP. [23]

II.8 Conclusion

Dans ce chapitre on a présenté l'outil Hadoop et ses composants principaux HDFS et MapReduce, l'architecture de cluster dans Hadoop, la façon de distribuer les données dans plusieurs machines (DataNode), les machines qui gèrent les métadonnées (NameNode), et la façon du traitement et de la gestion des données, et tous les concepts qui concernent l'outil théoriquement.

Chapitre III Système multi agents

III.1 Introduction :

L'intelligence artificielle (IA) s'inspire du raisonnement de l'être humain ou sa façon de concevoir des modèles, il l'est transcrit sous forme d'agent intelligent.

L'intelligence Artificielle Distribuée (IAD) consiste à distribuer l'expertise au sein d'une société d'entités appelées agents dont le contrôle et les données sont distribués, d'où le principal intérêt du paradigme multi-agents : ne pas uniquement concevoir des entités intelligentes, mais aussi mettre ces entités en relation de manière intelligente.

Le principe des systèmes multi agents (SMA) est de partager et de distribuer la connaissance et la capacité de raisonnement entre plusieurs agents. Chacun de ses agents est spécialisé dans un sous domaine du domaine du départ. Ce qui lui permet selon les ressources dont il dispose, de résoudre partiellement ou tout le problème, d'améliorer sa solution et de compléter les données qui lui manquent. Ces tâches se font par communications entres les agents.

III.2 Le concept d'agent :

III.2.1 Définition d'un agent :

On appelle agent une entité physique ou virtuelle : [24]

a. Qui est capable d'agir dans un environnement
b. Qui peut communiquer directement avec d'autres agents
c. Qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser)
d. Qui possède des ressources propres
e. Qui est capable de percevoir (mais de manière limitée) son environnement
f. Qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune)
g. Qui possède des compétences et offre des services
h. Qui peut éventuellement se reproduire
i. Dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose, et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit.

Un agent a trois capacités qui forment une boucle : la phase d'action modifie l'environnement, ce qui peut provoquer la naissance de nouveaux messages, lesquels seront à nouveau perçus par l'agent. La figure III.1 présente ces trois phases. [26]

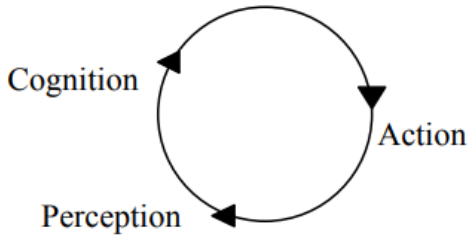


Figure III.1 Schéma de réalisation d'une tâche par un agent [26].

L'agent peut donc entreprendre différentes actions : percevoir une partie de son environnement ; agir sur son environnement (autonome) ; produire un raisonnement, en vue d'une action, d'une adaptation ou d'un résultat ; communiquer et coopérer.

II.2.2 Caractéristiques et propriétés d'un agent :

L'avancement des travaux en IAD et SMA a conduit les chercheurs à définir la notion d'agent mais aussi quelques-unes de ces caractéristiques. Il peut être [27] :

1) Autonome : Son comportement est fonction de ses perceptions qui agisse sur son état, et de sa représentation de l'environnement dans lequel il évolue. Aucun super contrôleur ne peut le piloter de l'extérieur.
2) Proactif : Il peut prendre des initiatives afin de satisfaire ses buts. Pour se faire, il n'est pas soumis à l'invocation d'une autre entité pour agir mais peut agir sur sa propre initiative.
3) Flexible : Il adapte son comportement à sa perception de son environnement et peut participer à des organisations (groupe) afin de mieux satisfaire son but.
4) Social : Il a la capacité d'interagir pour atteindre ses buts ou pour aider d'autres agents dans leurs activités.
5) Situé : Capacité à percevoir l'environnement au travers de métriques spatio-temporels dans lequel il peut agir de façon limitée.
6) intentionnel : c'est un agent guidé par ces buts. Une intention est la déclaration explicite des buts et des moyens d'y parvenir. Elle exprime donc la volonté d'un agent d'atteindre un but ou d'effectuer une action.
   Comme il peut être : [28]
7) rationnel : Les agents rationnels disposent de critères d'évaluation de leurs actions, et sélectionnent selon des critères les meilleures actions qui leur permettent d'atteindre le but. De tels agents sont capables de justifier leurs décisions. La notion de rationalité se rapporte au comportement cognitif de l'agent. Ce terme qualifie l'utilisation efficace des ressources par un agent.
8) Engagement : elle est l'une des qualités essentielles des agents coopératifs. Un agent coopératif planifie ses actions par coordination et négociation avec les autres agents. En construisant un plan pour atteindre un but, l'agent se donne les moyens d'y parvenir et donc s'engage à accomplir les actions qui satisfont leurs buts.

9) adaptatif : il est un agent capable de contrôler ces aptitudes (communicationnelles, comportementales, etc.). Un agent adaptatif est un agent de haut niveau de flexibilité.

III.2.3. Classification des différents types d'agents :

On observe différents types d'agents aux attributs spécifiques [29]:

- Les agents stationnaires : privilégiant un dialogue avec les bases de données
- Les agents mobiles : capables de se déplacer sur le réseau
- Les agents réactifs : agissant en grand nombre pour voir émerger des organisations complexes (image de la fourmilière)
- Les agents cognitifs : possédant des capacités de planification, de communication évoluée (intentionnalité de l'agent BDI : croyances, désirs et intentions).
- Les agents hybrides : combinent les deux philosophies: cognitif et réactif, au sein d'un même agent, afin de « réagir en réfléchissant ».
- Les agents interfaces : interagissant avec l'utilisateur selon un couple où l'utilisateur est engagé dans un processus coopératif
- D'autres types d'agents spécifiques au service web (mobile ou non) sont proposés : Les agents de recherche du Web, Les agents serveur du Web, Les agents de filtrage d'informations, Les agents de recherches documentaires. [28]

III.2.4. Différentes catégories et modèles d'agents :

Après avoir défini les agents, nous présentons dans cette partie les différents modèles d'agents, afin de comprendre leurs caractéristiques et leurs modes de fonctionnement. Nous distinguons deux grandes familles d'agents : les agents réactifs et les agents cognitifs.

Agents cognitifs :

C'est le premier modèle d'agents qui a été proposé. Il est nommé aussi agent délibératif. Il permet de planifier les actions d'un agent au sein de son environnement.

Généralement, ils coopèrent les uns avec les autres pour atteindre un but commun. Ils réagissent en fonction de leurs connaissances, leurs buts, de leurs échanges d'informations avec les autres agents et de la perception de l'environnement. Ils sont dotés de moyens et mécanismes de communication pour gérer les interactions avec d'autres agents (coopération, coordination et négociation) [30].

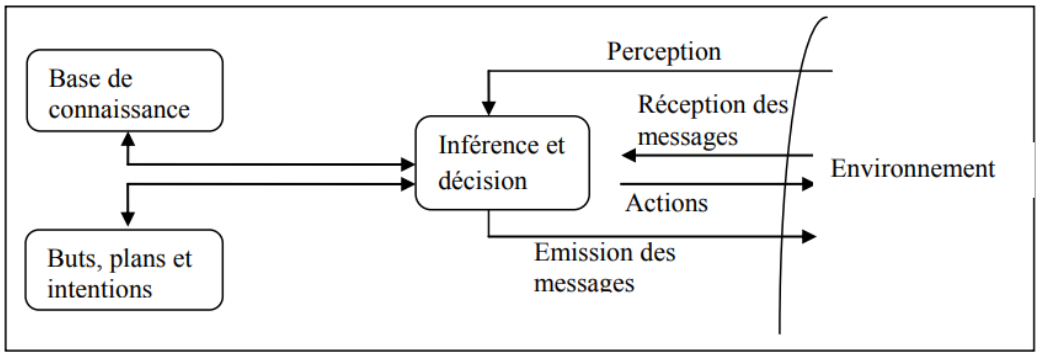


Figure III.2 Le modèle d'un agent cognitif.

L'agent cognitif traite généralement des informations qualitatives. Ces traitements peuvent être établis par l'intermédiaire des outils comme les Classifieurs Génétiques/ Neuronaux (CG/ CN) ou des Systèmes d'Expert (SE). Ces agents peuvent utiliser des mécanismes comme les systèmes à base de raisonnement par cas « Case Based Reasoning » (CBR), les systèmes à base de connaissances« Knowledge Base System » (KBS), les théories des jeux «Game Theory» (GT).

Agents réactifs :

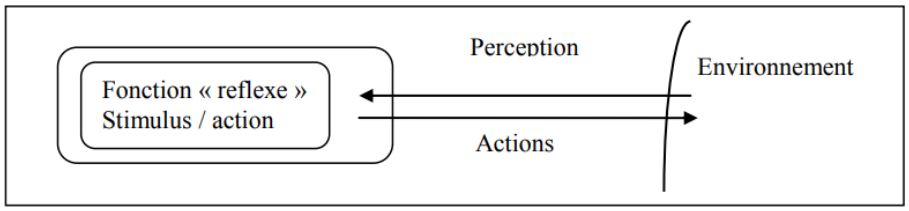
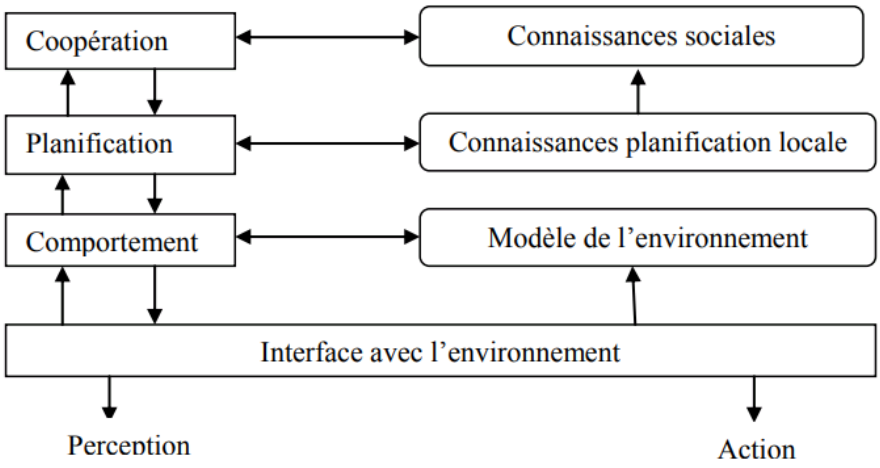


Figure III.3 Le modèle d'un agent réactif.

Ces agents ne font pas de planification de leurs actions. Ce sont des agents qui réagissent uniquement à leur perception de l'environnement et qui agissent en fonction d'elle. Ils traitent généralement des informations quantitatives tout en utilisant des calculs élémentaires ou d'optimisation, ils sont les plus simples à implémenter, mais ils présentent toutefois des limites dues aux points suivants [30]:

- L'agent n'a pas la représentation mentale de l'environnement, et doit choisir ses bonnes actions à partir des données locales uniquement
- Il n'est pas toujours possible de concevoir un comportement en fonction du but spécifié.

Agents hybrides :



III.4 Le modèle d'agents InteRRap

Les agents hybrides sont conçus pour combiner des capacités réactives à des capacités cognitives, ce qui leur permet d'adapter leur comportement en temps réel à l'évolution de l'environnement. Dans le modèle hybride, un agent est composé généralement de trois couches. Au plus bas niveau, on retrouve habituellement une couche purement réactive. La couche intermédiaire fait abstraction des données brutes et travaille plutôt avec une vision des connaissances de l'environnement. Finalement, la couche supérieure se charge des aspects sociaux de l'environnement (communication, coopération, négociation). Le modèle InteRRap proposé par Müller et Pischel, figure parmi les exemples du modèle hybride. [30]

III.3 Le concept des systèmes multi agents :

III.3.1 Définition des SMA :

On appelle système multi-agent (ou SMA), un système composé des éléments suivants [24]:

a. Un environnement, c'est-à-dire un espace disposant généralement d'une métrique.
b. Un ensemble d'objets situés dans l'espace. Ces objets sont passifs, c'est-à-dire qu'ils peuvent être perçus, créés, détruits et modifiés par les agents.
c. Un ensemble d'agents, qui représentent les entités actives du système.
d. Un ensemble de relations qui unissent des objets (et donc des agents) entre eux.
e. Un ensemble d'opérations permettant aux agents de percevoir, produire, consommer, transformer et manipuler des objets.
f. Des opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification, que l'on appellera les lois de l'univers.

Suivant les cas, les comportements des agents sont plus ou moins complexes et rationnels et l'organisation est plus ou moins adaptative. Les agents sont en général situés dans un

environnement (par exemple, topologique) contenant également des entités passives, manipulées par les agents (par exemple, des ressources, des données, des objets physiques…) et communément appelées objets. Chaque agent n'a qu'une connaissance partielle de son environnement et des autres agents. Un système multi-agent est donc intrinsèquement décentralisé [31].

III.3.2. Les étapes de la réalisation d'un SMA : [24]

1. Déterminer les agents et l'environnement
2. Décrire les lois de l'environnement
3. Identifier les perceptions et les influences (actions) produites par les agents
4. Déterminer les variables internes et capacités des agents
5. Définir les comportements des agents: Si les agents sont cognitifs: décrire la relation entre croyances, buts et actions et si les agents sont réactifs: décrire les stimuli, les tropismes (attraction, répulsion, évitement) ainsi que les tâches (combinaisons d'actions élémentaires).

III.3.3. Objectifs de travailler au niveau d'un système multi agents :

Définir et maîtriser différents modes d'interaction entre agents applicables dans la résolution de nombreux problèmes : la coexistence, la coordination, la coopération, la collaboration, la compétition, l'émergence, l'adaptation à la réalité, l'intégration d'expertise incomplète, la modularité, l'efficacité, la fiabilité, la réutilisation. [27]

III.3.4. Caractéristiques d'un SMA : [31]

- Chaque agent a des informations ou des capacités de résolution de problèmes limitées, ainsi chaque agent a un point de vue partiel.
- Il n'y a aucun contrôle global du système multi agents.
- Les données sont décentralisées.
- Le calcul est asynchrone.
- Un SMA peut-être [26]:
  - ✓ Ouvert : les agents y entrent et en sortent librement (ex: un café)
  - ✓ Fermé : l'ensemble d'agents reste le même (ex: un match de football)
  - ✓ Homogène : tous les agents sont construits sur le même modèle (ex: une colonie de fourmis)
  - ✓ Hétérogène : des agents de modèles différents, de granularité différentes (ex: l'organisation d'une entreprise)

III.3.5. Niveaux d'organisation : [24]

On peut distinguer trois niveaux d'organisation dans les systèmes multi-agents :

a. Le niveau micro- social : on s'intéresse aux interactions entre agents et aux différentes formes de liaison qui existent entre un petit nombre d'agents.

b. Le niveau des groupes : on s'intéresse aux structures intermédiaires et on étudie les différenciations des rôles et des activités des agents, l'émergence de structures organisatrices entre agents.
c. Le niveau des sociétés globales (ou populations) : l'intérêt se porte surtout sur la dynamique d'un grand nombre d'agents, ainsi que sur la structure générale du système et son évolution.

III.3.6. Différents types des SMA :

Les différents systèmes existants peuvent être composés d'agents réactifs ou cognitifs, suivant le problème traité. Ces systèmes peuvent appartenir à trois grandes catégories : Les systèmes multi experts, les systèmes multi robot et les systèmes multi agents de simulation. [25].

- Les systèmes **multi experts** vont modéliser l'interaction de plusieurs agents cognitifs, spécialistes de leur domaine et requis pour l'accomplissement d'une tâche complexe. Dans ce cas, les agents sont virtuels, ils n'existent pas physiquement.
- Les systèmes **multi robots** sont des systèmes regroupant des agents artificiels ayant une existence physique et engagée dans une tache commune. Ce sont des robots chargés de tache collective, comme le ramassage de minéral, …
- Les systèmes de simulation sont des systèmes qui servent de support à la modélisation de phénomènes de sociétés animales, en particulier en biologie et en éthologie. Dans ce cas, les agents concernés sont des agents réactifs et les simulations vont concerner des modèles biologiques à tester, comme la reproduction, ou l'influence des contraintes environnementales sur la société animale ou cellulaire.

Les systèmes multi-agent sont actuellement en plein essor. Il s'agit d'une nouvelle approche qui s'intéresse aux comportements individuels, aux interactions entre des entités autonomes et à l'émergence au niveau supérieur de l'ensemble du système de comportement complexe. Le formalisme de cette approche n'est pas encore fixé et fait l'objet de recherches.

Les applications très diverses des systèmes multi-agent ne facilitent pas la définition d'un formalisme définitif tant sur les principes que sur l'architecture des développements informatiques. [25]

III.4 Les interactions et communications :

L'agent fournit une structure spécifiant les protocoles de communication et d'interaction. L'environnement dans lequel l'agent évolue est ouvert, non centralisé et contient des agents autonomes et distribués qui peuvent agir soit pour leur intérêt personnel, soit en coopération avec les autres agents de l'environnement.

III.4.1. Définition d'interaction :

Une interaction est la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales. [24]

Les interactions sont variées et différentes selon le type d'agent et d'organisation. Nous pouvons trouver des interactions de coopération, de compétition et de coordination entre les agents. La coopération est nécessaire à un agent pour atteindre un de ses buts. La coordination, quant à elle, permet d'améliorer le fonctionnement global du système.

III.4.2. Différentes formes d'interaction : [27]

- Interaction directe : Un agent communique par envoi de messages asynchrone vers un autre agent ou ensemble d'agents.
- Interaction indirecte : la communication est réalisée au travers de l'environnement

III.4.3. Types de messages :

Pour communiquer, les agents doivent pouvoir envoyer et recevoir des messages à travers un réseau de communication. L'agent peut avoir les rôles passif, actif (les deux) et les fonctions de maître, esclave.

Dans ce contexte, les messages peuvent être de différents types. Les 2 types minimaux sont la requête et la réponse. N'importe quel agent (quel que soit son rôle) peut accepter ou refuser un message et donc recevoir une assertion. Pour tenir un rôle passif, un agent doit être capable de répondre, i.e. accepter une question externe et envoyer la réponse à la source (assertion).

Pour tenir un rôle actif, un agent doit pouvoir poser une question et faire des assertions. Il peut ainsi contrôler un autre agent par le biais des questions/réponses. Dans un fonctionnement de pair, l'agent a les deux rôles. La théorie des actes de langages a mis en évidence une typologie des messages. [32]

III.4.4. Niveaux de communication :

Les protocoles de communications sont définis à plusieurs niveaux. Le plus bas définit les méthodes d'interconnexion, celui intermédiaire, le format, la syntaxe ou le type de transfert ; au plus haut on retrouve la compréhension et la sémantique, cette dernière faisant référence (entre autres) au type du message. Un protocole peut être binaire ou n-aire, on peut le définir avec une structure de donnée à 5 champs : émetteur, le ou les récepteur, le langage du protocole, l'encodage et le décodage des informations, les actions à entreprendre par le (les) récepteurs. [32]

III.4.5. Définition de la COORDINATION dans les SMA :

Processus par lequel un ou plusieurs agents résonnent sur leurs actions locales et sur les actions des autres (par anticipation) pour assurer la cohérence des actions [33].

D'un point de vue multi agents, l'objectif de la coordination est de s'assurer que les activités des agents permettent de résoudre toutes les composantes du problème globales, les interactions entre les agents sont cohérentes et s'intègrent dans la solution globale, les groupes définis sont cohérents. [34]

II.4.6. Définition de la COLLABORATION dans les SMA :

La collaboration est caractérisée par trois espaces tels que l'espace de communication, l'espace de coordination et l'espace de production.

Une collaboration est un travail en commun; un travail entre plusieurs personnes ou agents qui produisent un résultat commun (produit final). Pour mener ce travail convenablement, les agents doivent se coordonner et communiquer ensemble. Pour coordonner, les agents doivent suivre l'activité des autres participants pour l'utilisation et le partage de la ressource commune. La coordination peut aussi avoir lieu à travers la communication entre agents. Suite à ceci, la communication entre les différents membres de l'équipe est primordiale pour le succès du travail collaboratif. [30]

II.4.7. Définition de la COOPERATION dans les SMA :

Les agents travaillent à la satisfaction d'un but commun, ou individuel dans le but d'améliorer le mode de travail des agents en termes de [35]:

- Validité et rationalité des informations échangées et des comportements,
- Efficacité des stratégies de résolution employées,
- Cohérence entre planification locale et globale,
- Rééquilibrage dynamique de la charge de travail.

Donc la coopération = collaboration + coordination + résolution de conflits avec l'ajout des techniques de négociation qui sont utilisées pour limiter les effets des conflits qui apparaissent. [27]

III.4.8. Actes de langage :

La théorie des actes de langage est un cadre d'analyse des échanges langagiers entre humains. Elle considère la communication comme des actions de requêtes, suggestion, engagement, réponse, etc. Un acte de langage possède 3 caractéristiques : la locution (le phénomène physique), l'illocution (l'intention que veut faire comprendre le locuteur en faisant cette phrase) et la per locution (l'action qui résulte). [32]

Les actes de langages sont une façon de définir le type des messages et de contraindre la sémantique de la communication.

KQML (Knowledge Query and Manipulation Language):

KQML est un protocole pour échanger de l'information entre agents. Son principal atout est que tout ce qui est nécessaire à la compréhension du message est inclus dans le message lui-même. [32]

Knowledge Interchange Format (KIF):

KIF est un langage logique pour la description dans le cadre de systèmes experts, bases de données, ... Il a été conçu comme langage intermédiaire, lisible par un programme et par un humain. La description du langage inclut une spécification pour la syntaxe et une pour la sémantique. KIF reste un langage se fondant sur la logique du 1er ordre. [32]

III.4.9. Protocoles d'interaction :

Dans la section précédente, il s'agissait de définir la communication entre les agents (Echange d'un seul message). Cette section s'attache quant à elle à la description des protocoles d'interaction (Echange d'une série de messages (conversation)). Il y a deux types d'interactions [32]:

- Agents concurrents : il faut maximiser l'utilité de chaque agent.
- Agents ayant des buts semblables ou des problèmes communs : Les aspects importants de ce type d'interaction sont : déterminer les buts communs, déterminer les tâches communes, éviter les conflits et mettre en commun les connaissances.

III.4.10. Négociation :

La négociation est un processus par lequel une décision commune à deux agents ou plus est prise ; chacun d'entre eux essayant d'atteindre leurs buts ou objectifs personnels. Les agents communiquent leur position (source du conflit) et se déplacent en faisant des concessions et en cherchant des alternatives.

Les principaux éléments sont le langage utilisé par les agents, le protocole suivi par les agents lors des négociations et le processus utilisé par chaque agent pour déterminer sa position, les concessions possibles et les critères d'accord. [32]

III.6. Conclusion

Dans ce chapitre nous avons détaillé les différents concepts concernant les agents et les systèmes multi agents avec leurs fonctions.

Après avoir connaitre tout celui-ci, les agents doivent coordonner leurs actions et avoir des mécanismes pour la résolution des conflits. Le mécanisme favori pour la résolution des conflits et la coordination, inspiré du modèle des humains, est la négociation.

Chapitre IV Conception et implémentation

IV.1 Introduction

Notre travail vise à proposer une modélisation multi-agents pour concevoir un système de recherche dans le contexte des big data. Il s'agit de faire une agentification et définir un ensemble d'agents couvant l'ensemble des tâches et des opérations nécessaires pour réussir une recherche efficace d'une ou de plusieurs informations dans une quantité gigantesque de données stockées dans une technologie Big Data. Nous appliquons la solution proposée dans l'écosystème Hadoop.

IV.2 Système proposé.

Parmi les méthodes qui couvrent le mieux le cycle de développement d'un système multi-agent, nous avons choisis la méthodologie AUML pour modéliser notre système. AMUL est basée essentiellement sur la notation standardisée UML (Unified Modeling Language) qui est le résultat de la collaboration d'un groupe de spécialistes du domaine du génie logiciel. AUML décrit le comportement d'agents grâce à des diagrammes d'activité, des graphes d'états et des diagrammes de séquence.

IV.3 Identification des agents

L'agent de coordination :

C'est l'acteur le plus important dans notre système, car c'est lui le responsable de la recherche des ressources de données pour un domaine spécifié, communiquer avec les autres agents du système et retourner le résultat finale, donc il contrôle le cycle de l'activité du système. Un Agent de coordination est chargé d'effectuer les tâches suivantes :

- Faire la recherche de ressources de différent type de données pour un sujet
- Collecte le résultat de recherche par type de donnée
- Envoyer les données aux agents de traitement associées selon leur type de données spécifié a traité
- Réception des résultats de traitement de chaque agent appelé à faire un traitement
- Combiner les résultats des agents de traitement pour fournir un résultat final

Les agents de traitement :

Ces agents reçoivent les ressources de données fournis par l'agent de coordination, faires leur traitement des données par l'utilisation de la technique Map-Reduce, et envoient leurs résultats a l'agent de coordination :

- **Agent de traitement des données de type texte**
- **Agent de traitement des données de type audio**

- **Agent de traitement des données de type vidéo**
- **Agent de traitement des données de type image**
- **Agent de traitement des données de type structurées**
- **Agent de traitement des données de type Autre**

IV.4 Architecture générales du système

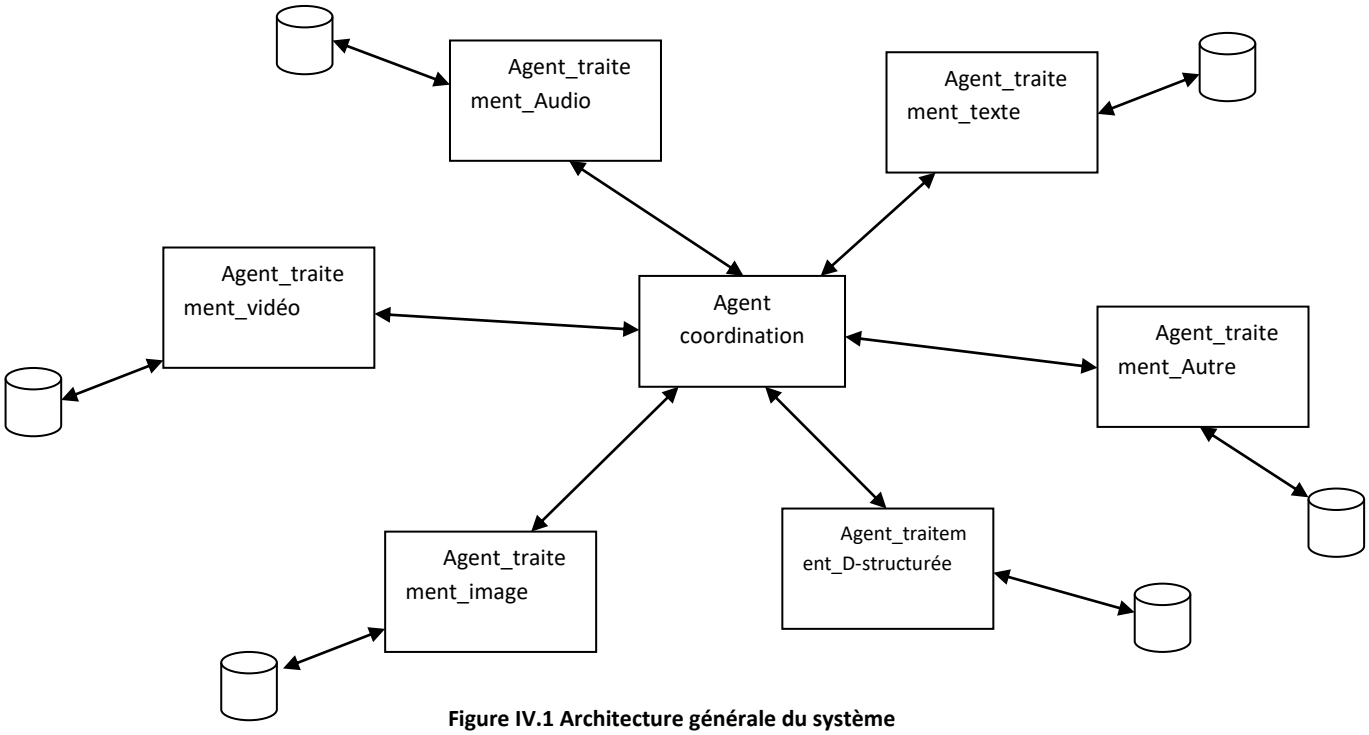


Figure IV.1 Architecture générale du système

IV.5 Diagramme de séquence

Le diagramme de séquence est une solution populaire de modélisation dynamique en langage UML, car il se concentrent plus précisément sur les lignes de vie, les processus et les objets qui vivent simultanément, et les messages qu'ils échangent entre eux pour exercer une fonction avant la fin de la ligne de vie.

La figure VII.1 montre le diagramme de séquence du système ou c1, c2,…..c12 sont les messages échangés entre les agents :

- C1 : l'agent de coordination envoi les données à l'agent de traitement de données de type texte
- C2 : l'agent de coordination envoi les données à l'agent de traitement de données de type vidéo
- C3 : l'agent de coordination envoi les données à l'agent de traitement de données de type image

- C4 : l'agent de coordination envoi les données à l'agent de traitement de données de type audio
- C5 : l'agent de coordination envoi les données à l'agent de traitement de données de type structuré
- C6 : l'agent de coordination envoi les données à l'agent de traitement de données de type autre
- C7, C8,…, C12 : envoi du résultat de traitement des données des agents de traitement à l'agent de coordination

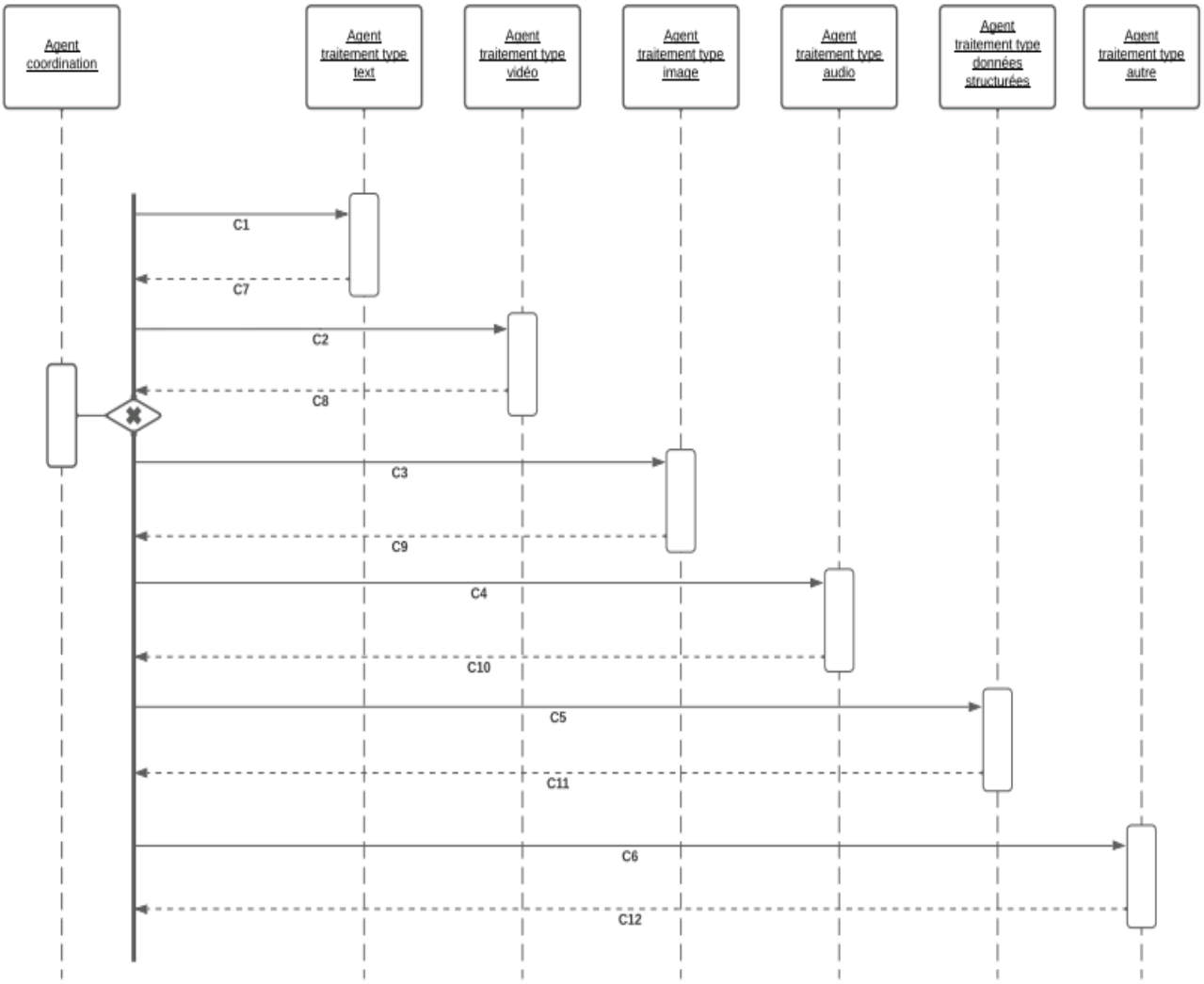


Figure IV.2 Diagramme de séquence du système

IV.6 Diagramme d'activité du système

Un diagramme d'activités est un organigramme illustrant les activités exécutées par un système.

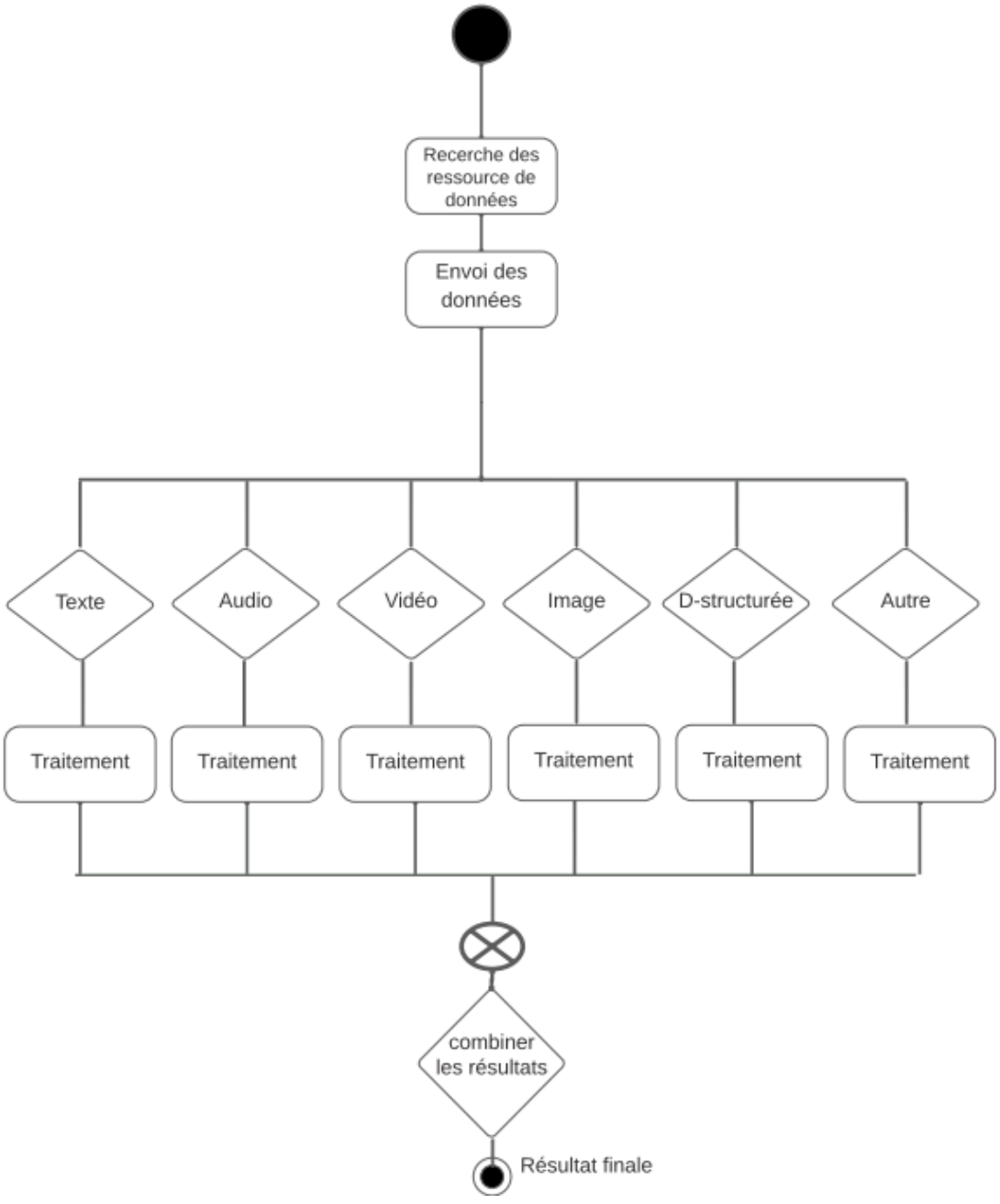


Figure IV.3 Diagramme d'activité du système

IV.7 Implémentation :

IV.7.1 Installation du cloudera :

- Avant de configurer la machine virtuelle Cloudera, vous devez disposer d'une machine virtuelle telle que VMware ou Oracle VirtualBox sur votre système.

- Nous utilisons Oracle VirtualBox pour configurer la machine virtuelle Cloudera QuickStart.
- Pour configurer la VM Cloudera QuickStart dans votre gestionnaire Oracle VirtualBox, cliquez sur « File », puis sélectionnez « Import appliance».

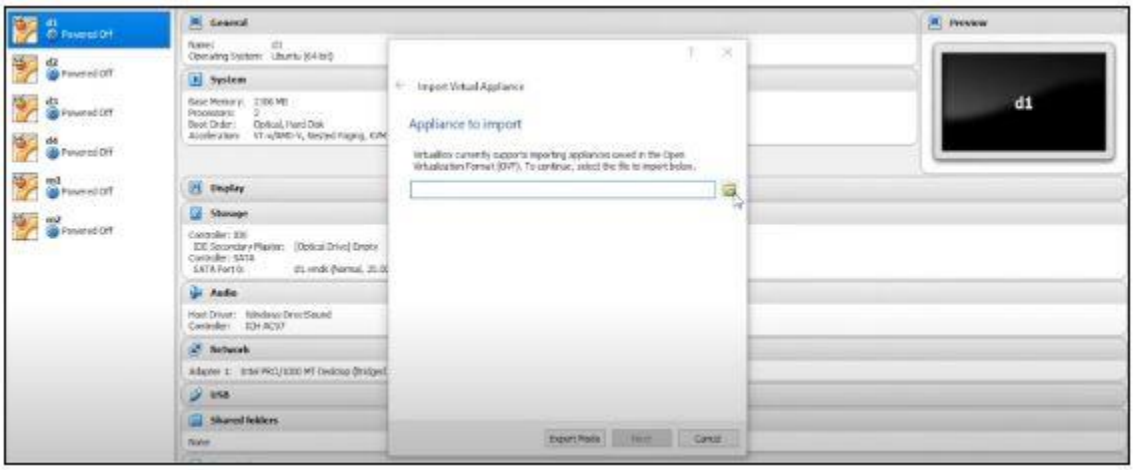


Figure IV.4 Importation de l'image de la machine virtuelle Cloudera QuickStart

- Choisissez l'image QuickStart VM en consultant vos téléchargements. Cliquez sur « Ouvrir » puis sur « Next». Vous pouvez maintenant voir les spécifications, puis cliquez sur "Import". Cela commencera à importer le fichier .vmdk de l'image du disque virtuel dans votre boîte VM.

- L'étape suivante consiste à configurer une machine virtuelle Cloudera QuickStart pour s'entraîner. Une fois l'importation terminée, vous pouvez voir la machine virtuelle Cloudera QuickStart dans le panneau de gauche.

Figure IV.5 La configuration du Cloudera VM a réussi

- La prochaine étape consistera à démarrer la machine en cliquant sur le symbole "Start" en haut.
- Une fois votre machine allumée, elle ressemblera à ceci :

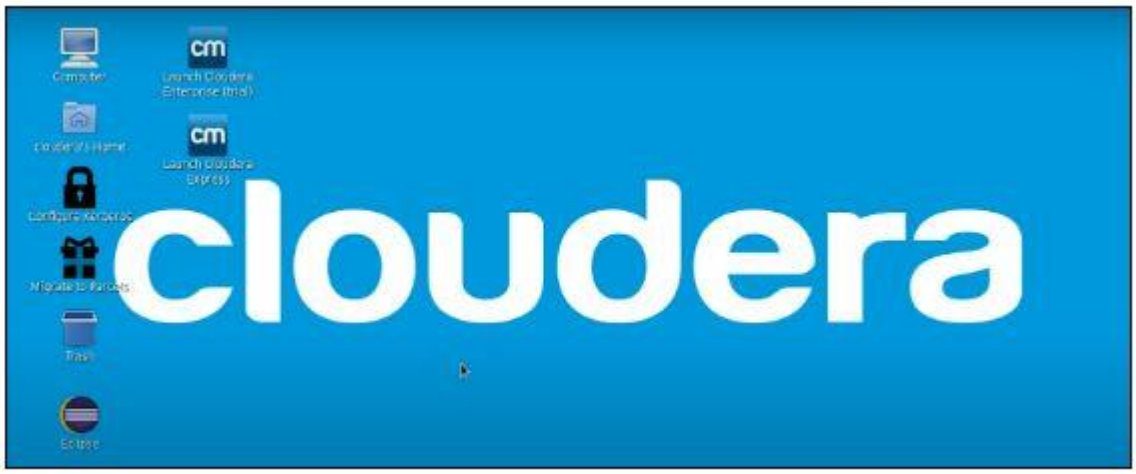


Figure IV.6 Fenêtre de cloudera

- Une fois que vous voyez que votre accès HDFS fonctionne correctement, vous pouvez fermer le terminal. Ensuite, vous devez cliquer sur l'icône suivante " Launch Cloudera Express", un écran apparaîtra avec la commande suivante :

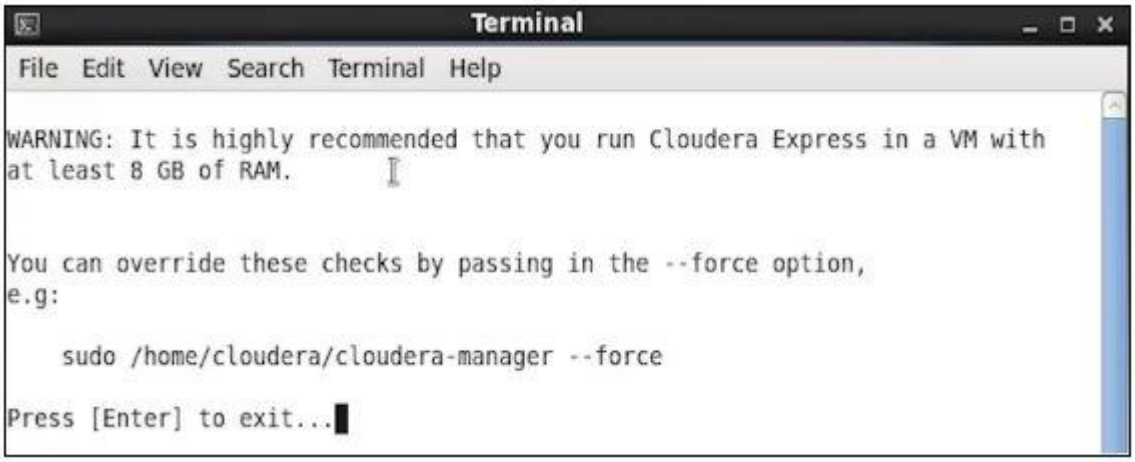


Figure IV.7 Terminal

- Vous devez copier la commande et l'exécuter sur un terminal séparé. Par conséquent, ouvrez un nouveau terminal et utilisez la commande ci-dessous pour fermer les services basés sur Cloudera. Il redémarrera les services, après quoi vous pourrez accéder à votre console d'administration.

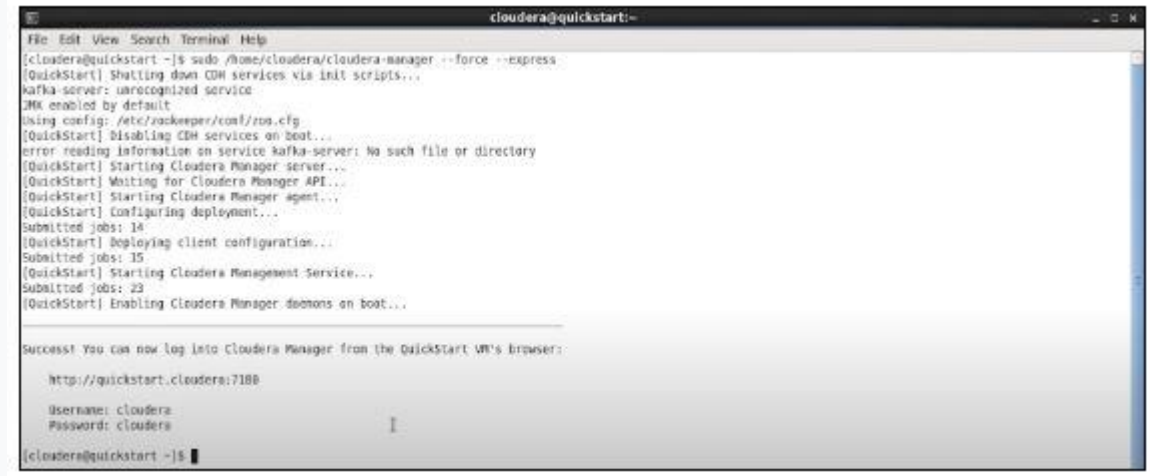


Figure IV.8 Redémarrage des services sur Cloudera QuickStart VM

- Maintenant que notre déploiement a été configuré, les configurations client ont également été déployées. De plus, il a redémarré le Cloudera Management Service, qui donne accès à la console d'administration Cloudera QuickStart à l'aide d'un nom d'utilisateur et d'un mot de passe.

IV.7.2 Démarrage de hadoop :

Hadoop peut être démarré en exécutant sur le name node en tant qu'utilisateur "hduser", la commande suivante :

```
hduser@NameNode:~% start-dfs.sh
```

IV.7.3 Implémentation d' un exemple en python sur Hadoop :

Dans cette phase, on va implémenter un exemple d'application MapReduce pour obtenir un aperçu de son fonctionnement. Cet exemple est une application simple qui cherche le total des ventes par magasin. Ce programme est le code de base utilisé pour comprendre le fonctionnement du paradigme de programmation de MapReduce.

- en utilisons un fichier input sous la forme suivante :

  date | temps | magasin| produit | cout | paiement

- Code mapper :

```
3  import sys
4  wordList = dict()
5  for line in sys.stdin:
6      line = line.strip()
7      words = line.split('|')
8
9      print '%s\t%s' % (words[2],words[4])# afficher cle valeur
```

Figure IV.9 mapper.py

- Code reducer :

```
from operator import itemgetter
import sys

current_magazin = None
current_prix = 0
magazin = None

wordList = dict()

for line in sys.stdin:

    line = line.strip()
    magazin, prix = line.split('\t', 1)

    try:
        prix = int(prix)
    except ValueError:
            continue

    if current_magazin == magazin:
        current_prix += prix
    else:
        if current_magazin:

            print '%s\t%s' % (current_magazin, current_prix)
        current_magazin = magazin
        current_prix = prix

if current_magazin == magazin:
    print '%s\t%s' % (current_magazin, current_prix)
```

Figure IV.10 reducer.py

Pour exécuter cet exemple on pace par les étapes suivantes :

- Copier le fichier dans le chemin suivant : /home/cloudera
- Créer un répertoire input dans HDFS : hadoop fs -mkdir /user/input
- Transférer le fichier input.txt du local vers HDFS : hadoop fs -copyFromLocal /home/cloudera/Exemple/input.txt input/
- Execution du programme Mapreduce :

```
hadoop jar /usr/lib/hadoop-0.20-
mapreduce/contrib/streaming/hadoop-streaming-2.6.0-mr1-
cdh5.12.0.jar
-Dmaperd.reduce,tasks=1
-file /home/cloudera/Exemple/mapper.py
-mapper "python /home/cloudera/Exemple/mapper.py"
-file /home/cloudera/Exemple/reducer.py
-reducer "python /home/cloudera/Exemple/reducer.py"
-input Exemple/input.txt #exemple
-output out
```

- Pour afficher le contenu du fichier output : hadoop fs -cat out/part-00000
- Pour supprimer le répertoire output : hadoop fs -rm -r out

IV.8 Conclusion :

Dans ce chapitre nous avons proposé un système multi agents. Définir un ensemble d'agents couvrant l'ensemble des tâches et des opérations nécessaires pour faire une recherche efficace d'une ou de plusieurs informations dans une grande quantité de données stockées dans une technologie Big Data.

Conclusion générale :

Les données devenant de plus en plus volumineuse et complexe, les bases de données traditionnelles sont limitées face à l'analyse et au traitement de ces données. Actuellement, de nombreuses recherches sont en cours dans ce domaine. A mesure que les données augmentent à un rythme plus rapide, il existe un besoin énorme d'outils et de technologies capables de les gérer.

On a utilisé les systèmes multi agents pour simplifier notre problème de recherche et traitement du big data. Les systèmes multi-agents ont des applications dans le domaine de l'intelligence artificielle où ils permettent de réduire la complexité de la résolution d'un problème en divisant le savoir nécessaire en sous-ensembles.

Références bibliographiques

[1] [BRASSEUR (C.). – Enjeux et usages du big data. Technologies, m´ethodes et mises en oeuvre, Paris, Lavoisier, p. 30 (2013)]

[2] [HELBING (D.) et POURNARAS (E.). – Build Digital Democracy : Open Sharing of Data that are Collected with Smart Devices would Empower Citizens and Create Jobs.Nature, Vol.527, Nov. 2015, Macmillan Publishers (2015)]

[3] http://www.researchgate.net/publication/279848651-Rapport-sur-le-Big-Data

[4] Journal of king saud university – computer and information sciences, p 433-434.

[5] http://www.redsen-consulting.com/2013/06/big-data.

[6] http ://big-data-iscomwiz.e-monsite.com/iscomwiz/le-big-data/les-5-v-du-bigdataa-connaitre.html

[7] http ://big-data-iscomwiz.emonsite.com/iscomwiz/le-big-data/les-5-v-du-big-dataa-connaitre.html

[8] https://www.oracle.com/fr/big-data/guide/what-is-big-data.htm.

[9] https ://www.filfil.eu/2019/05/13/le-big-data/.

[10] http://www.tutorialspoint.com/hadoop/hadoop_quick_guide.html.

[11] Hao Zhang, Gang Chen, In-Memory Big Data Management and Processing: A Survey.

[12] Tawfik Bourgi, Nesrine Zoghlami, Big Data for Transport and Logistics.

[13]M.CORINUS, T.Derey, J.Marguerie, W.Techer, N.Vic, Rapport d'étude sur le Big Data, SRS Day, p54, 2012.

[14] https://www.piloter.org/business-intelligence/technologie-bigdata.html.

[15] http://fr.wikipedia.org/wiki/Entrep%C3%B4t_de_donn%C3%A9es.

[16] Big data application and architecture ; de himanshu et soumendramohanty.

[17] Mekideche Mounir, Conception et implémentation d'un moteur de recherche à base d'une architecture Hadoop (Big Data), Avril 2015.

[18] https://fr.wikipedia.org/wiki/Hadoop#Historique

[20]http://mbaron.developpez.com/tutoriels/bigdata/hadoop/introduction-hdfs-mapreduce/

[21] Jonathan Lejeune, Hadoop:uneplate-formed'exécution de programme Map-reduce, École des Mines de Nantes, 83p, Janvier 2015.

[22]Benjamin Renaut, Hadoop/Big Data, Université de Nice Sophia-Antipolis, 114p, 2013-2014

[23] L. R. JDN, « Hbase : le nosql au service du big data »

[24] Jacques FERBER : Les systèmes multi-agents : Vers une intelligence collective. Inter Editions, Paris, France. 1995.

[25] Jean-Pierre MÜLLER : Des systèmes autonomes aux systèmes multi-agents : Interaction, émergence et systèmes complexes. Université Montpellier II. Mémoire d'habilitation, France. 8 novembre 2002.

[26] Jacques FERBER : La simulation multi agent (Agent based simulation). Partie1. LIRMM. Version 1.0. Université Montpellier II, France, Janv. 2009.

[27] http://personnel.univ-reunion.fr/courdier/cours/sma/2_agent_et_sma.pdf

[28] Alper CAGLAYAN and Colin HARRISON : Agent Sourcebook. Edition John Wiley & Sons. Canada, 1997.

[29] Joël QUINQUETON. LIRMM et CERIC : Introduction aux systèmes multiagent (SMA). Université de Montpellier, France.2005.

[30] Nardjes KHEZAMI : Un système multi agents pour l'analyse, la conception et l'évaluation de la collaboration appliquée au télétravail collaboratif via internet. Thèse de doctorat. Université d'Evry Val d'Essone, France.2005.

[31] Christine BOURJOT : SystèmesMulti-Agents:Modélisationet simulation informatique de comportements collectifs. Un Cours sur les SMA. Université de Nancy 2, France, 1998.

[32] Lamia HASSAINE : Concept ion et réalisation d'un système de négociation automatique applique aux ventes en enchère. Mémoire de fin d'études. Ecole nationale Supérieure d'Informatique (ESI) Oued-Smar, Algérie, 2008/2009.

[33] N.R. JENNING: Coordination techniques for distributed artificial intelligence. Foundation of distributed artificial intelligence. Willey and Sons. 1996.

[34] V.R. LESSER: Distributed problem Solving. Encyclopedia of AI. Edition Willey and Sons. 1998.

[35] http://opera.inrialpes.fr/people/Tayeb.Lemlouma/Papers/IAD_Presentation.pdf