UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTÉ DES SCIENCES ET TECHNIQUES FÈS
DÉPARTEMENT D'INFORMATIQUE

PROJET DE FIN D'ETUDES

MASTER SCIENCES ET TECHNIQUES
SYSTÈMES INTELLIGENTS & RÉSEAUX

ETAT DE L'ART DES BIG DATA

LIEU DE STAGE : LABORATOIRE SYSTÈMES INTELLIGENTS ET APPLICATIONS
LABORATOIRE SIGNAUX SYSTÈMES ET COMPOSANTS

RÉALISÉ PAR : TADIST KHAWLA

SOUTENU LE : 25/06/2015

ENCADRÉ PAR :

PR. MRABTI FATIHA

PR. NAJAH SAID

DEVANT LE JURY COMPOSÉ DE :

PR. MRABTI FATIHA

PR. NAJAH SAID

PR. BENABBOU ABDERRAHIM

PR. OUZARF MOHAMED

ANNÉE UNIVERSITAIRE 2014-2015

Dédicace

Je dédie ce Rapport aux institutions les plus formatrices de ma vie :

A mes chers parents

Rien au monde ne pourra vous exprimer mon amour, mon respect et ma reconnaissance pour la tendresse et les grands efforts et sacrifices que vous avez fait pour moi et pendant toute ma vie. Ma plus grande fierté est d'être votre fille.

A mes sœurs

Malgré nos différences, je tiens à exprimer que rien ne pourra jamais compenser votre amour et votre place dans mon cœur. Votre soutien est une dette que je ne pourrai jamais payer.

A Amine

Merci d'être présent dans ma vie.

A mes amis Soukaina, Hajar et Omar

Sachez que vous êtes très chers à mon cœur et que je vous suis très reconnaissant d'avoir toujours été là pour moi.

A mes amis Hamza, Jamal, Bilal et Hicham

Merci d'avoir toujours répondu présents ces deux dernières années et merci encore de les avoir rendues beaucoup plus intéressantes et amusantes qu'elles auraient dues être.

Je vous souhaite une vie pleine de réussite

Remerciement

Tout d’abord, je tiens à remercier Dieu de m’avoir donné la volonté pour réaliser ce travail.

La réalisation de ce projet a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.

Je tiens à remercier dans un premier temps Monsieur le Doyen de la faculté des sciences et techniques, ainsi que toute l’équipe pédagogique du département informatique.

Je remercie les deux laboratoires, laboratoire Systèmes Intelligents et Applications et laboratoire Signaux Systèmes et Composants et plus particulièrement le professeur Fatiha MRABTI pour son aide, sa disponibilité et ses précieux conseils qui m’ont été d’un appui considérable dans ma démarche.

Je remercie mes parents qui ont veillé à mon confort et qui m’ont toujours motivé et aidé.

Résumé

Les ‘Big Data’ sont devenus sans conteste le sujet du moment. L'augmentation continue et rapide du volume des données capturées par plusieurs entités a produit un flux énorme des données. Le problème ne se pose pas seulement dans le volume de données, il concerne également les types des données acquises, qui ne sont pas nécessairement structurées, ce qui peut causer des difficultés lors du traitement.

L'ère de «Big Data» induit en effet certains problèmes, mais il offre aussi de nombreuses solutions afin de faciliter aux organisations de transformer ces énormes quantités de données structurées, semi-structurées et non structurées à des données utiles, qui peuvent aider plusieurs entreprises à mieux comprendre leurs clients.

Afin de déduire de la valeur à partir des données acquises, ces derniers passent par un processus très compliqué qui est effectué grâce à plusieurs outils qui ont été développés afin que chacun accomplisse une tâche précise dans le processus de traitement des Big Data.

Abstract

The ‘Big Data’ is unquestionably one of the most important topics that we should pay attention to. If someone doesn’t know what the term ‘Big Data’ is, you are nowadays characterized as an unknowledgeable person and your chances of ever having a successful organization dealing with important data will be even poorer.

The main problematic in Big Data is that there is a rapid increase of the volume and details of the data captured by different organizations which produces a huge flow of data causing many problems while dealing with it, but the problem does not only arise in the amount of the captured data it also concerns the different types of this data which may stand in the way of gaining a value out of it.

The ‘Big Data’ era does indeed induce problems but it also offers many solutions in order to make it possible for the organizations to gain some knowledge out of the huge amounts of structured, semi-structured and mainly unstructured data and in order for that to happen, many tools have been developed that have helped turning Big Data into gainful information.

Table de matière

Dédicace 2
Remerciement 3
Résumé 4
Abstract 4
Table de matière 5
Liste de Figures 7
Liste des tableaux 7
Liste des acronymes 8
Introduction Générale 9
Chapitre I Introduction aux Big Data 11

Introduction 12

1. Origines et historique 12

2. Caractéristiques des Big Data 14

3. Traitement Cycle de vie des Big Data 15

4. Domaines d'application 20

Conclusion 21
Chapitre II Productions scientifiques Etat de l'art 22

Introduction 23

1. Big Data, contexte général 23

2. Architectures proposées 24

3. Stockage des données 25

4. Analyse des données 26

5. Sécurité 27

6. Big Data et domaine médicale 29

Conclusion 30
Chapitre III Technologies et application 31

Introduction 32

1. Les systèmes NoSQL (Not Only SQL) 32
   1.1. Naissance des bases de données NoSQL 32
   1.2. Idée générale sur les systèmes NoSQL 34

2. Le Framework Hadoop 40
   2.1. Présentation de Hadoop 40
   2.2. Composants Apache Hadoop 45

3. Langage R 48
   3.1. Opérations sur les données 48
   3.2. Modélisation de données dans R 49

4. Combinaison de Hadoop et R 50
   4.1. RHipe 50
   4.2. RHadoop 51
   4.3. Hadoop streaming 52

Conclusion 53

Conclusion générale et perspectives 54
Annexe 1 55
Annexe 4 61
Annexe 5 63
Annexe 6 64
Bibliographie 67
Webographie 68

Liste de Figures

Figure 1. Définitions des Big Data basées sur un sondage en ligne de 154 dirigeants mondiaux [1]. 12
Figure 2. Cycle de vie des Big Data 15
Figure 3. Les raisons les plus courantes de perte de données selon les 'Online Backup Geeks' [5] 18
Figure 4. Fréquence de distribution des documents contenant le terme 'Big Data' [1] 23
Figure 5. Architecture du système viseur d'experts [15] 27
Figure 6. Structure de données générale pour les notes cliniques [18] 29
Figure 7. Schéma explicatif du théorème de CAP [25] 33
Figure 8. Schéma explicatif des bases de données « Clé-Valeur » [25] 35
Figure 9. Schéma explicatif des bases de données « Colonne» [25] 37
Figure 10. Schéma explicatif des bases de données « document » [25] 38
Figure 11. Schéma explicatif des bases de données « Graphe » [25] 39
Figure 12. Architecture de Hadoop 41
Figure 13. Architecture du HDFS [27] 42
Figure 14. Architecture de MapReduce [29] 43
Figure 15. Architecture du Yarn [30] 44
Figure 16. Architecture Hue [34] 48
Figure 17. Composants de RHIPE [26] 51
Figure 18. Ecosystème de Rhadoop [26] 52
Figure 19. Composants de Hadoop streaming [26] 53
Figure 20. Architecture traditionnelle traitant les données structurées [12]. 55
Figure 21. Architecture traditionnelle traitant les données non structurées [12]. 55
Figure 22. Architecture Big Data proposé [12] 55
Figure 23. Architecture CAPIM [13] 56
Figure 24. Flux d'information de surveillance [13] 57
Figure 25. Architecture proposée [13] 57
Figure 26. Conception haut niveau de l'architecture de référence [36] 59
Figure 27. Modélisation de base de données NoSQL: (a) Notion par table (trois concepts présentés), (b) Nom de colonne fixe, (c) paire clé-valeur fixe, et (d) paire clé-valeur généralisée [18] 61
Figure 28. Structure de données XML [18] 62
Figure 29. (a) les données XML sont stockés avec un champ individuel dans la base de données compatible XML, (b) des fichiers XML contenant des données cliniques sont stockées sous forme d'une liste de fichiers dans la base de données XML native [18] 62
Figure 30. Pipeline d'analyse des Big Data en Biomédecine [24] 64

Liste des tableaux

Table 1. Exemples de solutions proposées pour générer, interpréter et visualiser les données biomédicales [24] 65
Table 2. Exemples de grandes entreprises qui offrent des solutions et des pipelines pour stocker, analyser et traiter l'information biomédicale complexe [24]. 65
Table 3. Exemples de compagnies qui offrent la génétique personnalisées et des solutions omiques [24]. 66

Liste des acronymes

ACID	Atomicity, Consistency, Isolation, Durability
ACM	Association for Computing Machinery
ADN	Acide Désoxyribo Nucléique
BI	Business Intelligence
CMS	Content Management System
CPU	Central Processing Unit
DFS	Distributed File System
DME	Durable Medical Equipment
EDW	Enterprise Data Warehouse
EHRs	Electronic Health Records
EMRs	Electronic Medical Records
HDFS	Hadoop Distributed File System
ICT	Information and Communications Technology
IGV	Integrated Genome Viewer
IT	Information Technology
JSON	JavaScript Object Notation
MPP	Massively Parallel Processing
NGS	Next-Generation Sequencing
NoSQL	Not only Structured Query Language
OCR	Optical Character Recognition
SAP	Systems, Applications and Products for Data processing
SGBD	Système de Gestion de Base de Données
SQL	Structured Query Language
XML	Extensible Markup Language

Introduction Générale

Le phénomène des 'Big Data' est né avec l'apparition des flux énormes de données crées par les différents aspects qui nous entourent, ces flux de données doivent être acquises, traitées pour pouvoir en tiré plusieurs déduction qui vont être utile par la suite. Le problème des Big Data ne se limite pas au volume des flux acquis mais aussi au manque de structuration qui les rend plus complexe et moins compréhensibles.

Les Big Data peuvent être considérées comme une révolution prometteuse des traitements des données. Ils ont apporté plusieurs avantages, ils peuvent aider à améliorer la prise de décision, réduire les coûts d'infrastructures informatiques via l'utilisation des serveurs standards et des logiciels open source, développer la réactivité et l'interactivité à l'égard des clients…

Les flux énormes des données générées à la seconde par plusieurs organisations font des étapes de traitement des données acquises un processus très délicat. Les 'Big Data' peuvent être considérés comme une révolution prometteuse des traitements des données.

Les Big Data engendrent plusieurs problématiques, le grand volume des Big Data rend leur stockage une opération très critiques. Les systèmes de stockage traditionnels ne supportent qu'un nombre limité de données, et ne peuvent pas supporter le grand flux de données généré maintenant.

Le risque de perte des données est toujours présent, il faut trouver des dispositifs de stockage qui aide à garder les données en sûreté.

Le stockage des Big Data n'est pas le seul problème, plus le nombre de données augmente plus le fait de les analyser est difficile. De plus dans plusieurs cas, le grand flux de données qui émerge doit être traité en temps réel, qui aussi ne manque pas de difficulté. Une autre opération qui fait partie d'un large et complexe processus de traitement de données.

Le processus de traitement passe par plusieurs étapes :

- Etape d'acquisition : les données sont acquises de différentes sources et peuvent être de différentes formes, texte, image, vidéo… Ce qui rend la structuration de ces données une tâche délicate. Ces données peuvent aussi être en *streaming* (nécessite un traitement à temps réel) ou bien *in situ* (nécessite que le processeur se déplace vers eux pour le traitement).
- Etape de stockage : vu la masse et la nature diverse de ces données, les systèmes de gestion de base de données relationnels ne sont plus suffisantes, d'où la nécessité de nouvelles méthodes de stockage.
- Etape d'analyse : Le but ici est d'extraire de la valeur à partir des données acquises.
- Etape de visualisation : Le résultat des analyses est visualisé de différentes manières tel les recommandations, de nouveaux services offerts….

Les données traitées sont de très grandes importances même quand elles sont encore brutes, avec la compétition brutale entre les organisations utilisant les Big Data, la sécurisation de ces données est une opération essentielle.

Les Big Data touchent presque tous les domaines qui peuvent exister, parmi ces domaines, on trouve le domaine informatique, le management, l'enseignement, le transport, la médecine…

Les technologies de l'information ouvrent la porte aux humains pour qu'ils puissent s'intégrer dans une société intelligente et conduisent à l'élaboration de plusieurs services modernes (e-commerce, logistique moderne et e-finance…)

La technologie moderne de l'information est en train de devenir le moteur de l'exploitation et du développement de tous les horizons de la vie [1], influençant plusieurs domaines, domaine de santé, commerce, finance, affaires, transport... ils ont pu saisir l'attention du milieu universitaire, du gouvernement et de l'industrie.

Durant ces dernières années l'intérêt que donnent les scientifiques aux Big Data augmente de jour en jour, le nombre des œuvres qui ont été publié et qui traitent les Big Data est énorme, dans ce document nous allons vous présenter quelques publications qui traitent des aspects des Big Data dont nous nous sommes aussi intéressés.

Pour traiter les Big Data, plusieurs nouveaux outils ont été présentés par de différentes organisations qui se rivalisent pour trouver l'outil ultime qui va combler tous les problématiques des Big Data. Le premier outil important est les systèmes NoSQL qui présentent de nouveaux horizons au niveau de gestion des bases de données. Le deuxième outil est HADOOP qui est un Framework très puissant qui traite plusieurs aspects des Big Data. Le dernier outil est le langage de statistiques R qui n'est pas seulement utilisé par les Big Data, mais qui a prouvé son efficacité en ce qui concerne l'analyse de ces derniers.

Dans le cadre de notre travail, nous avons établi une étude bibliographique sur les problématiques des Big Data et quelques solutions proposées. Dans le premier chapitre nous avons commencé par introduire les Big Data, sa définition, son origine, et historique, ses caractéristiques, les différentes étapes de traitement des données dans l'ère des Big Data et quelques domaines d'application. Dans le deuxième chapitre, nous avons essayé d'encercler l'état de l'art des Big Data. Pour finir, nous avons présenté les différentes nouvelles technologies utilisées pour le traitement des Big Data.

Chapitre I
Introduction aux Big Data

Introduction

Les Big Data s'imposent comme l'une des évolutions majeures des systèmes d'information, à la fois sur les plans métiers, fonctionnels et technologiques. Le concept Big Data est un écosystème riche et complexe, ce qui rend le fait de trouver une définition exacte et universelle très délicat.

Autrefois, le développement des nouvelles technologies apparaissait dans les publications techniques et académiques en premier lieu, mais le développement empressé et le consentement du concept par les secteurs publics et privés ont laissé peu de temps pour les technologies de mûrir académiquement.

Plusieurs cadres des organisations diffèrent dans leur compréhension des Big Data (figure1).

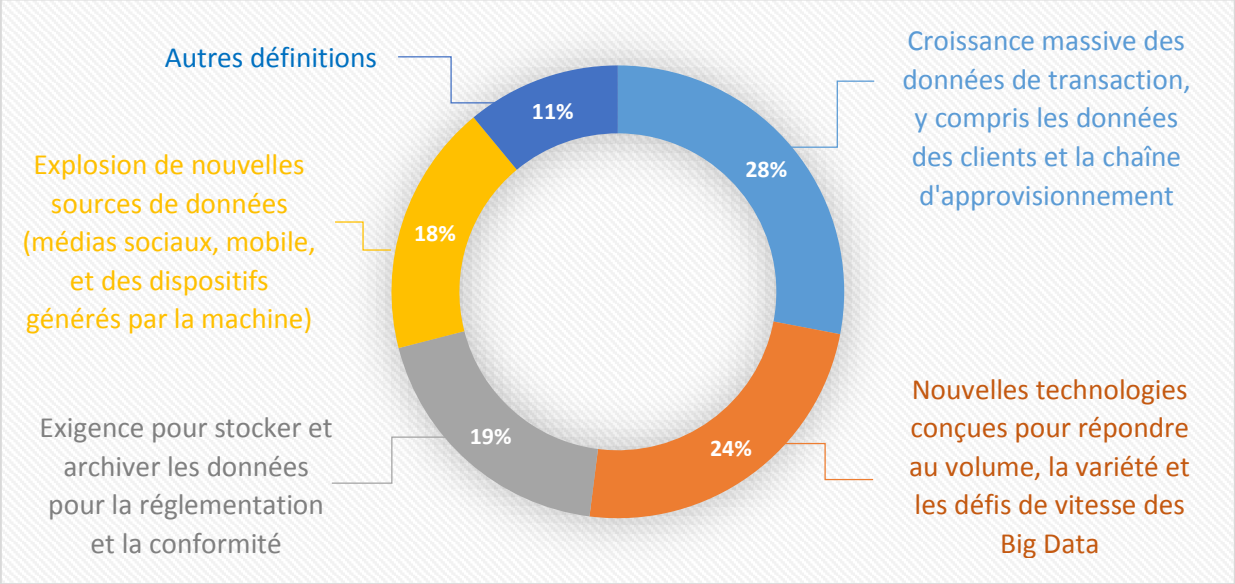


Figure 1. Définitions des Big Data basées sur un sondage en ligne de 154 dirigeants mondiaux en Avril 2012 [1].

1. Origines et historique

Les Big Data sont omniprésent aujourd'hui, toutefois leurs origines restent incertaines. Plusieurs affirment que la naissance des Big Data est causée par les problèmes des grands flux de données générés, rencontrés par les deux acteurs majeurs du Web Google et Yahoo. D'autre estime que la naissance du phénomène est beaucoup plus antérieure.

La différence entre les multiples opinions fait qu'on ne peut pas déterminer exactement quand est ce que ce phénomène a eu naissance, on ne peut que tracer le cours des raisons qui ont causé sa naissance, voici un bref historique des évènements marquants dans la vie des Big Data.

1941: Première utilisation du terme explosion des données, selon le dictionnaire d'Oxford [2].

1944: Fremont Rider[1] dans son œuvre "*The scholar and the future of research library*", estime que la taille des bibliothèques universitaires américaines se doublait tous les seize ans [2], ce qui va créer un grand problème de stockage.

1961: Derek Price[2] dans son œuvre "Science Since Babylon", retrace l'évolution des connaissances scientifiques en analysant la croissance du nombre de revues scientifiques et de documents publiés. Il conclut que le nombre de nouveaux journaux a augmenté de façon exponentielle plutôt que linéaire [2].

1967: B.A. Marron et P.A.D. de Maine dans leurs œuvre "Automatic data compression", déclarent que «L'explosion de l'information est a noté au cours des dernières années, et qu'il est essentiel que les exigences de stockage soient maintenus à un minimum» [2].

1989 : Première utilisation du terme Big data dans un article d'un magazine écrit par Erik Larson[3] [3].

1997: Michael Lesk[4] dans son œuvre "How much information is there in the world?" conclut qu'il peut y avoir quelques milliers de pétaoctets d'information tout dit, et la production de bande et disque atteindra ce niveau d'ici l'an 2000. Ainsi, en seulement quelques années, nous serons en mesure à tout sauver, aucune information ne devra être jetée, et l'ensemble d'information ne pourra jamais être regardé par un seul être humain [2].

1998: K.G. Coffman et Andrew Odlyzko dans son œuvre "The Size and Growth Rate of the Internet", concluent que «le taux de croissance du trafic sur l'Internet public, est d'environ 100% par an, beaucoup plus élevé que pour le trafic sur d'autres réseaux [2].

2001: Doug Laney dans son œuvre "3D Data Management: Controlling Data Volume, Velocity, and Variety", Une décennie plus tard, les «3V's» sont devenus les trois dimensions généralement reconnus définissant les Big Data, bien que le terme lui-même n'apparait pas dans cette note2].

2005: - Apparition de Hadoop, un framework open source des Big Data développé par Apache [3].

- Naissance du Web 2.0 dont le contenu est généré par les utilisateurs [3].

2008 : Globalement 9.57 zettabites (9.57 trillion gigabytes) d'information sont traité par les tous CPU du monde [3].

2014: - L'utilisation d'Internet sur les mobiles dépasse son utilisation sur les ordinateurs de bureau pour la première fois [3].

- 88% des cadres ont répondu à une enquête internationale disant que l'analyse des Big Data est une priorité absolue [3].

---

[1] (25 mai 1885 - 26 Octobre 1962) écrivain, poète, éditeur, inventeur, généalogiste, et bibliothécaire américain.

[2] (22 janvier 1922 - 3 septembre 1983) physicien anglais, spécialiste de l'histoire des sciences et de la science de l'information. Considéré comme le « père » de la scientométrie.

[3] Auteur américain de quelques romans policiers historiques

[4] Informaticien américain

2. Caractéristiques des Big Data

Parmi les définitions des Big Data on trouve celle par caractéristiques. Plusieurs analyseurs de data, depuis la publication de Doug Laney “3D Data Management: Controlling Data Volume, Velocity, and Variety” approuve que les Big Data se caractérise par trois caractéristiques [1]:

- Volume
- Variété
- Vélocité

- **Volume**

L’ordre de grandeur des données, de nos jours les tailles des Big Data sont rapportées dans plusieurs téraoctets et pétaoctets. L’accroissement du volume vient de l’augmentation du nombre d’individus observés, de la fréquence d’observation et d’enregistrement des données et du nombre d’éléments observés [6].

- **Variété :**

L'hétérogénéité structurelle dans un ensemble de données [1].

- **Les données structurées :** Réfère à des données tabulaires trouvées dans des tableurs tabulaires trouvé dans des tableurs ou des bases de données relationnelles, constituent 5% de toutes les données existantes.
- **Les données non structurées :** données qui manquent parfois la structuration requise par les machines pour faciliter l’analyse Les textes, images, audio, et vidéo sont des exemples de ce type de données.
- **Les données semi-structurées,** le format de données semi-structurées ne sont pas conformes à des normes strictes. Par exemple, les Extensible Markup Langage (XML), qui sont un exemple typique de données semi-structurées.

- **Vélocité :**

La vitesse de génération des données et la vitesse d’analyse et de prise de décision. La prolifération des appareils numériques tels que les téléphones intelligents et les capteurs a conduit à un taux de création de données sans précédent et a créé un besoin croissant d'analyse en temps réel [1].

En plus de ces 3 caractéristiques précédents, d’autres caractéristiques des Big Data ont apparu :

- **Véracité :**

Fais référence au manque de fiabilité inhérent à certaines sources de données. Par exemple, les sentiments des clients dans les médias sociaux sont de nature incertaine, car ils entraînent le jugement humain. Pourtant, ils contiennent des informations précieuses. Ainsi, la nécessité de traiter des données imprécises et incertaines est une autre facette des Big Data, qui est adressé à l'aide des outils et des analyses développées pour la gestion et l'exploitation des données incertaines [1].

- **Variabilité et complexité :**

La variation des débits de données. Souvent, la vélocité des Big Data est non conforme et a des pics et des creux périodiques. La complexité réfère au fait que les données importantes sont générées à travers une multitude de sources. Cela impose un défi crucial : la nécessité de se connecter, accoupler, nettoyer et transformer les données reçues des différentes sources [1].

- **Valeur :**

Les données reçues dans la forme originale a généralement une faible valeur par rapport à son volume. Toutefois, une valeur élevée peut être obtenue par l'analyse de grands volumes de ces données [1].

3. Traitement Cycle de vie des Big Data

Depuis leur acquisition, les données sont traitées de manière délicate pour qu'elles puissent être bénéfiques pour l'organisation accueillante et pour les utilisateurs finaux des services proposés. Les données passent par différentes étapes de traitement :

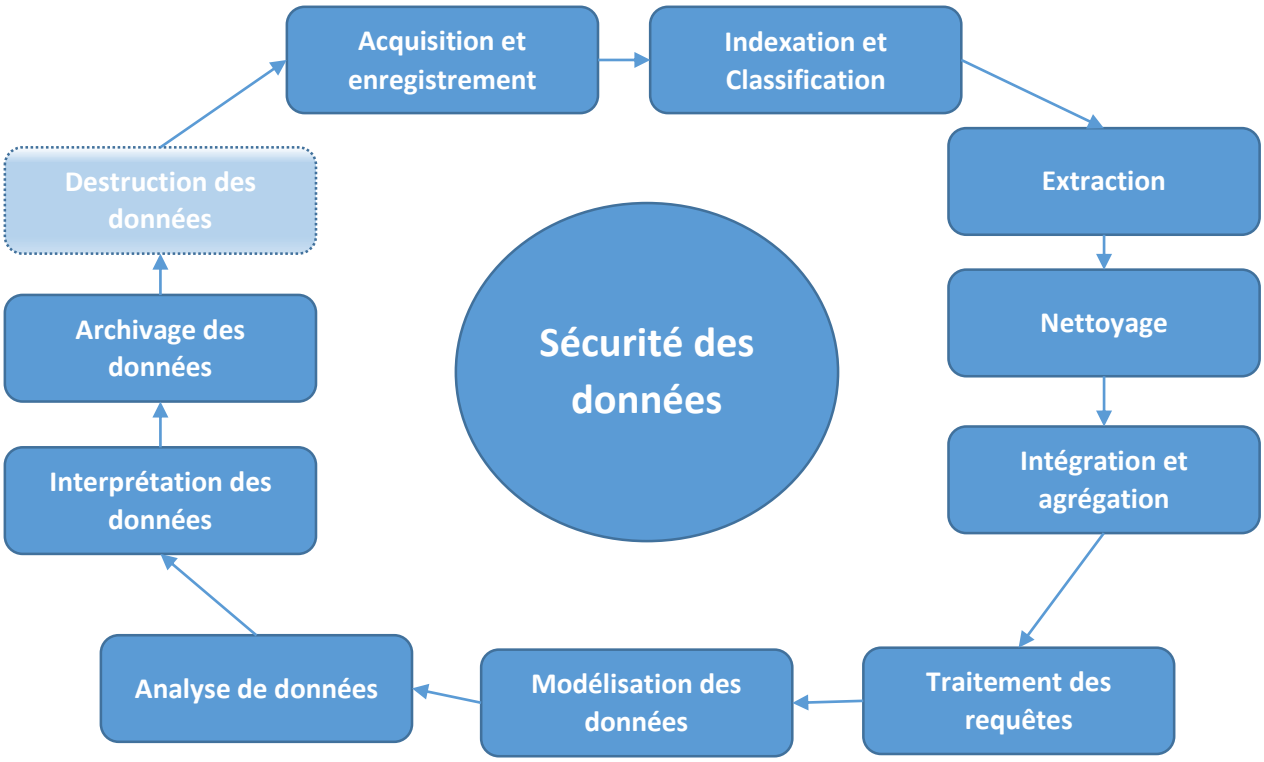


Figure 2. Cycle de vie des Big Data

3.1.1. Acquisition et enregistrement des données

Les données sont enregistrées à partir de plusieurs sources qui génèrent des données. Une grande partie de ces données est sans intérêt, et elle peut être filtrée et compressée par ordre de grandeur. Le but de cette étape est de définir les filtres appropriés en faisant sorte qu'ils n'ignorent pas les informations utiles [4].

Un grand défi de l'enregistrement est de générer automatiquement les métadonnées correctes pour décrire les données enregistrées, comment elles sont enregistrées et leurs mesures. Un autre défi est l'obtention de la source des données [4].

3.1.2. Indexation et Classification des données

L'indexation est une opération qui pointe sur l'emplacement des dossiers, fichiers et enregistrements. Dans cette étape, une identification de l'emplacement des ressources est réalisée afin de faciliter l'étape de classification.

La classification des données nous permet de catégoriser les données selon certaines caractéristiques et/ou formats pour faciliter la détermination des valeurs des données et l'accès à ces derniers.

Les risques peuvent inclure des pertes de données sur des indexes non existants ou une allocation de ressource pour des raisons de sécurité inapproprié à cause d'une mauvaise classification des données qui finit par dévaloriser son importance [5].

3.1.3. Extraction des données

Souvent, les informations recueillies ne sont pas dans un format près à l'analyse. Nous ne pouvons pas laisser les données dans les formes d'origine en les analysants. Mais, nous devons plutôt exiger un procédé d'extraction qui extrait les informations requises par les sources sous-adjacentes et les exprime sous une forme structurée adaptée pour l'analyse [4].

3.1.4. Nettoyage des données

Plusieurs données peuvent nous tromper. Par exemple, plusieurs clients peuvent choisir de masquer les comportements à risques et les personnes qui accueillent ces données peuvent donner de mauvaises analyses. Les travaux existants sur le nettoyage des données supposent des contraintes bien connues sur des données valides ou des modèles d'erreur bien compris [4].

3.1.5. Intégration et agrégation des données

Compte tenu de l'hétérogénéité des données, il ne suffit pas de les enregistrer. Si on stocke seulement l'ensemble de données dans le référentiel, il est peu probable que quiconque ne sera jamais en mesure de trouver ou d'utiliser ces données. Avec des métadonnées adéquates la probabilité augmente mais malgré cela, ce processus reste difficile dû à des différences dans les détails et dans la structure d'enregistrement de données [4].

Pour une analyse efficace à grande échelle, tout cela doit se faire de façon complètement automatisée. Cela nécessite que les différences dans la structure de données et la sémantique s'expriment dans des formes qui sont compréhensibles aux machines. Le processus d'intégration des données nécessite un environnement de travail très fort. [4]

3.1.6. Traitement des requêtes

Les méthodes d'interrogation et d'exploitation des Big Data sont fondamentalement différentes des analyses traditionnelles sur des petits échantillons de données. Les Big Data sont souvent bruyants, dynamiques, hétérogènes, interdépendants et indignes de confiance. Néanmoins, même les Big Data bruyants pourrait être précieux plus que des petits échantillons de données.

Les Big Data interconnectés forment de grands réseaux d'informations hétérogènes, où la redondance des informations peut être explorée afin de compenser pour les données manquantes, pour faire face aux cas contradictoire, pour valider les relations de confiance pour découvrir les relations et les modèles cachés… [4]

3.1.7. Modélisation des données

L'exploitation des données requiert des données intégrées, nettoyées, fiables, efficacement accessibles… et des environnements adaptés pour les Big Data. En même temps, l'exploitation des données elle-même peut également être utilisée pour aider à améliorer la qualité et la fiabilité des données, à comprendre sa sémantique, et à fournir des fonction d'interrogation intelligentes [4].

Les outils de modélisation traditionnels ne sont plus efficaces dans l'ère des Big Data, ils ne sont pas pratiques ni utiles lorsque les données sont d'un nombre immense et d'une mauvaise structuration. Vu l'importance de cette étape dans le processus, de nouvelles technologies ont été présenté pour combler les nouvelles problématiques [4].

3.1.8. Analyses des données

Le problème avec l'analyse actuelle des Big Data est le manque de coordination entre les systèmes des bases des données, qui hébergent les données et fournit une interrogation SQL, avec des paquets d'analyses qui exécutent diverses formes de traitement non-SQL. Les analyses, aujourd'hui, sont entravées par un processus fastidieux d'exportation des données à partir des bases de données, en effectuant un processus non-SQL et en récupérant les données [4].

Les Big Data permettent également à la prochaine génération d'analyser des données interactives avec des réponses en temps réel. Dans l'avenir, les requêtes vers les Big Data seront automatiquement générés pour la création de nouveau contenu sur les sites web, pour remplir des listes et des recommandations des éléments les plus populaire, et pour fournir une analyse ad hoc de la valeur des données qui va permettre de décider de stocker ou de jeter ces données [4].

3.1.9. Interprétation des données

Avoir la capacité d'analyser les Big Data n'est pas d'une grande utilité si les utilisateurs ne peuvent pas comprendre cette analyse. En fin de compte, un décideur qui est fourni avec le résultat d'analyse doit interpréter ces résultats. Habituellement, cela consiste à examiner toutes les hypothèses retenues et retracer l'analyse.

En bref, il est rarement suffisant de fournir seulement les résultats, il faut plutôt fournir des informations supplémentaires qui expliquent comment chaque résultat a été dérivé et fondé précisément sur ce qui est en entré.

En étudiant la meilleur façon de capturer, de stocker et de capturer les métadonnées, nous pouvons créer une infrastructure pour fournir aux utilisateurs la capacité à la fois d'interpréter les résultats analytiques obtenus et de répéter l'analyse avec de différents hypothèses, paramètres ou ensemble de données [4].

3.1.10. Archivage des données

Les données ont une certaine durée de vie. À un certain point, elles deviennent inutilisables, obsolètes, et/ou remplacées par de nouvelles données plus pertinentes. Considérons les coûts de

stockage des données et la responsabilité potentielle pour la négligence des données obsolètes, un protocole devrait donc être élaboré pour déterminer quelles données doivent être archivé, quand, où, et pour combien de temps [5].

3.1.11. Destruction des données

Détruire les données veut dire rendre ces données illisibles. Mais la question qui se pose est dans l'environnement digital en ligne d'aujourd'hui, les données peuvent-elles vraiment être détruites et ne jamais être revues ? Les systèmes caches, les backups multiples, et d'autres pratiques informatiques nous répondent qu'il est impossible de détruire les données définitivement. Toutefois, cela n'empêche pas les entreprises de tenter de vider leurs entrepôts de données et de mettre à la retraite leurs vieux dispositifs (ou en abandonnant des dispositifs sans supprimer les données qu'il retenait)… [5]

3.1.12. Sécurisation des données

a. *Perte de données*

Les données peuvent être perdues dans plusieurs façons. Selon les 'Online Backup Geeks' les raisons les plus courantes (figure 2).

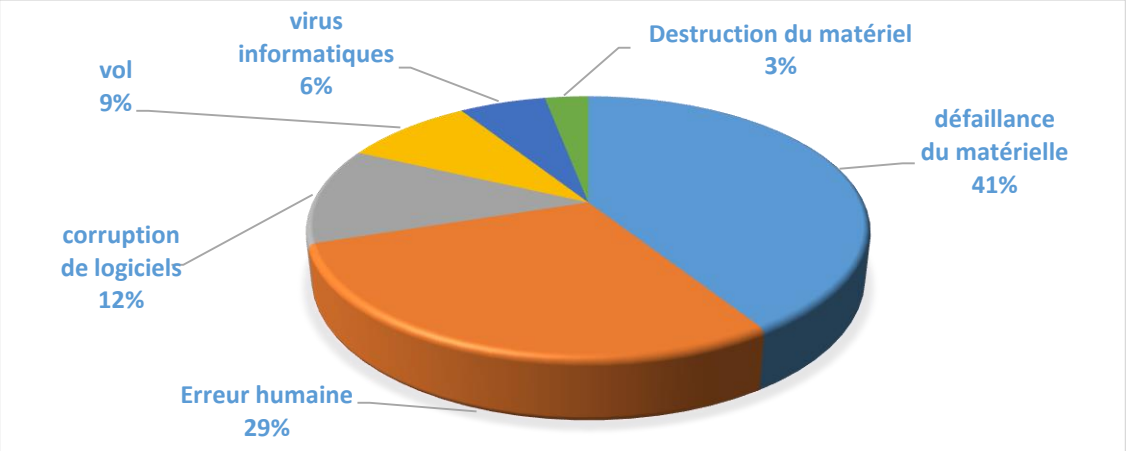


Figure 3. Les raisons les plus courantes de perte de données selon les 'Online Backup Geeks' [5]

Pour faire face à la destruction et à la défaillance du matérielle, il faut faire en sorte que les protecteurs d'électricité sont en place et les protocoles de sauvegarde sont mise en œuvre peuvent soulager le risque si l'inévitable arrive. Une autre solution consiste à acheter des générateurs de secours ou d'autres sources d'énergie alternatives qui peuvent faire face au dysfonctionnement principal. Veiller à ce que les dispositifs sont physiquement sécurisés est une autre pratique nécessaire.

Une fois que le dispositif est sécurisé, la question qui se pose est, est-ce que les données qui résident dans ces dispositifs sont sécurisées ? Les autorisations d'accès aux données dans les dispositifs via des mots de passe ou d'autres méthodologies, ainsi que s'assurer que les données sont cryptées, ne sont pas les seules bonnes règles mais il faut aussi de certains règlements et lois [5].

b. *Accès aux données*

L'accès aux données détermine qui a le pouvoir d'utiliser certaines données pour des raisons précises. L'accès est déterminé par la position de l'individu dans une organisation. Les niveaux d'accès de données peuvent changer au cours de la durée de travail d'un employé avec l'entité en raison de changements de sa position [5].

L'accès aux données doit être géré et contrôlé afin que les modifications de l'autorité de l'individu soient notées dès que possible pour éviter les risques potentiels de l'accès non autorisé ou de comportements malveillants contre les données de l'organisation [5].

Pour faire face à cela, une liste des meilleurs pratiques et des mesures à prendre pour réduire la probabilité de ces risques a été établie [5] :

1. Les données de vérification d'accès, la première étape consiste à examiner les protocoles et les politiques d'accès aux données.
2. Inventaire de permission, cet étape nécessite d'identifier et de lister qui a accès à quelles données et pourquoi.
3. inventaire de hiérarchisation de données, après remplissage de l'audit et de l'inventaire des autorisations, il faut également identifier les types de données qui sont dans les systèmes de l'entreprise. Cela permet de spécifier si les données sont inutilisées et ne sont plus utiles ou si les données sont encore utilisées.
4. Aligner les groupes de sécurité aux données. D'après le privilège des données, il faut déterminer le niveau de contrôle de sécurité dont elles ont besoin et quelles sont les stratégies à mettre en place.
5. Identifier les propriétaires des données, la plupart des individus sont des utilisateurs qui utilisent des données qui sont fournies par quelqu'un d'autre. Il faut savoir qui fournit les données ainsi que qui sont responsables de ces données.

c. *Protection des données : Backup*

Les contrôles d'accès aident à la sécurisation des données en se concentrant sur qui est autorisé à accéder aux données et à quelles données sont-ils autorisés à accéder. Une autre clé de la protection des données est le plan de sauvegarde 'backup plan', une procédure documentée pour créer une version (ou des versions) dupliquées des données importantes de la société qui sont sécurisées et qui peuvent être facilement accessibles en cas de besoin [5].

Le backup est une planification de l'échec qui minimise le risque de perte de données qui ne peuvent pas être récupérées. Le backup n'est pas synonyme de la reprise après perte, mais elle est une partie essentielle de celle-ci. Il est basé sur l'idée que la duplication des données et offre la sauvegarde de la redondance en cas de problème [5].

d. *Récupération des données*

Il y a eu de nombreuses histoires de sociétés qui ont fait faillite après une violation de données importante. La plupart de ces sociétés échouent parce que les données perdues ne peuvent pas être récupérés en temps opportun, voire jamais, conduisant à une cessation des activités d'affaires ou d'une amende substantielle d'un organisme de réglementation [5].

Un élément clé de tout plan de backup est la fonction de restauration. Après qu'un incident est découvert, un rapport d'incident est généralement compilé et connecté indiquant certains faits de

l'incident pour un examen ultérieur et pour qu'il devienne une partie de la base de connaissances institutionnelles en termes de sécurité et de gestion des risques [5].

4. Domaines d'application

Les Big Data ne se caractérisent pas seulement par leur vitesse de progression, mais aussi par leur capacité d'adaptation dans tous les domaines qui nous entourent.

- **Informatique :** Surveillance des machines, de grands réseaux, et détection de dysfonctionnements ou d'incidents sécuritaires [6].
- **Transports :** Fixation dynamique du prix des billets d'avion, amélioration du trafic routier par géolocalisation, recherche de la station-service la plus proche, des places libres de stationnement… [6]
- **Marketing :** Usages familiers des Big Data comprennent "les systèmes de recommandation", proposition des suggestions d'achat en fonction des intérêts antérieurs d'un client par rapport à des millions d'autres [7].
- **Grande distribution :** Analyse des tickets de caisses et croisement avec les données du programme de fidélité [6].
- **Ressources humaines :** Analyse des CV enrichie par la détection des liens noués par le candidat sur les réseaux sociaux [6].
- **Enseignement :** Proposition de différents cours, exercices et examens aux étudiants et professeurs pour aider à améliorer l'enseignement [6].
- **Domaine public :** Affectation des ressources policières en prédisant où et quand des crimes sont les plus susceptibles de se produire, trouver des associations entre la qualité de l'air et la santé… [7].
- **Les réseaux sociaux :** Actuellement, un nombre gigantesque de données est collecté à l'aide des réseaux sociaux, qui en tire profit et proposent plusieurs services en revanche. Par exemple : [8]
  - **LinkedIn**[5] collecte les données de ses utilisateurs, et propose plusieurs services tels que « People you may know ».
  - **Netflix**[6] utilise les données collectées pour créer des recommandations à ses utilisateurs, collecte les évènements qui sont analysés en ligne et hors ligne, pour créer des recommandations de vidéos.
  - **Twitter**[7] utilise les données collectées pour créer des requêtes de suggestion en temps réel et offre de nouveaux services comme « Who to follow ».
- **Domaine de santé :** Le domaine de santé est aussi un domaine qui génère un ensemble de données important à traité, il y a eu plusieurs études des Big Data en science de la santé. Par exemple : [9]
  - Systèmes de recommandations en soins de santé.
  - Surveillance des épidémies sur internet.
  - Capteur de surveillance de l'état de santé et du suivi de la sécurité alimentaire.
  - Déduction de la qualité de l'air en utilisant les Big Data.

[5] Réseau social professionnel en ligne créé en 2003 à Mountain View (Californie)
[6] Entreprise américaine proposant des films et séries télévisées en flux continu sur Internet
[7] Outil qui permet à un utilisateur d'envoyer gratuitement de brefs messages

Conclusion

Grâce aux multiples avantages qu'ils apportent, les Big Data ont rencontré un grand succès et une ouverture du publique sur l'idée, mais malgré cela ce phénomène est aussi livré avec plusieurs inconvénients qui sont très difficiles à surmonter [10]:

- **On ne peut pas tous quantifier,** la transformation des activités réel en data ne peut pas toujours être parfaite, la quantification d'un élément peut porter des erreurs qui vont se multiplier lors de la combinaison avec d'autres quantifications de d'autres éléments qui peuvent à lors tour porter des erreurs.
- **Qu'est-ce que et qui n'est pas couvert par la datafication ?** Contre toute attente, il y a toujours des personnes qui sont restées en dehors du nuage Big Data, ces derniers sont en général négligés lors des études effectués en se basant sur les données générées en web.
- **La complexité des algorithmes des Big Data,** il requière une connaissance profonde des mathématiques appliquées, et des écosystèmes informatiques.

Comme tout nouveau phénomène dans le monde de l'informatique, les Big Data est une épée à deux tranchants, qui est livrée avec des avantages et des inconvénients, si on peut surmonter les inconvénients, on pourra être capable de jouir des multiples avantages des Big Data. Dans le chapitre suivant, nous allons présenter de différentes œuvres discutant de différents aspects des Big Data.

Chapitre II

Productions scientifiques Etat de l’art

Introduction

Les Big Data est un domaine naissant en théorie, mais durant ces dernières années plusieurs laboratoires se sont intéressés à ce domaine, qui est encore brute et qui mérite l'attention grâce à son impact sur tous les domaines qui nous encerclent.

La naissance du phénomène a surpris les chercheurs, puisqu'il a commencé à se développer dans les entreprises qui ont essayé d'adapter leurs architectures à l'inflation des quantités de données qu'ils doivent traiter, mais ils n'ont pas pu combler les besoins qui augmentent de manière exponentiel et avec une très grande vitesse. Pour remédier à ce problème, plusieurs chercheurs ont essayé de cerner les différents aspects des Big data et de proposer des solutions probables.

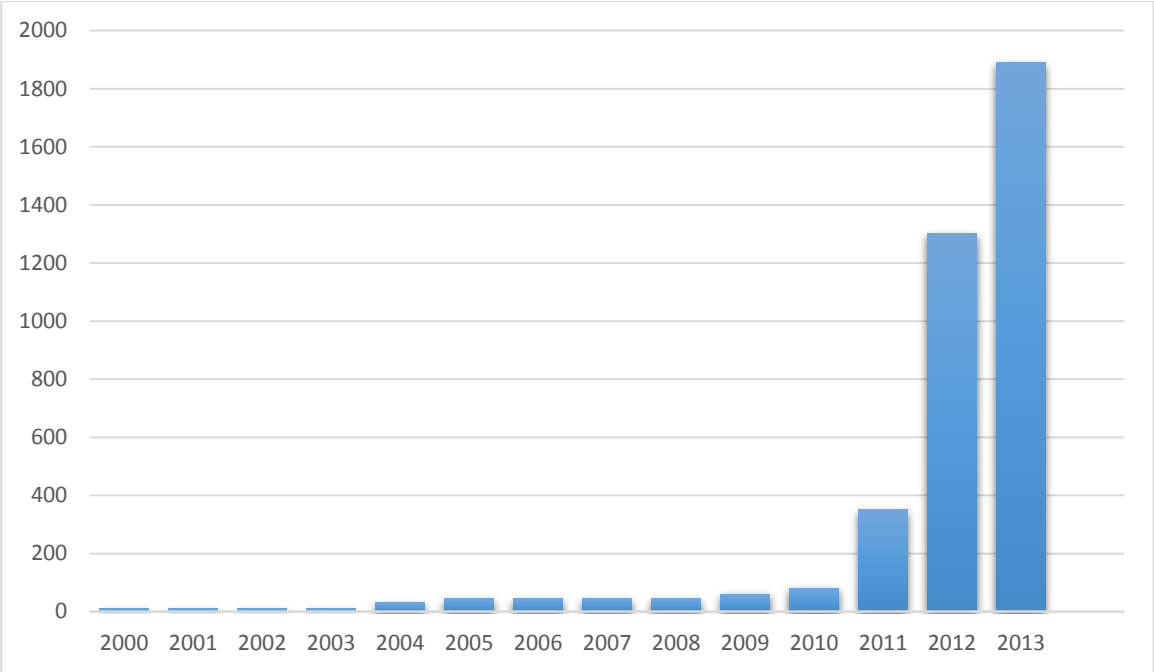


Figure 4. Fréquence de distribution des documents contenant le terme 'Big Data' dans la librairie de recherche « ProQuest » [1]

1. Big Data, contexte général

Plusieurs publications se sont limitées à définir et à cerner le concept Big Data avec tous les éléments qu'il engendre, ils ont essayé de définir les Big Data avec toutes ces problématiques.

XiaolongJin et all [36] ont décrit la signification des Big Data et les initiatives de plusieurs pays et organisations en ce domaine, en effet, plusieurs pays comme l'Amérique, l'Angleterre, la France le Japon et plusieurs d'autres pays ont lancé des initiatives portant sur les Big Data.

Ils discutent aussi l'impact des Big Data en plusieurs domaines :

- Impact sur le développement national ;
- Impact sur le développement industriel ;
- Impact sur les recherches scientifiques ;
- Impact sur la recherche interdisciplinaire émergente ;
- Comment les Big Data aident le monde à mieux percevoir le présent ;
- Comment les Big Data aident le monde à mieux prédire l'avenir.

Cet article nous permet d'avoir une vision en chiffres réels sur l'importance des Big Data, leur influence sur le monde entier, et leur efficacité malgré les différents défis qui leurs font face, comme la complexité des données, la complexité du calcul, le grand volume, la multitude des sources de données la complexité des systèmes de traitements…

Dans le même axe, Nir Kshetri [11] s'est concentré sur les perceptions des consommateurs et des entreprises et de leurs réponses au phénomène des Big data, ils ont présenté plusieurs sondages réalisés durant ces dernières années, qui ont indiqué qu'une large proportion d'organisations n'est pas préparée pour faire face aux problèmes de sécurité. Les consommateurs ont aussi exprimé leur préoccupation pour le manque d'honnêteté dans les organisations qui utilise mal leurs informations personnelles.

De plus, l'auteur a noté les bénéfices, les coûts et les externalités des Big Data, pour commencer ils ont cité les bénéfices sociaux et économiques, et leur cout dans les mêmes secteurs. Il a aussi présenté les caractéristiques des Big Data d'une manière différente, en les reliant avec l'intimité, la sécurité et le bien-être des consommateurs.

2. Architectures proposées

Malgré l'avancement du côté pratique sur le côté théorique en Big Data, plusieurs auteurs ont présenté leurs propres architectures des Big Data selon un domaine précis sur lequel ils ont choisi de travailler, en essayant de présenter une architecture qui pourra combler le besoin des organisations.

Samson Oluwaseun Fadiyaa et all [12] ont choisi de se focaliser sur l'étape de stockage des données, en présentant tout d'abord, le besoin et les conditions de stockage. Ils ont présenté deux architectures de stockage existantes, la première est une architecture de stockage traditionnelle qui n'est dédiée qu'aux données structurées (Annexe 1).

Ils ont présenté trois nouveaux apports majeurs:

1- Analyse directe sur des traitements parallèle massive des entrepôts de données
2- Analyse indirecte sur Hadoop
3- Analyse directe sur Hadoop

Par contre C. Dobre, F. Xhafa [13] se sont focalisés sur l'impact des Big Data sur la construction d'une ville intelligente, pour commencer ils ont défini les problématiques, les défis et les challenges dans une ville intelligente, qui dépendent totalement sur le contexte d'exécution des applications. Le contexte d'exécution des applications est tout information qui peut être obtenue et procédée par un système pour identifier la situation d'une entité (personne, emplacement, ou un objet), et adapte le comportement d'un système à une situation précise.

Ils ont présenté une plateforme existante qui supporte les contextes intelligents qui est une plate-forme de surveillance qui intègre les services destinés à recueillir des données de contexte (lieu, profil et caractéristiques des utilisateurs…). Ces services intelligents utilisent les capacités de détection de smartphones modernes et les tablettes, éventuellement augmentés par des capteurs externes. L'architecture proposée se compose de plusieurs couches (annexe 2), Chacun fournissant une fonction spécifique:

(1) Collecte d'informations de contexte,
(2) Stockage et agrégation des informations de contexte,
(3) Construction de règles d'exécution sensibles au contexte,
(4) Visualisation et interaction de l'utilisateur.

Contrairement aux deux architectures précédentes Pekka Pääkkönen, Daniel Pakkala [36] ont essayé de construire une architecture de référence, qui englobe les étapes de traitement des Big Data avec le flux de données passant entre ces étapes. Ils ont divisé le processus de traitement en les étapes suivants :

- Extraction de données
- Chargement et prétraitement des données
- Traitement des données
- Analyse des données
- Chargement et transformation des données
- Visualisation des données

Ils ont aussi divisé les données selon deux caractéristiques, la mobilité et la structure, et ils ont présenté différents type de stockage et d'entrepôts utilisés dans chaque étape du processus de traitement, une spécification de jobs et de modèles a aussi été présentée (Annexe 3).

L'article présente aussi les infrastructures de différents réseaux sociaux, Facebook, LinkdIn, Tweeter, NetFlix… avec une adaptation de ces architectures à l'architecture proposée par les auteurs, et les différents outils qu'utilisent chacun de ces réseaux.

Pour finir les auteurs ont aussi présenté une classification des outils existant dans le marché pour le processus de traitement des Big Data dans différentes organisations.

3. Stockage des données

L'un des caractéristiques primaires des Big Data est le Volume, ces Big Data doivent être stocké avant tout pour qu'on puisse en tiré de la valeur. Plusieurs articles se sont intéressés au stockage des données, ces derniers ont discuté l'importance de cet étape et ont essayé de l'expliquer, parmi ces articles, il y en a ceux qui ont proposé de nouvelles architectures et plateformes pour un meilleur stockage des Big Data.

Paolo Atzeni et all [21] ont présenté une nouvelle démarche qui offre une interface uniforme qui permet l'accès aux données stockées dans différents systèmes NoSQL, sans savoir à l'avance quel type de système, et aussi donner possibilité d'utilisation de différents systèmes au sein d'une seule application, ce système.

Ils ont commencé par décrire les systèmes NoSQL, leurs modèles de données et les ont classés sous plusieurs catégories en se basant sur plusieurs critères. Ils ont proposé trois catégories: Les bases de données orienté colonnes, les bases de données orienté documents, les bases de données clé-valeur, qui sont caractérisé selon leur différence en manière de réalisation des opérations, des détails spécifiques de la structure et dans les aspects architecturaux.

- **Les bases de données orienté colonnes**, qui ressemble à première vue à une table dans un SGBDR à la différence qu'avec une BD NoSQL orientée colonne, le nombre de colonnes est dynamique.
- **Les bases de données orienté documents**, qui se base sur le paradigme clé valeur. La valeur, dans ce cas, est un document de type JSON ou XML. L'avantage est de pouvoir récupérer, via une seule clé, un ensemble d'informations structurées de manière hiérarchique. La même opération dans le monde relationnel impliquerait plusieurs jointures.

- **Les bases de données clé-valeur**, où les données sont, donc, simplement représentées par un couple clé/valeur. La valeur peut être une simple chaîne de caractères, un objet sérialisé. L'absence de structure ou de typage ont un impact important sur le requêtage.

Thirumalaisamy Ragunathan and all [23] ont pointé sur l'importance des DFS, systèmes de gestion de fichier moderne, en les comparants avec plusieurs d'autres outils de stockage. Ils ont donné exemple avec le système de gestion de fichier Hadoop HDFS en expliquant les composants de ces derniers le Namenode, le Secondary Namenode et le Datanode.

Ils ont proposé de nouveaux algorithmes de lecture pour améliorer la performance des opérations de lecture dans les différents DFS.

Ils ont nommé leurs algorithmes « Proposed Caching and Speculation-Based Algorithms », dans le traitement spéculative 'Speculation processin', une tâche pourra être effectuée avant que l'on sache si cette tâche est nécessaire ou non, Ensuite, en fonction des conditions sur les sorties de la tâche accomplis ils seront acceptés. Ce type de traitement permettra selon eux de réduire le temps d'attente et peut améliorer les performances.

Dans leurs algorithmes, ils considèrent un DFS qui consiste en un NameNode et de multiples DataNode et tous ces nœuds sont reliés par un réseau. Les entrepôts des NameNode, des métadonnées et des données stockent les données et exécute des programmes d'application de l'utilisateur.

4. Analyse des données

L'analyse des données en Big Data est une étape très importante et très complexe, plusieurs documents ont choisi de traiter l'analyse des données en proposant quelques techniques qui aide à analyser ces grands échantillons de données.

Jianshan Sun, WeiXu, JianMa, JiasenSun [15] Compte tenu des facteurs de pertinence des sujets traités en Big Data, ils ont proposé une nouvelle approche de modélisation pour recommander des experts dans les communautés scientifiques. La méthode de recommandation proposée est bien évaluée et comparée avec certains modèles de recommandation couramment utilisés.

Le système expert proposé emploie l'analyse de pertinence, de l'analyse de qualité, et de modules d'analyse de connectivité pour construire un modèle chercheur plus complet pour trouver un expert. Le processus et les principaux éléments du système proposé sont présentés dans la figure 11.

Le profilage 'profiling' : C'est le processus d'identification et de détermination des informations et des attributs qui peuvent être utilisés pour caractériser un objet donné.

La modélisation 'modeling' : ils ont présenté trois types de modélisation :

- Module d'analyse de pertinence, ce module propose une méthode de pondération sémantique mot-clé pour déterminer la pertinence du contenu des experts candidats.
- Module d'analyse de qualité, ce module propose une analyse de requête sensible à la qualité pour évaluer le niveau d'expertise des candidats experts.
- Module d'analyse de connectivité, Ce module propose une analyse de connectivité pour classer les candidats experts par l'hypothèse que les chercheurs accumulent les expertises à partir de la collaboration avec d'autres experts.

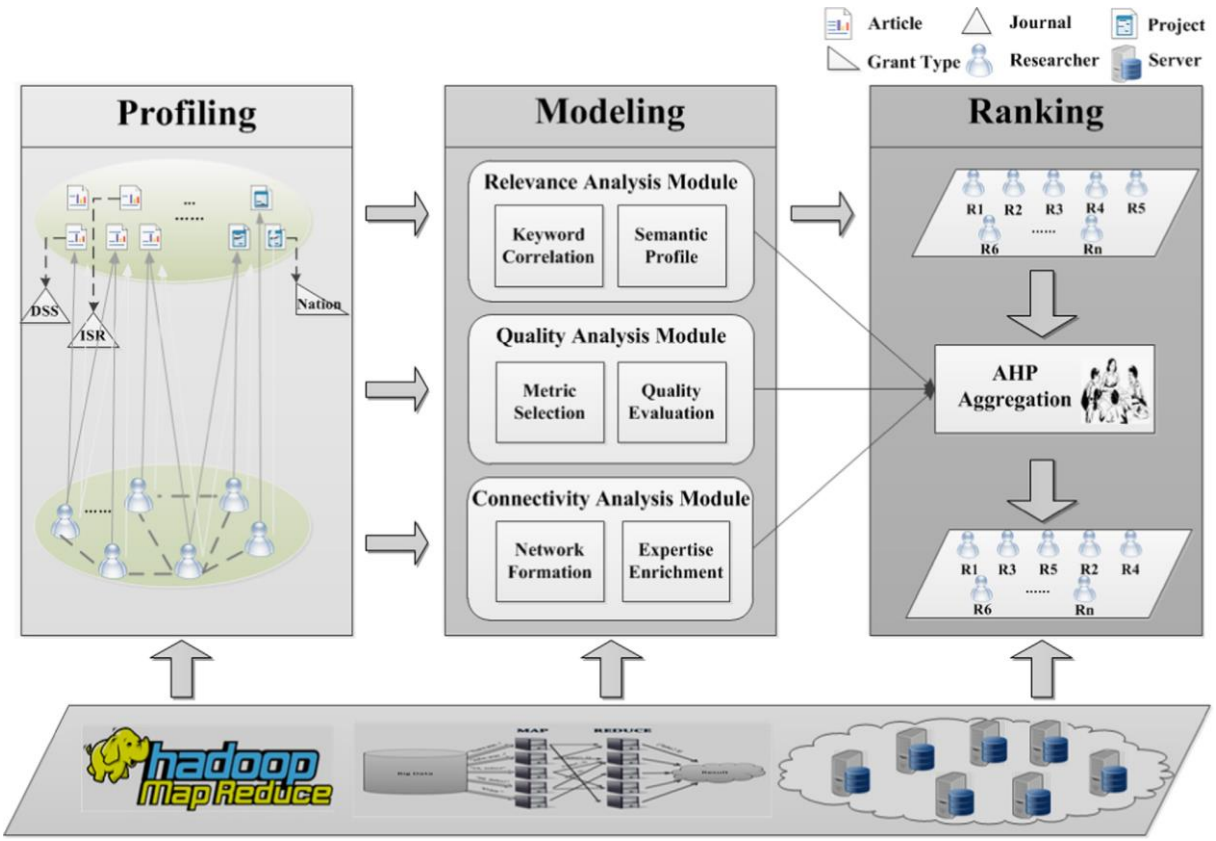


Figure 5. Architecture du système viseur d'experts [15]

Dans ce travail, ils ont aussi mis en œuvre le Framework MapReduce Hadoop sur une infrastructure de cloud computing, pour effectuer tâches de calcul à large échelle impliqués dans le système viseur d'expert.

5. Sécurité

Guillermo Lafuente [16] a discuté plusieurs grands défis que rencontrent les Big Data en termes de sécurité, il a proposé plusieurs mesures qui peuvent être prises pour aider à faire en sorte que les informations soient traitées avec la diligence requise.

- **Anonymisation des données :** c'est un processus important, il faut veiller à ce que toutes les informations sensibles soient anonymes. Les données anonymisées pourraient être recoupées avec d'autres données, par conséquent, le chiffrement de ces données est également primordial.
- **Cryptage des données :** Lors du stockage des données, les organisations devront le chiffrer. Le cryptage est bien sûr la principale solution pour garantir que les données restent protégées.
- **Contrôle surveillance d'accès :** Les mécanismes de contrôle d'accès adéquats seront la clé dans la protection des données. Une meilleure approche est de protéger les informations en utilisant le cryptage des données qui peuvent seulement être décryptées si l'entité essayant d'accéder à l'information est autorisée par une politique de contrôle d'accès.
- **Approches politique et de conformité :** Le principal défi présenté par Big Data est de savoir comment identifier les pièces sensibles de l'information qui sont stockés dans le jeu de données non structurées.

Les organisations doivent s'assurer qu'ils isolent des informations sensibles. Elles doivent exécuter une évaluation des risques sur les données qu'ils recueillent. Elles devraient se demander si elles recueillent des informations qui devraient être maintenu privé et établir des politiques adéquates qui protègent les données et le droit à la vie privée de leurs clients.

- **Gouvernance et cadres juridiques :** Le principal problème du point de vue de la gouvernance est que les Big Data est un concept relativement nouveau et les procédures politiques donc n'ont pas été totalement créés.

L'auteur a aussi proposé plusieurs conseils pratiques pour faire face aux problèmes de sécurité :

- Créer des politiques qui permettent l'accès aux utilisateurs qui sont autorisées.
- Protéger les communications : les données en transit doit être suffisamment protégé pour assurer la confidentialité et l'intégrité.

Dans le même axe, B.Saraladevi and all [17] ont révélé les challenges de la collecte et l'analyse d'un grand ensemble de données qui contient de nombreux renseignements et des informations brutes sur une base de données. Ils ont étudié la sécurité dans la plateforme Hadoop qui est utilisé pour stocker, gérer et distribuer des données à travers plusieurs grands nœuds de serveur.

Ce document présente les grands enjeux de données et plusieurs d'autres problématiques sur la question de sécurité qui se posent dans l'architecture basée sur les couches nommé systèmes de fichiers distribués de Hadoop (HDFS). La sécurité HDFS est améliorée en utilisant trois approches comme Kerberos, l'algorithme et le NameNode.

Ils pointent aussi sur la défaillance au niveau de sécurité de Hadoop en affirmant que le Secteur Gouvernemental et les organisations n'utilisent pas Hadoop pour stocker des données précieuses en raison de préoccupations de sécurité. Ils assurent la sécurité dans l'environnement extérieur de Hadoop comme les pare-feu et les systèmes de détection d'intrusion.

Certains auteurs assure que HDFS Hadoop est sécurisé et peut éviter le vol, les vulnérabilités seulement en cryptant les blocs et les systèmes de fichiers individuels dans Hadoop. Même si d'autres auteurs ont cryptées les blocs et les nœuds en utilisant des techniques de cryptage, mais aucun algorithme parfait est mentionné pour maintenir la sécurité dans l'environnement Hadoop. Afin d'accroître la sécurité dans cet environnement, des approches sont mentionnés ci-dessous.

- **Mécanisme Kerberos** : Le mécanisme Kerberos est utilisé pour améliorer la sécurité dans HDFS. Dans HDFS la connexion entre le poste client et le Namenode est obtenue en utilisant RPC. Ici, le jeton ou Kerberos est utilisé pour authentifier une connexion RPC. Si le client a besoin d'obtenir un jeton, le client fait usage de la connexion authentifiée Kerberos.
- **Approche Algorithme Bull Eye** : Cet algorithme balaye les données avant qu'ils soient autorisés à entrer dans les blocs et aussi après l'entrée. Ainsi cet algorithme se concentre uniquement sur les données sensibles qui comptent sur les informations stockées dans les nœuds de données.
- **Approche Namenode** : Dans HDFS il y a un problème si NameNode devient indisponible, afin d'augmenter la sécurité dans la disponibilité des données, il est préférable d'utiliser deux NameNode. Le premier NameNode est considéré comme maître et l'autre comme esclave.

6. Big Data et domaine médicale

Parmi les domaines influencés par le changement immense dans la manière de traitement des Big Data, nous nous somme intéressé par le domaine médical, vu que c'est un domaine critique dont les données sont de très grande importance et qui ont besoin d'informatisation vu qu'ils sont difficiles à traiter par l'être humain seul. Pour cela, nous présentant un ensemble de publications qui présentent de nouvelles méthodes.

Ken Ka-Yin Lee, and all [18] se sont focalisés sur l'étape de stockage des données médicales, ils ont présenté trois approches en appuyant sur le volume et la complexité de ce type de données qui sont considérées comme les données les plus complexes mais aussi les intéressantes.

Pour traiter correctement les données et les rendre accessibles, il est nécessaire de formater l'information avec une structure exploitable par les ordinateurs et les convertir en un format structuré. Les données cliniques brutes peuvent être structurées par une structure de données d'arbre.

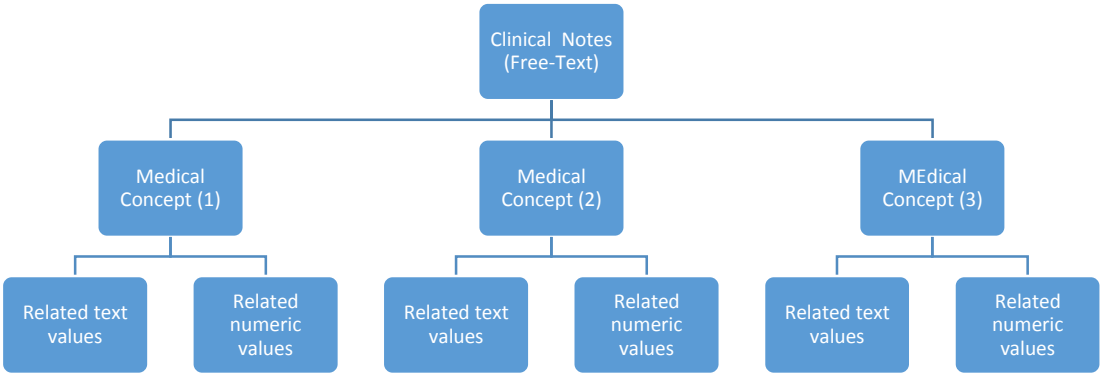


Figure 6. Structure de données générale pour les notes cliniques [18]

Le but principal de cette disposition est d'enfermer hiérarchiquement une liste de concepts au plus haut niveau, permettant aux sous-concepts à joindre le concept principal d'une manière flexible. Le champ de notes cliniques de texte libre est au plus haut niveau de chaque entrée. L'entrée des notes cliniques contient des concepts médicaux, avec des propriétés numériques ou textuelles. Cette disposition offre un maximum de flexibilité d'inclure autant de concepts médicaux nécessaires pour un enregistrement. Chaque concept peut avoir un certain nombre de propriétés, en forme textuelle ou numérique.

Ils ont aussi présenté une description sémantique de la définition de données dans la table ci-dessous, qui est utilisé pour les trois approches qui présentées et testées dans cet article sont :

- Modélisation de base de données NoSQL : Des bases de données NoSQL à faible coût, sans schéma, et horizontalement évolutive pour accompagner de nouvelles ressources informatiques lorsque lors du besoin (Annexe 4, partie A).
- Modélisation base de données XML : qui supporte nativement le stockage de données de façon hiérarchique (Annexe 4, partie B).

Les performances de base de données NoSQL, de base de données compatible XML et de base de données XML native sont comparées dans cet article en utilisant des données réelles fournies par

un médecin généraliste. Ils ont testé ces méthodes de plusieurs perspectives, temps de requête, efficacité de la préparation des données, évolutivité, extensibilité et flexibilité.

Ils ont conclu que l'approche NoSQL est une alternative viable à la conception de base de données relationnelle car il offre de meilleures performances de requête, tout en conservant certain degré d'évolutivité et de flexibilité. Alors que l'utilisation de XML pour le développement de base de données cliniques est une solution prometteuse, mais cette méthode n'est pas encore prête en particulier dans un environnement de grande production ou un environnement sensible au temps. Les NoSQL se trouve être une approche flexible pour la gestion des données cliniques, mais il ne répond pas en termes d'évolutivité et d'extensibilité par rapport aux approches de XML.

Dans la même ligne TaoHuang, et all [19] se sont aussi intéressés au domaine médical, ils ont commencé par cité les différentes étapes qui doivent être suivies dans un problème Big Data, pour eux chaque data scientist doit suivre les étapes suivantes :

- Choisir un problème
- Générer les données à l'aide des capteurs, moniteurs, profilage moléculaire ou extraction des données à partir des bases de données publiques après la mise en place d'un objectif pratique.
- Effectuer un prétraitement des données pour obtenir des informations significatives.
- L'idée de l'étude sera découverte à partir des données traitées par une analyse statistique.
- Les résultats d'analyse seront présentés à l'utilisateur final sous forme de rapport, recommandations en ligne ou prise de décision.

Les auteurs de cet article ont présenté de différents types d'études Big Data réalisées en science de la santé (Annexe 5):

- **Systèmes de recommandations en soins de santé**
- **Surveillance des épidémies sur internet**
- **Capteur de surveillance de l'état de santé et du suivi de la sécurité alimentaire**

Cet article présente aussi plusieurs sources de données, plusieurs techniques d'analyse de données, et de différents outils de visualisation de données avec leurs caractéristiques, points forts et points faibles, et pour finir, ils ont défini quelques perspectives des Big Data en domaine de la santé.

Fabricio F. Costa [24] se sont focaliser sur la rapidité de développement des ordinateurs et l'internet alors qu'il y a un manque d'infrastructures de calcul qui est nécessaire pour générer, maintenir, transférer et analyser les informations de biomédecine à grande échelle en toute sécurité en intégrant des données omiques avec d'autres ensembles de données, tels que les données cliniques de patients. Ils ont présenté leur propre vision du pipeline par lequel passe les données biomédical en cour de traitement (annexe 6).

Conclusion

Pour pouvoir trouver de bons résultats, les auteurs de ces différentes publications ont utilisé de nouvelles technologies qui différent des anciens outils utilisés sur les données traditionnelles. Dans le chapitre à suivre, nous allons présenter les principaux changements de technologies entre les données traditionnelles et les Big Data, et aussi les principaux nouveaux outils qui ont été développés pour faire face aux nouvelles exigences des Big Data.

Chapitre III
Technologies et application

Introduction

Avec l'inflation des données, plusieurs outils ont été développés pour aider à traiter ces données, les anciennes technologies ne sont plus suffisantes, voir même plus valables dans certains cas.

Dans ce chapitre nous allons présenter les principaux changements dans la manière de traitement des données. Le premier changement principal est celui des systèmes de gestion de base de données avec l'apparition des nouveaux systèmes NoSQL. Le deuxième outil que nous allons présenter est un outil qui a créé une révolution dans ce domaine avec ses performances et ses capacités qui comblent presque toutes les étapes du processus de traitement des données qui a été nommé HADOOP.

1. Les systèmes NoSQL (Not Only SQL)

La naissance des bases de données NoSQL, est du à plusieurs raisons, dont la raison principale est de combler les nouveaux besoins de gestion des données créées aujourd'hui.

1.1. Naissance des bases de données NoSQL

1.1.1. Limites des SGBD relationnelles-transactionnelles

Les SGBD Relationnels offrent un système de jointure entre les tables permettant de construire des requêtes complexes impliquant plusieurs entités, un système d'intégrité référentielle permettant de s'assurer que les liens entre les entités sont valides.

Les SGBD Relationnels sont généralement transactionnels avec une gestion de transactions respectant les contraintes ACID (Atomicity, Consistency, Isolation, Durability), ces derniers ont un contexte fortement distribué, dont les mécanismes ont un coût considérable.

Avec la plupart des SGBD relationnels, les données d'une base de donnée liées entre elles sont placées sur le même nœud du serveur, si le nombre de liens est important, il est de plus en plus difficile de placer les données sur des nœuds différents.

Il y a une nécessité de distribution des traitements de données entre différents serveurs alors il est difficile de maintenir les contraintes ACID à l'échelle du système distribué entier tout en maintenant des performances correctes.

La plupart des SGBD « NoSQL » relâchent les contraintes ACID, ou même ne proposent pas de gestion de transactions.

- **Théorème de Brewer ou de CAP** 3 propriétés fondamentales pour les systèmes distribués :
  - **C oherence** *ou Consistance* : tous les nœuds du système voient exactement les mêmes données au même moment.
  - **A vailability** *ou Disponibilité* : la perte de nœuds n'empêche pas les survivants de continuer à fonctionner correctement, les données restent accessibles.
  - **P artition tolérance** *ou Résistance au partitionnement* : le système étant partitionné, aucune panne moins importante qu'une coupure totale du réseau ne doit l'empêcher de répondre correctement (en cas de partitionnement en sous réseaux, chacun doit pouvoir fonctionner de manière autonome).

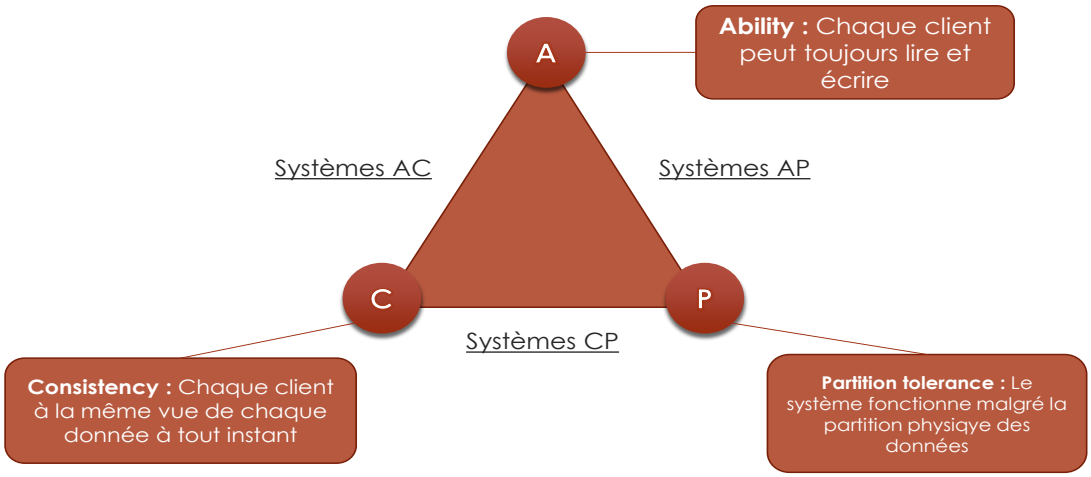


Figure 7. Schéma explicatif du théorème de CAP [25]

- Les **SGBDR** assurent les propriétés de Consistance et de Disponibilité (Availability) => système AC.
- Les **SGBD « NoSQL »** sont des systèmes CP (Cohérent et Résistant au partitionnement) ou AP (Disponible et Résistant au partitionnement).

1.1.2. Les nouveaux besoins en gestion de données

Les NoSQL sont un essor des très grandes plateformes et applications Web (Google, Facebook, Twitter, LinkedIn, Amazon, …), qui répond au problème du volume considérable de données à gérer par ces applications nécessitant une distribution des données et leur traitement sur de nombreux serveurs.

D'où de nouvelles approches de stockage et de gestion des données ont apparu permettant une meilleure scalabilité dans des contextes fortement distribués, une gestion d'objets complexes et hétérogènes sans avoir à déclarer au préalable l'ensemble des champs représentant un objet. Ces approches sont regroupées derrière le terme NoSQL (proposé par Carl Strozzi), ne se substituant pas aux SGBD Relationnels mais les complétant en comblant leurs faiblesses.

Ce sont des Systèmes distribués Système logiciel permettant de coordonner plusieurs ordinateurs, reliés par un réseau local (LAN), Communiquant généralement par envoi de messages, et utilisant une architecture distribuée fonctionnant sur du matériel peu spécialisé et facilement remplaçable en cas de panne.

Application particulière : Distribution de données sur plusieurs serveurs organisés en « data centers » dont le but est de gérer des volumes de données très importants, et d'assurer une continuité de service en cas d'indisponibilité de service sur un serveur.

Dans le cas des Big Data, on dispose d'un très grand ensemble de données sur lesquelles on doit appliquer des traitements à l'aide de **2 stratégies** :

- Par distribution des traitements (scaling des traitements) :
  - On distribue ces traitements sur un nombre de machines important afin d'absorber des charges très importantes

- On envoie les données aux endroits appropriés, et on enchaîne les exécutions distantes (scénario type Workflow implémentable avec des web services)

- Par distribution des données (scaling des données) :
  - On distribue les données sur un nombre important de serveurs afin de stocker de très grands volumes de données
  - On « pousse » les programmes vers ces serveurs (plus efficace de transférer un petit programme sur le réseau plutôt qu'un grand volume de données – Ex : algorithme MapReduce).

1.2. Idée générale sur les systèmes NoSQL

1.2.1. Définition

Les Base de données NoSQL adoptent une représentation de données non relationnelle, elles ne remplacent pas les BD relationnelles mais elles sont une alternative, un complément apportant des solutions plus intéressantes dans certains contextes, ils apportent aussi une plus grande performance dans le contexte des applications Web avec des volumétries de données exponentielle, elles utilisent une très forte distribution de ces données et des traitements associés sur de nombreux serveurs, et elle font un compromis sur le caractère « ACID » des SGBDR pour plus de scalabilité horizontale et d'évolutivité.

L'adoption croissante des bases NoSQL par des grands acteurs du Web (Google, facebook, …) implique une multiplication des offres de systèmes NoSQL.

1.2.2. Caractéristiques

- Pas de schéma pour les données ou existence de schéma dynamique.
- Données de structures complexes ou imbriquées.
- Données distribuées : partitionnement horizontal des données sur plusieurs nœuds (serveurs) généralement par utilisation d'algorithmes «MapReduce».
- Réplication des données sur plusieurs nœuds.
- Privilégient la Disponibilité à la Cohérence (théorème de CAP) : AP (Disponible + Résistant au partitionnement) plutôt que CP (Cohérent + Résistant au partitionnement)
  - ⇨ N'ont en général pas de gestion de transactions.
- Mode d'utilisation : peu d'écritures, beaucoup de lectures.

1.2.3. Typologie des bases de données NoSQL

Stocker les informations de la façon la mieux adaptée à leur représentation => différents types de BD NoSQL :

- Type « Clé-valeur / Key-value » : basique, chaque objet est identifié par une clé unique constituant la seule manière de le requêter
  - Voldemort, Redis, Riak, …
- Type « Colonne / Column » : permet de disposer d'un très grand nb de valeurs sur une même ligne, de stocker des relations « one-to-many », d'effectuer des requêtes par clé (adaptés au stockage de listes : messages, posts, commentaires, ...)
  - HBase, Cassandra, Hypertable, …

- Type « Document » : pour la gestion de collections de documents, composés chacun de champs et de valeurs associées, valeurs pouvant être requêtées (adaptées au stockage de profils utilisateur)
  - MongoDBn CouchDB, Couchbase, …
- Type « Graphe » : pour gérer des relations multiples entre les objets (adaptés aux données issues de réseaux sociaux, …)
  - Neo4j, OrientDB, …

a. Bases de données NoSQL « Clé-valeur »

Elles fonctionnent comme un grand tableau associatif et retourne une valeur dont elle ne connaît pas la structure :

- Leur modèle peut être assimilé à une table de hachage (hashmap) distribuée
- Les données sont simplement représentées par un couple clé/valeur
- La valeur peut être une simple chaîne de caractères, ou un objet sérialisé…
- Cette absence de structure ou de typage ont un impact important sur le requêtage : toute l'intelligence portée auparavant par les requêtes SQL devra être portée par l'applicatif qui interroge la BD.
- Implémentations les plus connues :
  - **Amazon Dynamo** (**Riak** en est l'implémentation Open Source)
  - **Redis** (projet sponsorisé par VMWare)
  - **Voldemort** (développé par LinkedIn en interne puis passage en open source).
- Chaque objet est identifié par une clé unique (seule façon de le requêter)
- La structure de l'objet est libre, souvent laissé à la charge du développeur de l'application (XML, JSON, ...), la base ne gérant généralement que des chaînes d'octets

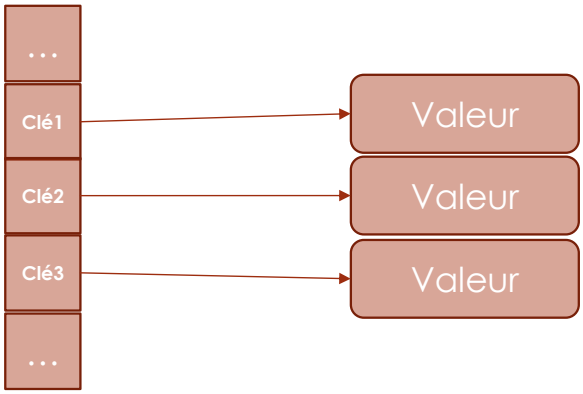


Figure 8. Schéma explicatif des bases de données « Clé-Valeur » [25]

- Leur exploitation est basée sur 4 opérations (CRUD):
  - C reate : créer un nouvel objet avec sa clé → create(key, value)
  - R ead : lit un objet à partir de sa clé → read(key)
  - U pdate : met à jour la valeur d'un objet à partir de sa clé → update(key, value)
  - D elete: supprime un objet à partir de sa clé → delete(key)
- Elles disposent généralement d'une simple interface de requêtage HTTP REST accessible depuis n'importe quel langage de développement
- Ont des performances très élevées en lecture et en écriture et une scalabilité horizontale considérable
- Le besoin en scalabilité verticale est faible du fait de la simplicité des opérations effectuées.

- Utilisations principales des BD NoSQL type « Clés-Valeurs » :
  - Dépôt de données avec besoins de requêtage très simples
  - Système de stockage de cache ou d'information de sessions distribuées (quand l'intégrité relationnelle des données est non significative)
  - Les profils, préférences d'utilisateur
  - Les données de panier d'achat
  - Les données de capteur
  - Les logs de données
  - …
- Forces :
  - Modèle de données simple
  - Bonne mise à l'échelle horizontale pour les lectures et écritures :
    - ⇨ Evolutivité (scalable)
    - ⇨ Disponibilité
    - ⇨ Pas de maintenances requises lors d'ajout/suppression de colonnes
- Faiblesses :
  - Modèle de données TROP simple :
    - ⇨ Pauvre pour les données complexes
    - ⇨ Interrogation seulement sur clé
    - ⇨ Déporte une grande partie de la complexité de l'application sur la couche application elle-même

b. Bases de données NoSQL « Colonne »

- Les données sont stockées par colonne, non par ligne
  - On peut facilement ajouter des colonnes aux tables, par contre l'insertion d'une ligne est plus coûteuse
  - Quand les données d'une colonne se ressemblent, on peut facilement compresser la colonne
  - Modèle proche d'une table dans un SGBDR mais ici le nombre de colonnes :
    - ⇨ est dynamique
    - ⇨ peut varier d'un enregistrement à un autre ce qui évite de retrouver des colonnes ayant des valeurs NULL.
  - Implémentations les plus connues :
    - ⇨ HBase (Open Source de BigTable de Google utilisé pour l'indexation des pages web, Google Earth, Google analytics, ...)
    - ⇨ Cassandra (fondation Apache qui respecte l'architecture distribuée de Dynamo d'Amazon, projet né de chez Facebook)
    - ⇨ SimpleDB de Amazon.
- Les principaux concepts associés sont les suivants :
  - Colonne :
    - ⇨ Entité de base représentant un champ de donnée,
    - ⇨ Chaque colonne est définie par un couple clé / valeur,
    - ⇨ Une colonne contenant d'autres colonnes est nommée supercolonne.
  - Famille de colonnes :
    - ⇨ Permettent de regrouper plusieurs colonnes (ou supercolonnes),
    - ⇨ Les colonnes sont regroupées par ligne,

⇨ Chaque ligne est identifiée par un identifiant unique (assimilées aux Tables dans le modèle relationnel) et sont identifiées par un nom unique.

- Supercolonnes :
  ⇨ Situées dans les familles de colonnes sont souvent utilisées comme les lignes d'une table de jointure dans le modèle relationnel.

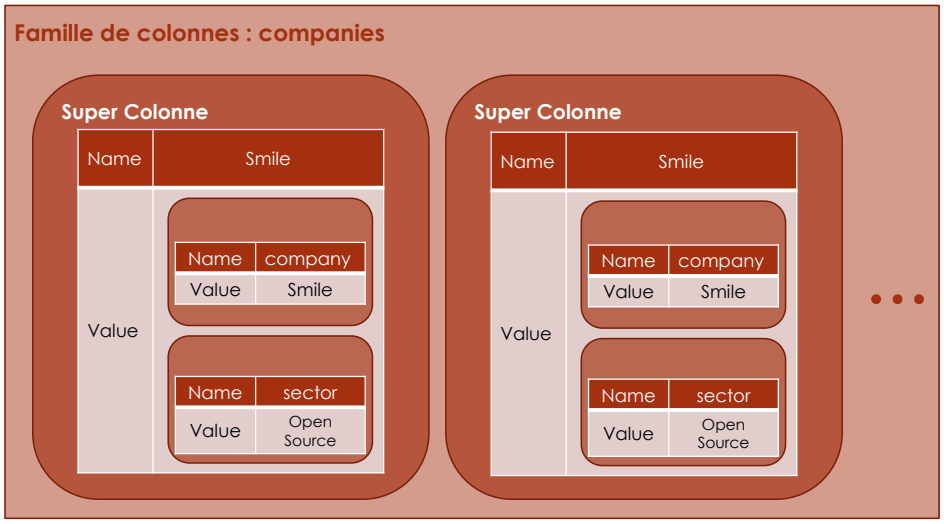


Figure 9. Schéma explicatif des bases de données « Colonne» [25]

- Elles sont les plus complexes à appréhender des BD NoSQL, même si au final on a un schéma assez proche des bases documentaires.
- Elles sont très utilisées pour les traitements d'analyse de données et dans les traitements massifs.
- Elles offrent plus de flexibilité que les BD relationnelles:
  - Il est possible d'ajouter une colonne ou une supercolonne à n'importe quelle ligne d'une famille de colonnes, colonnes ou supercolonne à tout instant.
- Utilisations principales des BD NoSQL type « Clés-Valeurs » :
  - Les BD NoSQL type « Colonne » sont principalement utilisées pour :
    - Netflix l'utilise notamment pour le *logging et l'analyse de sa clientèle*
    - Ebay l'utilise pour l'optimisation de la recherche
    - Adobe l'utilise pour le traitement des données structurées et de Business Intelligence (BI)
    - Des sociétés de TV l'utilisent pour *cerner leur audience* et gérer le *vote des spectateurs* (nb élevé d'écritures rapides et analyse de base en temps réel (Cassandra)
    - Peuvent être de bons magasins d'analyse des données semi-structurées
    - Utilisé pour la journalisation des événements et pour des compteurs
    - …

➨ Forces :

- Modèle de données supportant des données semi-structurées
- Naturellement indexé
- Bonne mise à l'échelle à l'horizontale
- MapReduce souvent utilisé en scaling horizontal
- On peut voir les résultats de requêtes en temps réel

- Faiblesses :
  - A éviter pour des données interconnectées : si les relations entre les données sont aussi importantes que les données elles-mêmes
  - A éviter pour les lectures de données complexes
  - Exige de la maintenance - lors de l'ajout / suppression de colonnes et leur regroupements
  - Les requêtes doivent être pré-écrit, pas de requêtes ad-hoc définis "à la volée": NE PAS utiliser pour les requêtes non temps réel et inconnues.

c. *Bases de données NoSQL « Document »*

- Elles stockent une collection de "documents"
  - Elles sont basées sur le modèle « clé-valeur » mais la valeur est un document en format semi-structuré hiérarchique de type JSON ou XML (possible aussi de stocker n'importe quel objet, via une sérialisation)
  - Les documents n'ont pas de schéma, mais une structure arborescente : ils contiennent une liste de champs, un champ a une valeur qui peut être une liste de champs, ...
  - Implémentations les plus connues :
    - ⇨ CouchDB (fondation Apache)
    - ⇨ RavenDB (pour plateformes « .NET/Windows » - LINQ)
    - ⇨ MongoDB, Terrastore, …
  - Un document est composé de champ et des valeurs associées
  - Ces valeurs :
    - ⇨ Peuvent être requêtées
    - ⇨ Sont soit d'un type simple (entier, chaine de caractère, date, ...)
    - ⇨ Soit elles-mêmes composées de plusieurs couples clé/valeur.
  - Bien que les documents soient structurés, ces BD sont dites "schemaless", il n'est pas nécessaire de définir au préalable les champs utilisés dans un document
  - Les documents peuvent être très hétérogènes au sein de la BD
  - Permettent d'effectuer des requêtes sur le contenu des documents/objets ce qui n'est pas possible avec les BD clés/valeurs simples
  - Elles sont principalement utilisées dans le développement de CMS (Content Management System - outils de gestion de contenus).

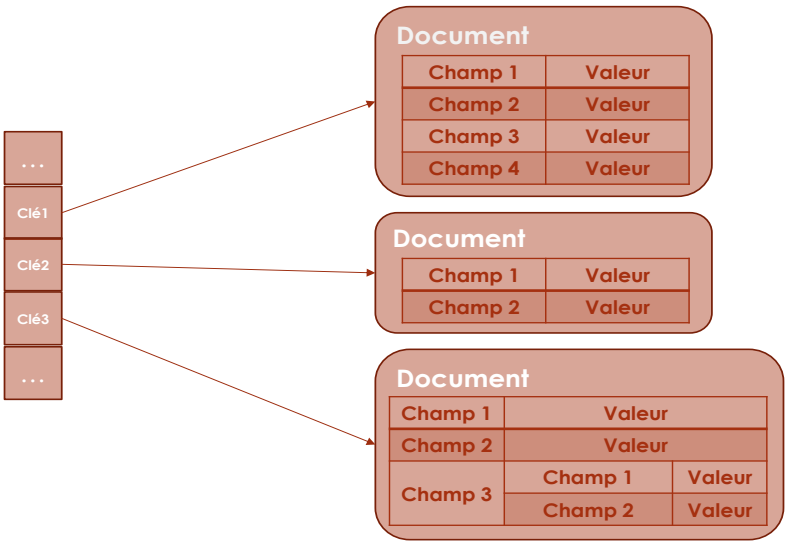


Figure 10. Schéma explicatif des bases de données « document » [25]

- Utilisations principales des BD NoSQL type « Document » :
  - Enregistrement d'événements
  - Systèmes de gestion de contenu
  - Web analytique ou analytique temps-réel
  - Catalogue de produits
  - Systèmes d'exploitation
  - …
- Forces :
  - Modèle de données simple mais puissant
  - Pas de maintenance de la BD requise pour ajouter/supprimer des «colonnes»
  - Forte expressivité de requêtage (requêtes assez complexes sur des structures imbriquées)
- Faiblesses :
  - Inadaptée pour les données interconnectées
  - Modèle de requête limitée à des clés (et indexes) peut alors être lent pour les grandes requêtes

d. Bases de données NoSQL « Graphe »

- Permettent la modélisation, le stockage et la manipulation de données complexes liées par des relations non-triviales ou variables
- Modèle de représentation des données basé sur la théorie des graphes
- S'appuie sur les notions de nœuds, de relations et de propriétés qui leur sont rattachées.
- Implémentations les plus connues :
  - Neo4J
  - OrientDB (fondation Apache)
  - …
- Elles utilisent :
  - Un moteur de stockage pour les objets (similaire à une base documentaire, chaque entité de cette base étant nommée nœud)
  - Un mécanisme de description d'arcs (relations entre les objets), arcs orientés et avec propriétés (nom, date, ...)
- Elles sont bien plus efficaces que les BDR pour traiter les problématiques liées aux réseaux (cartographie, relations entre personnes, ...)
- Sont adaptées à la manipulation d'objets complexes organisés en réseaux: cartographie, réseaux sociaux...

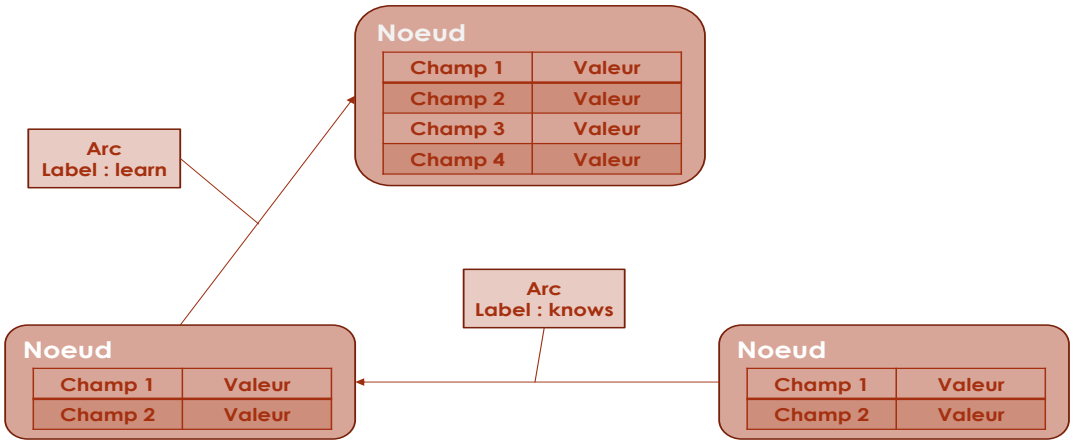


Figure 11. Schéma explicatif des bases de données « Graphe » [25]

- Utilisations principales des BD NoSQL type « Graphe » :
  - Moteurs de recommandation
  - Business Intelligence (BI)
  - Semantic Web
  - Social computing
  - Données géospatiales
  - Généalogie
  - Web of things
  - Catalogue des produits
  - Sciences de la Vie et calcul scientifique (bioinformatique, …)
  - Données liées, données hiérarchiques
  - Services de routage, d'expédition et de géolocalisation
  - Services financiers : chaîne de financement, dépendances, gestion des risques, détection des fraudes, …
- Forces :
  - Modèle de données puissant
  - Rapide pour les données liées, bien plus rapide que SGBDR
  - Modèles d'interrogation bien établis et performants : Tinkerpop pile (fournit un ensemble commun d'interfaces permettant aux différentes technologies informatiques graphiques de travailler ensemble, que le développeur utilise en cas de besoin), SPARQL et Cypher
- Faiblesses :
  - Fragmentation (sharding) :
    - Même si elles peuvent évoluer assez bien
    - Pour certains domaines, on peut aussi fractionner.

2. Le Framework Hadoop

Hadoop est un Framework java qui est très populaire lors des traitements des Big Data à cause de son écosystème qui aide à la réalisation des différentes étapes de ce traitement, Dans cette partie nous allons essayer de présenter ce Framework et ses différents composants.

2.1. Présentation de Hadoop

2.1.1. Diffèrents modes de Hadoop

Hadoop est utilisé avec trois modes différents: mode standalone, mode pseudo, et mode complètement réparti.

- **Le mode standalone:**

Dans ce mode, on peut ne pas commencer tous les démons Hadoop. Au lieu de cela, il suffit d'appeler « ~/Hadoop-directory/bin/hadoop » et Hadoop va exécuter une opération Hadoop comme un processus unique de Java. Ceci est recommandé pour des fins de test. Ceci est le mode par défaut et il n y'a aucun besoin de configurer quoi que ce soit. Tous les démons, comme NameNode, DataNode, JobTracker et TaskTracker sont exécutés dans un seul processus Java.

- **Le mode pseudo:**

Dans ce mode, Hadoop est configuré pour tous les nœuds. Une machine virtuelle Java (JVM) est généré pour chacun des composants Hadoop ou démons comme des mini clusters sur un seul hôte.

➡ **Le mode complètement réparti:**

Dans ce mode, Hadoop est répartie sur plusieurs machines. Les hôtes dédiés sont configurés pour les composants Hadoop. Par conséquent, les processus de la JVM distinctes sont présents pour tous les démons [26].

2.1.2. Architecture Hadoop

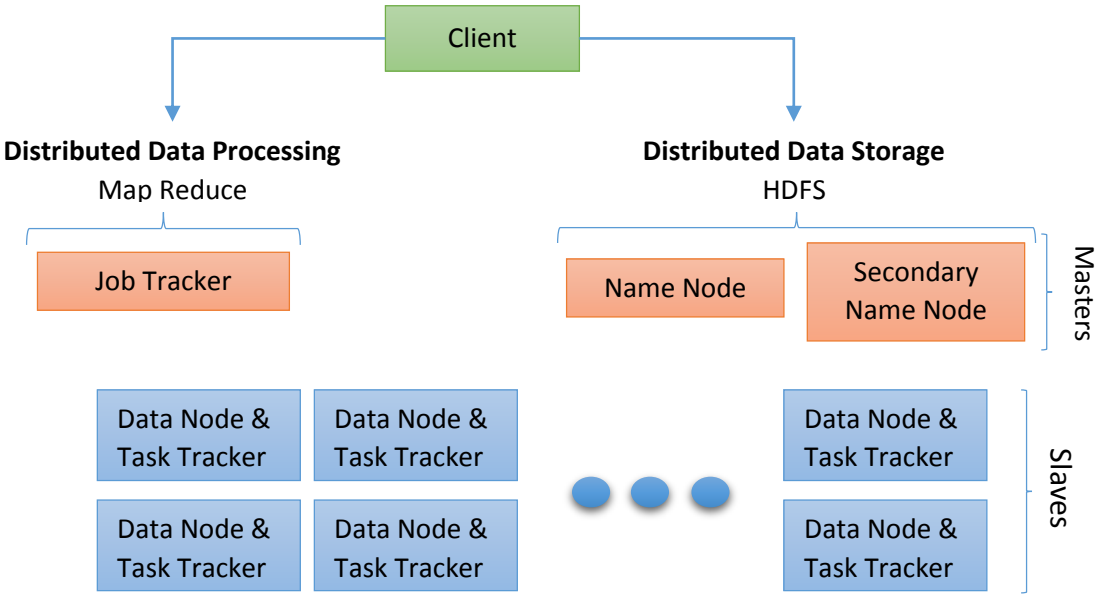


Figure 12. Architecture de Hadoop

Hadoop est spécialement conçu pour deux concepts de base: HDFS et MapReduce. Les deux sont basés sur les calculs distribués. MapReduce est considéré comme le cœur de Hadoop qui effectue un traitement parallèle sur des données distribuées.

2.1.3. HDFS

HDFS est dérivée du concept de système de fichiers de Google. Une caractéristique importante de Hadoop est le partitionnement de données et de calcul à travers de nombreux (plusieurs milliers) d'unités, et l'exécution de calculs d'applications en parallèle, à proximité de leurs données. Sur HDFS, les fichiers de données sont répliqués en tant que séquences de en des clusters [26].

- Caractéristiques de HDFS :
  - Tolérant aux pannes ;
  - Fonctionne avec du matériel de base ;
  - Capable de gérer de grands ensembles de données ;
  - Paradigme de maître-esclave.

- Architecture de HDFS :

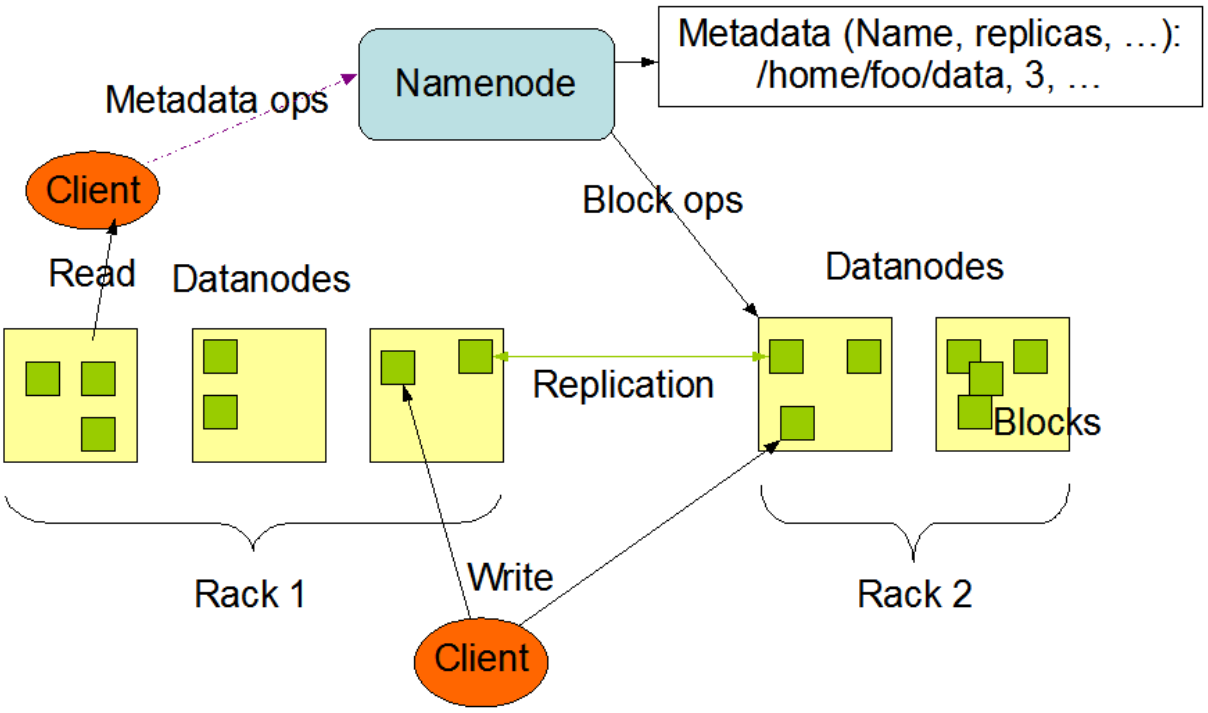


Figure 13. Architecture du HDFS [27]

HDFS peut être présenté comme une architecture maître/esclave. Le maître de HDFS est nommé NameNode l'esclave est DataNode.

Le NameNode est un serveur qui gère l'espace de noms du système de fichiers et ajuste l'accès (ouverture, fermeture, renommer..) à des fichiers par client. Il divise les données d'entrée en blocs et annonce quel bloc de données sera stocké dans quel DataNode.

La DataNode est une machine esclave qui stocke les répliques de l'ensemble de données partitionné et sert les données une fois les requêtes proviennent. Il effectue également la création et la suppression des blocs.

Le mécanisme interne de HDFS divise le fichier dans un ou plusieurs blocs; ces blocs sont stockés dans un ensemble de nœuds de données. Dans des circonstances normales de réplication le facteur de réplication est trois, la stratégie HDFS est de placer la première copie sur le nœud local, la deuxième copier sur l'unité locale avec un nœud différent, et une troisième copie dans différentes unités avec différents nœuds.

Comme HDFS est conçu pour supporter de gros fichiers, la taille de bloc HDFS est définie comme 64 MB. Si nécessaire, cela peut être augmenté.

- **Composant de HDFS :**

HDFS est géré avec l'architecture maître-esclave qui inclut les composants suivants:

- NameNode: Maître du système HDFS. Il maintient les répertoires, fichiers, et gère les blocs qui sont présents sur les DataNodes.
- DataNode: Ce sont des esclaves qui sont déployés sur chaque machine et fournissent le stockage réel. Ils sont responsables de servir des données de lecture-écriture pour les clients.

- NameNode secondaire: Il est chargé de l'exécution des points de contrôle périodiques. Donc, si le NameNode échoue à tout moment, il peut être remplacé par une image instantanée stockée par les points de contrôle de NameNode secondaires.

2.1.4. MapReduce

Le MapReduce est un modèle de programmation pour le traitement de grands jeux de données répartis sur un grand cluster. MapReduce est le cœur de Hadoop. Son paradigme de programmation permet d'effectuer le traitement massif de données sur des milliers de serveurs configurés avec des clusters Hadoop. Il est dérivé de Google MapReduce [26].

MapReduce de Hadoop est un cadre logiciel pour écrire des applications facilement, qui traite de grandes quantités de données (jeux de données de plusieurs térabytes) en parallèle sur de grandes clusters (en milliers de nœuds) de matériel de base d'une manière fiable, et à tolérance de pannes.

Ce paradigme MapReduce est divisé en deux phases, le mapping et la réduction qui traitent principalement des paires de données clé-valeur. Les tâches du mapping et de la réduction s'exécutent séquentiellement dans un cluster, la sortie de la phase du mapping devient l'entrée pour la phase de réduction. Ces phases sont expliquées comme suit:

- **Phase de mapping** : Une fois divisé, les ensembles de données sont affectées à la tâche tracker pour effectuer la phase du mapping. Après effectuation de la tache on a en sortie des paires de clés et de valeurs.
- **Phase de réduction** : Le nœud maître recueille alors les réponses à tous les sous-problèmes et les combine en quelque sorte pour former la sortie.

- Architecture de MapReduce :

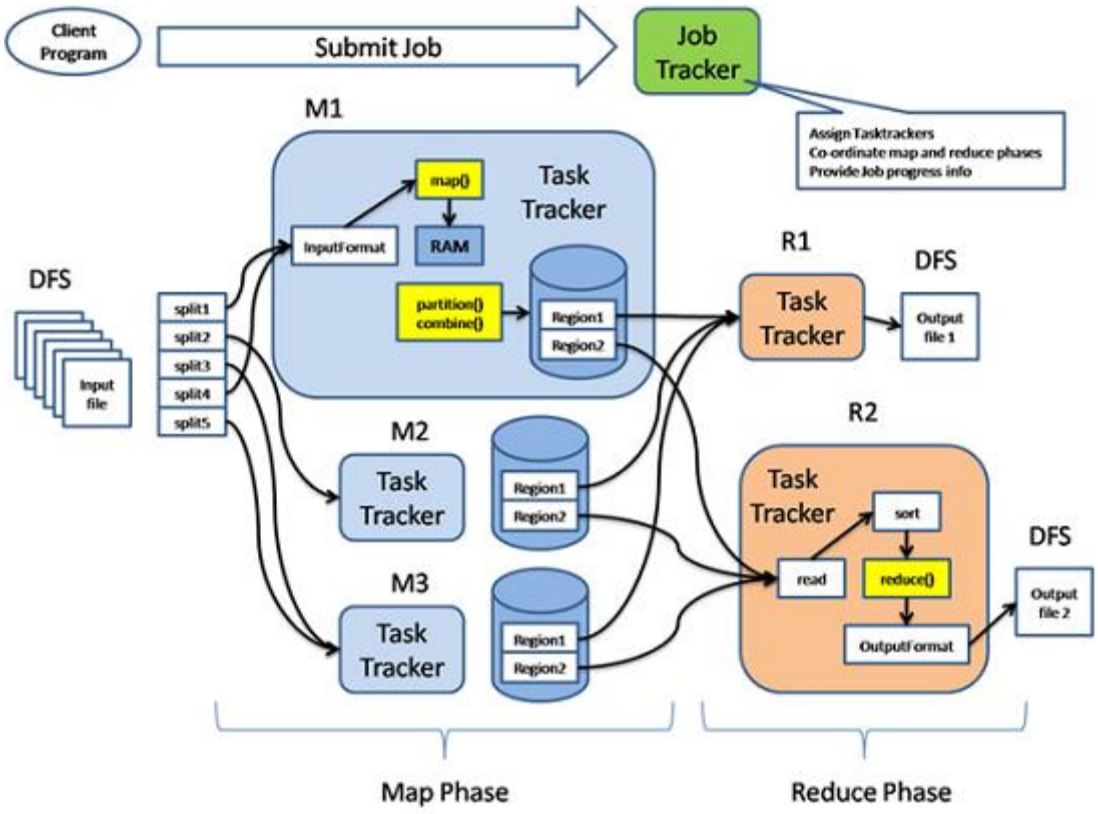


Figure 14. Architecture de MapReduce [29]

MapReduce est également mis en œuvre par sous une architecture maître-esclave. Le MapReduce Classique contient la soumission de jobs, l'initialisation des jobs, la répartition des tâches, l'exécution des tâches, le progrès et le statut de la mise à jour, et les activités liées à l'achèvement des jobs, qui sont principalement gérés par le nœud de JobTracker et exécutés par le nœud TaskTracker.

L'application client soumet une tâche à au JobTracker. Puis l'entrée est divisée sur le cluster. Le JobTracker calcule ensuite le nombre de maps et de réductions à traiter. Il commande le TaskTracker pour commencer à exécuter le travail. Les TaskTracker copient les ressources dans une machine locale et lance les JVM aux mapping et à la réduction sur les données. Parallèlement à cela, le TaskTracker envoie périodiquement la mise à jour du JobTracker, qui permet de mettre à jour l'identificateur du job (JobID), le statut du job, et l'utilisation des ressources.

- Composants de MapReduce :

MapReduce est géré avec une architecture maître-esclave inclus avec les composants suivants:

- JobTracker: Ceci est le nœud maître du système de MapReduce, qui gère les jobs et les ressources dans le cluster TaskTracker. Le JobTracker essaie de planifier chaque Map au plus près des données réelles en cours de traitement sur le TaskTracker, qui est en cours d'exécution sur le même DataNode que le bloc sous-jacente.
- TaskTracker: Ce sont les esclaves qui sont déployés sur chaque machine. Ils sont responsables de l'exécution du mapping et de la réduction des tâches selon les instructions du JobTracker.

2.1.5. Yarn

Le yarn est ou bien le Hadoop MapReduce NextGen est la deuxième version de MapReduce, l'idée fondamentale de MRv2 est de diviser les deux grandes fonctionnalités du JobTracker, gestion des ressources et planification/surveillance des jobs, en deux démons séparés. L'idée est d'avoir un ResourceManager (RM) global et un ApplicationMaster (AM) par application. Une application est soit un seul job dans le sens classique de MapReduce ou un DAG de jobs [30].

Le ResourceManager et l'esclave par nœud, le NodeManager (NM), forment le Framework de calcul de données. Le ResourceManager est l'autorité ultime qui arbitre les ressources entre toutes les applications dans le système.

L'ApplicationMaster par application est, en effet, une bibliothèque spécifique qui est chargé de la négociation des ressources du ResourceManager et utilise le NodeManager pour exécuter et suivre les tâches.

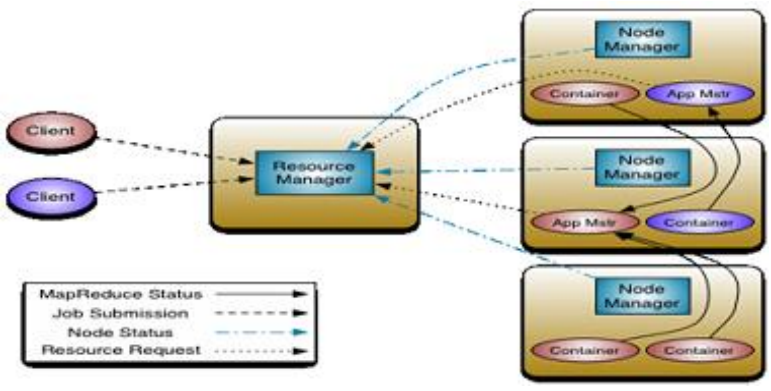


Figure 15. Architecture du Yarn [30]

Le ResourceManager comporte deux composantes principales: l'ordonnanceur (Scheduler) et le manager d'applications (ApplicationsManager).

L'ApplicationsManager est responsable de l'acceptation des soumissions de jobs, la négociation du premier conteneur pour l'exécution de l'ApplicationMaster d'application spécifique et fournit le service pour redémarrer le conteneur ApplicationMaster en cas d'échec.

Le NodeManager est le Framework par machine qui est chargé de conteneurs, le suivi de leur utilisation des ressources (CPU, mémoire, disque, réseau) et les rapports de la ressemblance au ResourceManager/Scheduler.

L'ApplicationMaster par application est responsable de négocier les conteneurs de ressources appropriées du Scheduler, le suivi de leur état et de suivi des progrès.

Le MRV2 maintient la compatibilité API avec la version stable précédente (hadoop-1.x). Cela signifie que tous les taches Mapping et réduction devraient toujours fonctionner de la même manière sur le dessus de MRv2 avec juste une recompilation.

2.2. Composants Apache Hadoop

2.2.1. Hbase

HBase est un entrepôt de Big Data distribué pour Hadoop qui permet un accès lecture/écriture aléatoire, en temps réel aux Big Data. Il est conçu comme un modèle de stockage de données en colonnes innové après inspiré par Google BigTable [26].

2.2.2. HCatalog

HCatalog est une couche de gestion de table et de stockage pour Hadoop qui permet aux utilisateurs avec différents outils de traitement des données, Apache Pig, Apache MapReduce, et Apache Hive de lire plus facilement et écrire des données sur une grille.

L'abstraction des tables dans HCatalog présente aux utilisateurs une vue relationnel de données dans le système de fichiers distribués de Hadoop (HDFS) et garantit que les utilisateurs se soucient pas sur où et dans quel format de leurs données sont stockée. HCatalog affiche les données du format RCFile, des fichiers texte ou des fichiers de séquence dans une vue tabulaire. Il fournit également des API REST pour que les systèmes externes puissent accéder aux métadonnées de ces tables [31].

2.2.3. Hive

Hive est un entrepôt de données basé sur Hadoop comme cadre élaboré par Facebook. Il permet aux utilisateurs de lancer des requêtes dans des langages SQL-like, comme HiveQL, qui sont fortement abstraite à Hadoop MapReduce. Cela permet aux programmeurs SQL sans aucune expérience sur MapReduce à utiliser le warehouse et le rend plus facile à intégrer avec le business intelligence et les outils de visualisation pour le traitement des requêtes en temps réel [26].

2.2.4. Pig

Pig est une plate-forme open source basé sur Hadoop pour analyser les grands ensembles de données à grande échelle par l'intermédiaire de son propre langage SQL-like: Pig Latin.

Ceci fournit une interface simple d'utilisation et de programmation pour de massifs et complexes de calcul de données. Il est également plus facile à développer, il est plus optimisé et extensible. Apache Pig a été développé par Yahoo.

Actuellement, Yahoo et Twitter sont les principaux utilisateurs de Pig. Pour les développeurs, l'utilisation directe des API Java peut être fastidieux, mais limite aussi l'utilisation par le programmeur Java de la flexibilité de la programmation Hadoop. Donc, Hadoop fournit deux solutions qui permettent de rendre la programmation Hadoop pour la gestion des données et l'analyse de données avec MapReduce plus facile, ce sont Pig et Hive, qui sont toujours source de confusion [26].

2.2.5. Sqoop

Hadoop fournit une plate-forme de traitement de données et bases de données relationnelles, data warehouse, et d'autres bases de données non-relationnelles qui transfèrent rapidement de grandes quantités de données dans une nouvelle façon. Apache Sqoop est un outil mutuel de données pour importer des données à partir des bases de données relationnelles à Hadoop HDFS et à l'exportation de données à partir de bases de données relationnelles HDFS.

Il fonctionne avec la plupart des bases de données relationnelles modernes, telles que MySQL, PostgreSQL, Oracle, Microsoft SQL Server et IBM DB2, et de l'entrepôt de données d'entreprise. L'API d'extension Sqoop fournit un moyen pour créer de nouveaux connecteurs pour le système de base de données. En outre, la source Sqoop arrive avec certains connecteurs de bases de données populaires. Pour effectuer cette opération, Sqoop transforme d'abord les données dans Hadoop MapReduce avec une certaine logique de la création et de la transformation du schéma de base de données [26].

2.2.6. Flume

Apache Flume, est un service disponible fiable de collecte efficace des données, de l'agrégation, et du déplacement de grandes quantités de données en continu dans le système de fichier distribué de Hadoop (HDFS). Il a une architecture simple et flexible basée sur le streaming des flux de données; il est robuste et tolérant aux pannes avec des mécanismes de fiabilité accordables pour le basculement et la récupération [32].

2.2.7. Oozie

Apache oozie est une application Web Java utilisée pour planifier des tâches Apache Hadoop. oozie combine plusieurs jobs séquentiellement dans une unité de travail logique. Il est intégré avec la pile Hadoop, avec Yarn comme son centre d'architecture, et il soutient les jobs pour MapReduce, Apache Pig, Apache Hive, et Apache Sqoop. Oozie peut également planifier des tâches spécifiques à un système, comme les programmes Java ou les scripts shell [28].

2.2.8. Zookeeper

Zookeeper est également un sous-projet Hadoop utilisé pour la gestion des autres composants de hadoop, Hive, Pig, HBase, Solr, et d'autres projets. Zookeeper est un service de coordination des applications open source distribué, qui gère la synchronisation et de la configuration et des services de nommage tels que la maintenance des applications distribuées. Dans la programmation, la conception Zookeeper est un style de modèle de données très simple, tout comme la structure de l'arbre des systèmes répertoires [26].

2.2.9. Ambari

Apache Ambari est un outil basé sur le Web qui prend en charge le pôle d'approvisionnement, de la gestion et de la surveillance de Hadoop. Ambari gère la plupart des composants Hadoop, notamment HDFS, MapReduce, Hive, Pig, HBase, Zookeeper, Sqoop et HCatlog entant gestion centralisée.

En outre, Ambari est capable d'installer la sécurité basée sur le protocole d'authentification Kerberos sur le cluster Hadoop. De plus, il fournit une authentification basée sur les rôles d'utilisateur, l'autorisation et les fonctions d'audit pour que les utilisateurs gèrent LDAP intégré et Active Directory [26].

2.2.10. Mahout

Mahout est une bibliothèque populaire de data mining. Il lui faut des algorithmes les plus populaires de data mining, pour effectuer le regroupement, la classification, la régression et la modélisation statistique pour créer des applications intelligentes. En outre, c'est une bibliothèque d'apprentissage automatique évolutive.

Apache Mahout est distribué sous une licence commerciale conviviale du logiciel Apache. Le but de Mahout Apache est de construire une communauté dynamique, réactive et diversifiée pour faciliter les discussions non seulement sur le projet lui-même mais aussi sur d'autres cas d'utilisation potentiels. [26]

2.2.11. Avro

Apache Avro ™ est un système de sérialisation de données. Qui fournit:

- Des Structures de données riches.
- Un format compact, rapide, des données binaires.
- Un fichier conteneur, pour stocker les données persistantes.
- Appel de procédures distantes (RPC).
- Intégration simple avec les langages dynamiques. La génération de code n'est pas nécessaire pour lire ou écrire des fichiers de données, ni pour utiliser ou mettre en œuvre des protocoles RPC. La génération de code est comme une option d'optimisation, dont la mise en œuvre n'est nécessaire que pour les langages statiquement typés [33].

2.2.12. Hue

Hue est un ensemble d'applications web qui permettent d'interagir avec un cluster CDH. L'Application Hue permet de parcourir les HDFS et de travailler avec les requêtes de Hive et de Cloudera Impala, les jobs de MapReduce, et les workflows d'oozie...[34]

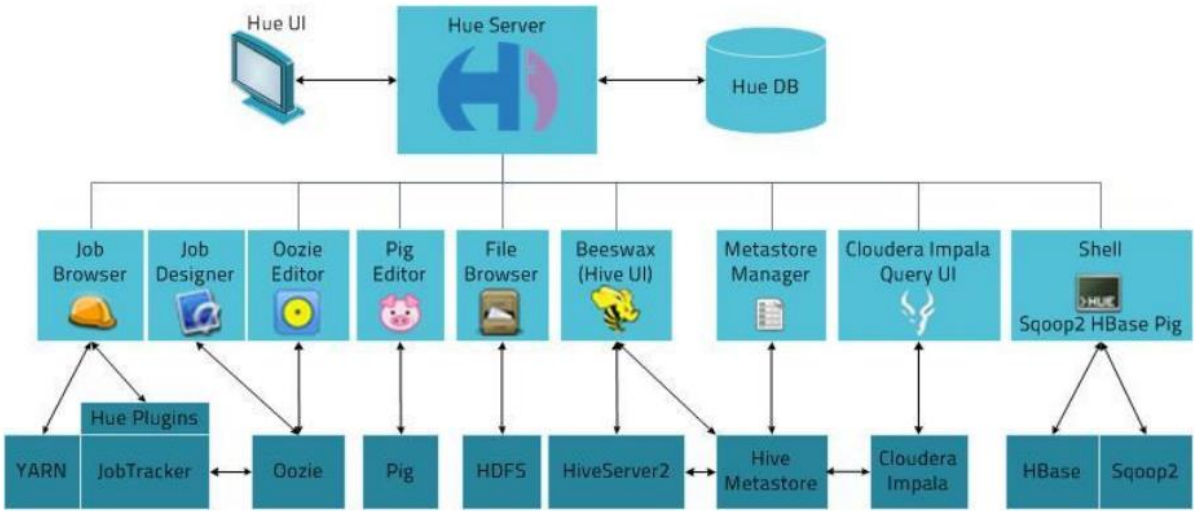


Figure 16. Architecture Hue [34]

2.2.13. Solr

Solr est plateforme logiciel open source de recherche. Apache Solr est hautement évolutive, qui supporte des moteurs de recherche distribués et la réplication de l'index. Cela permet de créer des applications Web avec des moteurs de recherche de texte puissants, la recherche à facettes, l'indexation en temps réel, le clustering dynamique, l'intégration de base de données, et de traitement des documents riche [26].

3. Langage R

Comme outils qui ont montré leur efficacité dans le domaine des Big Data, on trouve le langage R, qui est un langage de programmation qui traite des données et des analyses statistiques. Le langage R peut être utilisé en le combinant avec Hadoop pour augmenter la capacité d'analyse chez Hadoop.

Il y a plus de 3000 paquets de R et la liste augmente de jour en jour. Il serait au-delà de la portée de tout livre, même de tenter d'expliquer tous ces paquets. On va se concentrer uniquement sur les principales caractéristiques de R et packages les plus fréquemment utilisés et les plus populaires [26].

3.1. Opérations sur les données

R permet à un large éventail d'opérations. Les opérations statistiques, telles que la moyenne, min, max, la probabilité, la distribution, et la régression. Les opérations d'apprentissage machine, comme la régression linéaire, la régression logistique, la classification et le clustering. Les opérations de traitement de données universel :

- Le nettoyage des données: nettoyer les ensembles de données massifs
- L'exploration de données: explorer toutes les valeurs possibles d'ensembles de données
- L'analyse des données: effectuer des analyses sur des données avec la visualisation des données d'analyse descriptives et prédictives, qui est, la visualisation de l'analyse des programmes de sortie

Pour construire une application d'analyse efficace, nous avons parfois besoin d'utiliser la programmation API pour déterrer les données, de les analyser avec des services opportuns, et de les visualiser par des services tiers. Aussi, pour automatiser le processus d'analyse de données, la programmation sera la caractéristique la plus utile à quoi on doit faire face.

R possède son propre langage de programmation pour opérer sur les données. En outre, le paquet disponible peut aider à intégrer R avec d'autres fonctions de programmation. R prend en charge les concepts de la programmation orientée objet. Il est également capable de s'intégrer avec d'autres langages de programmation tels que Java, PHP, C, et C ++.

Il existe plusieurs paquets qui agissent comme des fonctions de couche intermédiaire de programmation pour aider à l'analyse de données [26].

3.2. Modélisation de données dans R

La modélisation des données est une technique d'apprentissage machine pour identifier le modèle caché de la base de données anciennes, et ce modèle aidera à prédire la valeur sur les mêmes données.

Cette technique se concentre fortement sur les anciennes actions de l'utilisateur et apprend leur goût et leurs choix. La plupart de ces techniques de modélisation de données ont été adoptées par de nombreuses organisations populaires pour comprendre le comportement de leurs clients en fonction de leurs anciennes transactions.

Amazon, Google, Facebook, eBay, LinkedIn, Twitter, et de nombreuses autres organisations utilisent l'extraction de données pour modifier les applications de définition. Les techniques d'exploration de données les plus courants sont les suivants [26]:

- La régression
- La classification
- Le clusering
- La recommandation

- **La régression:**

Cette relation aidera à prédire la valeur de la variable de futurs événements. Les prévisions de ventes de produits ou de services et les prévisions du prix des stocks peut être atteint grâce à cette régression. R fournit cette fonction de régression par la méthode de lm, qui est par défaut présente dans R.

- **La classification:**

Permet de classer les observations en une ou plusieurs étiquettes. La probabilité de vente, la détection de la fraude en ligne, et la classification du cancer (pour la science médicale) sont des applications courantes des problèmes de classification. Google Mail utilise cette technique pour classer les e-mails comme spam ou non. Les caractéristiques de classification peuvent être servies par GLM, glmnet, ksvm, SVM, et randomForest dans R.

- **Le clustering:**

Cette technique est tout au sujet de l'organisation des articles similaires dans des groupes à partir de la collection de pièces. La segmentation de l'utilisateur et la compression d'image sont les

applications les plus courantes de clustering. La segmentation du marché, analyse de réseau social, en organisant le regroupement de l'ordinateur, et l'analyse des données astronomiques sont aussi des applications de clustering. 'Google News' utilise ces techniques pour grouper les articles d'actualité similaires dans la même catégorie. Le clustering peut être atteint par les KNN, KMeans, dist, pvclust et méthodes Mclust à R.

- **La recommandation:**

Les algorithmes de recommandation sont utilisés dans les systèmes de recommandation où ces systèmes sont des techniques d'apprentissage automatique. Amazon est un portail e-commerce bien connu qui génère 29% du chiffre d'affaires par le biais de systèmes de recommandation. Les systèmes de recommandation peuvent être mises en œuvre par recommander() avec le paquet de recommenderlab dans R.

4. Combinaison de Hadoop et R

L'analyse avec R rencontre plusieurs problèmes liés à la grande quantité de données. R charge les données en mémoire sauf si l'ensemble de données est grand. Par conséquent, afin de traiter de grands ensembles de données, la puissance de traitement de R peut être considérablement amplifiée en la combinant avec la puissance d'un cluster Hadoop avec ces capacités de traitement parallèle. Ainsi, des algorithmes de R peuvent être utilisés ou bien un traitement sur les clusters Hadoop pourra effectuer les tâches de traitement des Big Data avec succès [26].

4.1. RHipe

RHIPE signifie l'environnement de programmation intégré de R et Hadoop [35]. C'est une fusion de R et Hadoop. Il a d'abord été développé par Saptarshi Guha pour sa thèse de doctorat au Département de statistique de l'Université Purdue en 2012. Actuellement elle est poursuite par l'équipe de département de statistique à l'Université Purdue et d'autres groupes de discussion Google actives.

Le paquet RHIPE utilise la technique de division et de recombinaison pour effectuer des analyses de données sur les Big Data. Dans cette technique, les données sont divisées en sous-ensembles, le calcul est effectué sur ces sous-ensembles par des opérations spécifiques d'analyse de R, et la sortie est combinée. RHIPE a principalement été conçu pour atteindre deux objectifs qui sont comme suit:

- permettre d'effectuer une analyse en profondeur de grandes ainsi que les petites données.
- Permettre aux utilisateurs d'effectuer les opérations d'analyse au sein de R en utilisant un langage de niveau inférieur.

RHIPE est conçu avec plusieurs fonctions qui aident à exécuter le système de fichiers distribué de Hadoop (HDFS) ainsi que les opérations MapReduce utilisant une console R simple [26].

4.1.1. Architecture de RHipe

Nous allons maintenant essayer de comprendre le fonctionnement de l'ensemble de la bibliothèque RHIPE développé pour intégrer R et Hadoop pour une analyse efficaces des Big Data.

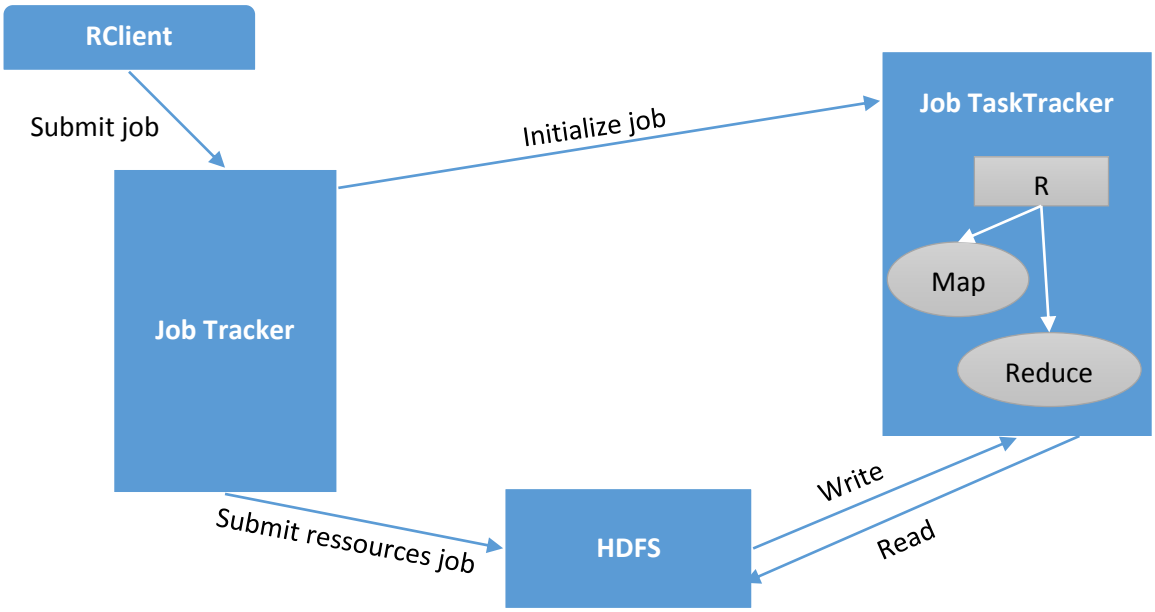


Figure 17. Composants de RHIPE [26]

Il existe un certain nombre de composants Hadoop qui seront utilisés pour des opérations d'analyse de données avec R et Hadoop. Les composants de RHIPE sont les suivantes:

- **RClient**: RClient est une application de R qui appelle le JobTracker pour exécuter le job avec l'indication de plusieurs ressources de jobs de MapReduce tels que Mapper, Réducteur, format d'entrée, format de sortie, fichier d'entrée, fichier de sortie, et d'autres plusieurs paramètres qui peuvent gérer les jobs MapReduce avec RClient.
- **JobTracker**: Un JobTracker est le nœud maître des opérations MapReduce Hadoop pour l'initialisation et le suivi des jobs MapReduce sur le cluster Hadoop.
- **TaskTracker**: TaskTracker est un nœud esclave dans le cluster Hadoop. Il exécute les jobs MapReduce selon les ordres donnés par le JobTracker, récupère les blocs de données d'entrée, et exécute un Mapper spécifique à R et effectue une réduction au-dessus. Enfin, la sortie sera écrite sur le répertoire HDFS.
- **HDFS**: HDFS est un système de fichiers répartis sur les clusters Hadoop avec plusieurs nœuds de données. Il fournit des services de données pour diverses opérations sur les données.

4.2. RHadoop

RHadoop est une collection de trois packages R pour fournir de grandes opérations de données avec un environnement R. Il a été développé par Revolution Analytics, qui est le premier fournisseur commercial d'un logiciel basé sur R. RHadoop est disponible avec trois principaux packages R: rhdfs, rmr, et rhbase. Chacun d'entre eux offrent des services de Hadoop différents.

- **rhdfs** : C'est une interface de R qui fournit la facilité d'utilisation de HDFS dans une console R. Comme les programmes Hadoop MapReduce écrivent leur sortie sur HDFS, il est très facile d'y accéder en appelant les méthodes de rhdfs. Le programmeur de R peut facilement effectuer les opérations d'écriture et de lecture sur les fichiers de données distribuées. Fondamentalement, le paquet rhdfs appelle l'API HDFS pour exploiter les sources de données stockées sur HDFS.

- **rmr** : C'est une interface de R pour fournir une facilité d'emploi MapReduce Hadoop à l'intérieur de l'environnement. Ainsi, le programmeur de R doit seulement diviser sa logique d'application dans les phases de mapping et de réduction et les soumettre avec les méthodes de rmr. Après cela, rmr appelle l'API MapReduce streaming avec plusieurs paramètres de job comme répertoire d'entrée, répertoire de sortie, mapper, réducteur, et ainsi de suite, pour effectuer le job de R MapReduce sur un cluster Hadoop.
- **rhbase** : C'est une interface de R pour faire fonctionner la source de données HBase stockée au niveau du réseau distribué par l'intermédiaire d'un serveur d'épargne. Le paquet de rhbase est conçu avec plusieurs méthodes pour l'initialisation et les opérations lecture/écriture.

Il n'est pas nécessaire d'installer tous les trois packages RHadoop pour exécuter les opérations MapReduce Hadoop avec R et Hadoop. Si nous avons stocké notre source de données d'entrée dans HBase, nous devons installer rhbase.

Comme Hadoop est plus populaire pour ses deux caractéristiques principales, MapReduce Hadoop et HDFS, ces deux fonctionnalités seront utilisées dans la console de R avec l'aide de rhdfs de RHadoop et les packets rmr. Ces packages sont suffisants pour exécuter MapReduce Hadoop à partir de R. Fondamentalement, rhdfs fournit des opérations sur les données HDFS alors que rmr fournit des opérations d'exécution de MapReduce.

RHadoop comprend également un autre paquet appelé vérification rapide, qui est conçu pour le débogage des jobs développés de MapReduce défini par le paquet rmr.

4.2.1. Architecture de RHadoop

Puisque Hadoop est très populaire en raison de HDFS et MapReduce, 'Revolution Analytics' a développé des packages séparés de R, à savoir, rhdfs, rmr, et rhbase. L'architecture de RHadoop est indiquée dans le schéma suivant:

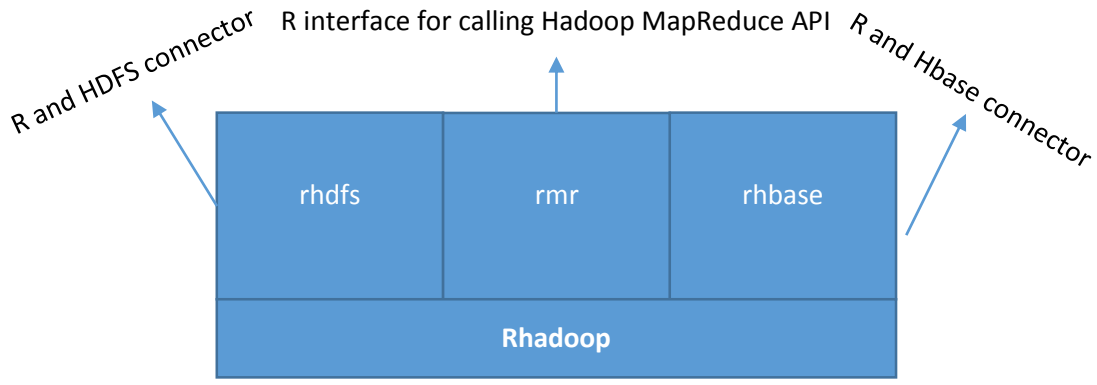


Figure 18. Ecosystème de Rhadoop [26]

4.3. Hadoop streaming

Hadoop streaming est un utilitaire de Hadoop pour exécuter les jobs MapReduce Hadoop avec des scripts exécutables tels que le Mapper et le réducteur. Avec cela, le fichier d'entrée est imprimé sur un flux (stdin), qui est fourni en tant qu'entrée au Mapper et la sortie (stdout) du Mapper est fournie comme une entrée au réducteur, enfin, le réducteur écrit la sortie dans le répertoire HDFS.

Le principal avantage du streaming Hadoop est qu'il permet aux jobs programmé Java ainsi qu'aux jobs non-Java MapReduce d'être exécutées sur des clusters Hadoop. En outre, il prend soin de l'état d'avancement d'exécution des jobs MapReduce. La diffusion Hadoop supporte les langages de programmation Perl, Python, PHP, R et C ++. Pour exécuter une application écrite dans d'autres langages de programmation, le développeur a juste besoin de traduire la logique de l'application dans le mappeur et le réducteur avec les éléments clés et la valeur de sortie.

Dans le schéma suivant, nous pouvons identifier les différentes composantes des jobs MapReduce de Hadoop streaming.

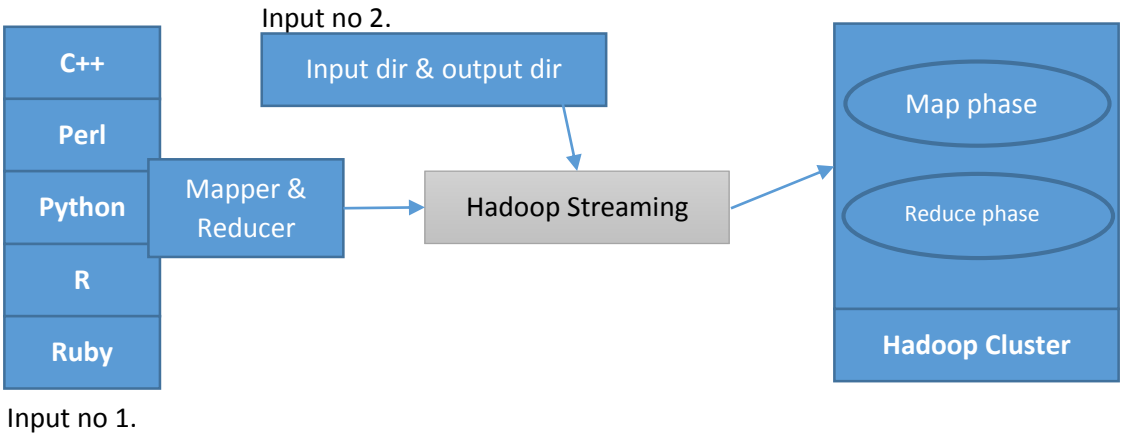


Figure 19. Composants de Hadoop streaming [26]

Conclusion

L'évolution des technologies utilisées durant le traitement des données est à noter, alors que les systèmes SQL traditionnel limitent l'utilisateur à un certain schéma précis, les nouveaux systèmes NoSQL donnent plus de liberté à l'utilisateur à plusieurs niveaux. De plus, la gestion de stockage, analyse et recherches est maintenant améliorée à l'aide de nouveaux outils comme Hadoop et son écosystème.

Conclusion générale et perspectives

Les Big Data peuvent être considéré comme une révolution dans le domaine de gestion des données. L'adaptation aux grands volumes, aux manques de structuration et aux besoins de vitesse de traitement est accomplissement plausible.

De nos jours, les données sont devenues de grande importance, plusieurs organisations se battent pour en rassembler le maximum afin de les interpréter et proposer de différents services utiles dans plusieurs domaines, transport, marketing, enseignement, réseaux sociaux, Santé…

L'intérêt que nous avons apporté au sujet des Big data ne vient pas du vide, ce dernier est un champ très prometteur qui engendre plusieurs champs d'application. Comme nous avons noté dans le premier chapitre, les Big Data touchent différents domaines, transport, marketing, enseignement, réseaux sociaux, santé… Malheureusement, l'analyse de ces données, à cause de leur volume, de leur hétérogénéité et de leur complexité est de plus en plus délicate. Chose qui s'est répercutée sur l'évolution des TI.

Ce phénomène a vu le jour dans le monde technique avant de voir les chercheurs scientifiques s'y intéressé. Afin de dégager des axes de recherche dans ce domaine, nous avons mené une étude bibliographique qui a aidé à développer quelques thématiques :

- Domaine médical :

Le but de l'informatisation de ces derniers est de pouvoir donner naissance à des diagnostiques correctes sans recourir aux méthodes d'examens traditionnels, qui sont parfois pas très équitables. En effet, plusieurs patients se plaignent de l'incompétence de certains médecins, du manque d'intérêt que leur apporte un médecin ou même de la difficulté qu'ils trouvent à avoir un rendez-vous pour être soigner.

Pour faire suite à cette étude, nous avons pensé à appliquer ces nouvelles méthodes de Big Data en domaine médical, en créant une architecture convenable et l'utiliser dans une application web qui va acquérir les différentes informations d'un patient à l'aide de questionnaires et les stocker entant que dossier médical de ce dernier qui va être analysé par des méthodes d'analyse de Big Data. Comme résultat de cette analyse nous allons avoir un ensemble de statistiques, de recommandations et de rapports, qui vont être classé selon la ressemblance entre les diagnostiques et les symptômes. Ces classements vont être à la portée de différents médecins de différents pays qui vont pouvoir aider à améliorer les diagnostiques.

Pour pouvoir réaliser cela, nous comptant utiliser les outils présentés précédemment, un outil de gestion de base de données NoSQL vu les grands volumes et de l'hétérogénéité des données médicales, le Framework HADOOP avec les différents composants de son écosystème, et enfin le langage R pour une meilleur interprétation des résultats.

Annexe 1

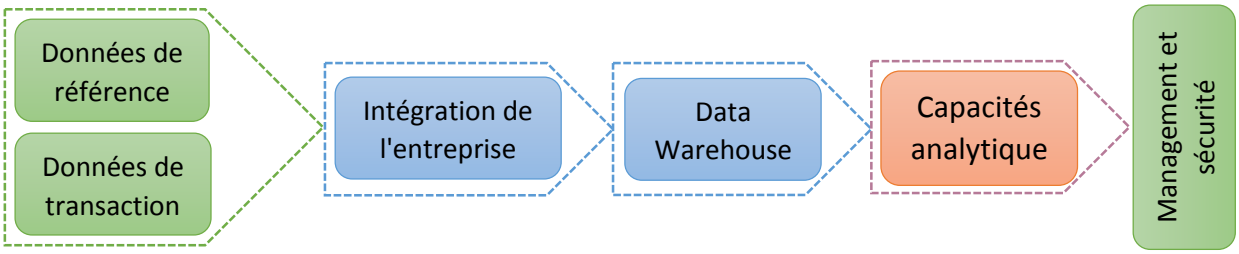


Figure 20. Architecture traditionnelle traitant les données structurées [12].

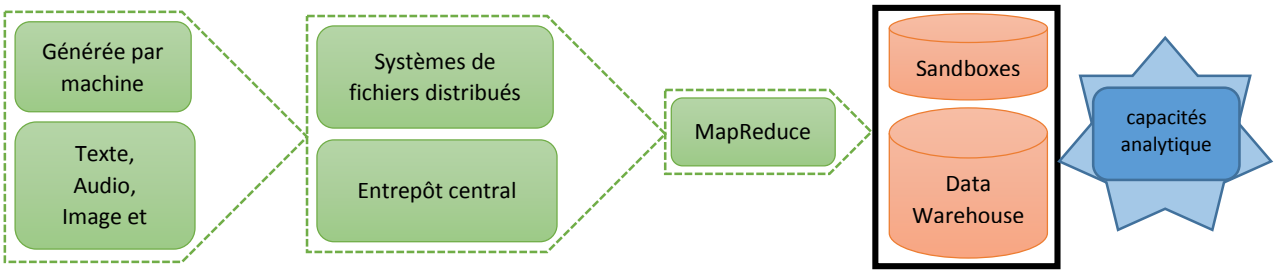


Figure 21. Architecture traditionnelle traitant les données non structurées [12].

Après le processus du mapping et de réduction effectués par le MapReduce, les résultats de l'étape de réduction s'intègrent dans les environnements des datawarehouse pour qu'on puisse avoir des rapports BI traditionnels, sémantiques, de corrélation et de capacités statistiques. En fin de compte il est intéressant d'avoir des capacités analytiques qui mélange les plateformes traditionnel BI avec les capacités des Big Data en visualisation des données [12].

L'approche proposée effectue une analyse sur les Frameworks Hadoop MapReduce et les systèmes de clusters distribués, comme NoSQL et le système de fichiers distribués de Hadoop (HDFS). Au total, le fichier interrogé pour l'analyse sont réalisé entant que problème de MapReduce à travers des Big Data non stucturées placées dans NoSQL et système de fichiers distribués de Hadoop (HDFS). D'après eux, cette approche, peut créer des Big Data à cout faible, de meilleurs résultats, hautement évolutives et une tolérance aux pannes. La figure 21 décrit l'approche de données signifiante proposée.

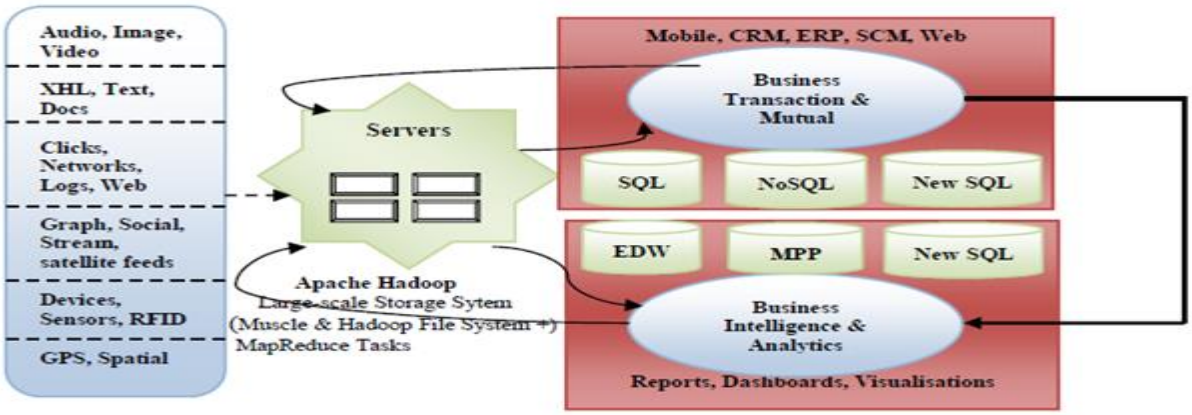


Figure 22. Architecture Big Data proposée [12]

Hadoop Apache agit entant que l'ensemble des serveurs Big Data. Il a de grandes capacités de stockage, combinaison, et traduction des données multi structurées en des dispositions utiles et de valeur. Par exemple, Hipe Apache est un élément associé à Hadoop qui correspond à l'intérieur des BI et des classes d'analyse puisqu'il est généralement utilisé pour l'interrogation et l'analyse des données dans hadoop en un mode SQL-like. Hadoop apache peut aussi être inclut avec les composants EDW, MPP et des composants NewSQL comme HP Vertica, Teradata, EMC Greenplum, Aster Data, IBM Netezza, SAP Hana et plusieurs d'autres technologies.

De plus, Hbase de Hadoop un entrepôt clé/valeur NoSQL qui est habituellement utilisé pour construir des applications de future génération extrêmement réactive.

Hadoop Apache peut aussi être inclut avec d'autres technologies SQL, NoSQL, et NewSQL comme MySQL, IBM DB2, PostgreSQL, Oracle, MongoDB, Terracotta,GemFire, SQLFire, VoltDB, serveur Microsoft SQL et plusieurs d'autres.

Ils ont conclu que les tendances de data l'intégration des sciences appliquées aide à assurer les flux de données entre les systèmes cité au-dessus.

Annexe 2

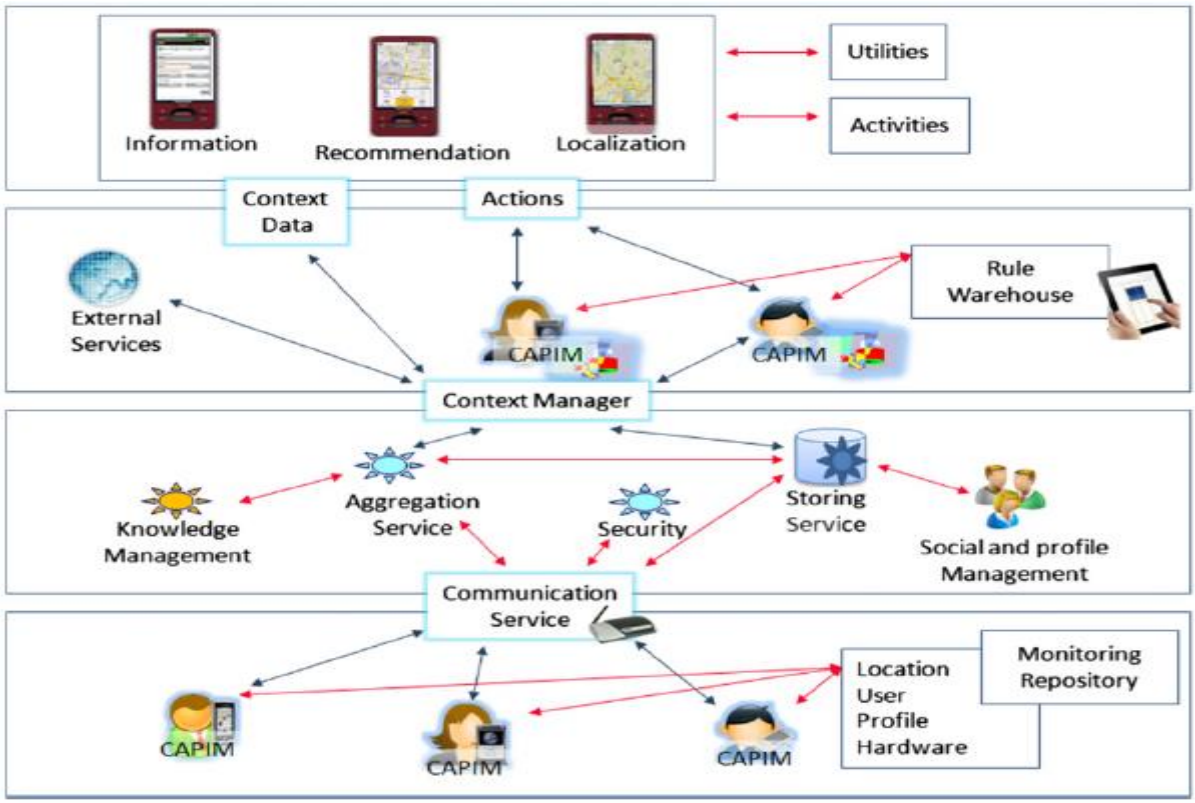


Figure 23. Architecture CAPIM [13]

Le flux de surveillance est sous le contrôle d'un gestionnaire de contexte (Figure 22), orchestrant le flux d'informations entre les services de surveillance. Selon la fonction en charge, les services de surveillance sont regroupés en plusieurs catégories.

Les services de surveillance Push and Pull sont directement responsables de la collecte des informations de contexte. Le service Push réagit aux changements de contexte, qui à son tour déclenche des notifications soient envoyées au gestionnaire de contexte.

Le gestionnaire de contexte transmet les données d'intérêt pour le service de filtre, qui à son tour peut produire de nouvelles informations de contexte (ce qui est possible à partir de plusieurs sources de données).

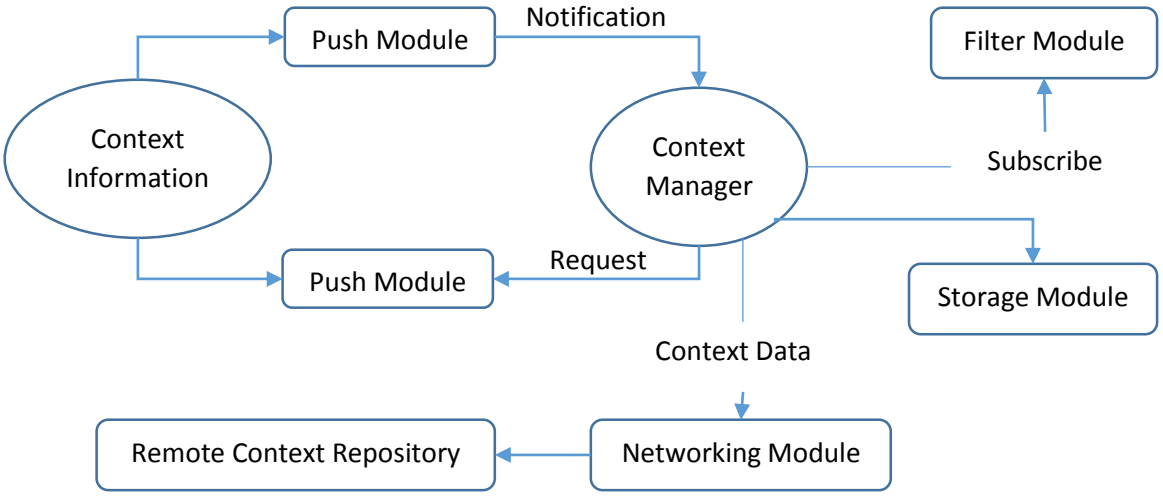


Figure 24. Flux d'information de surveillance [13]

Cette plateforme a rencontré plusieurs défis liés à l'efficacité de stockage des grands volumes de données détectées, et la nouvelle méthode présentée dans cet article a essayé de surmonter avec leur nouveau système de stockage intelligent.

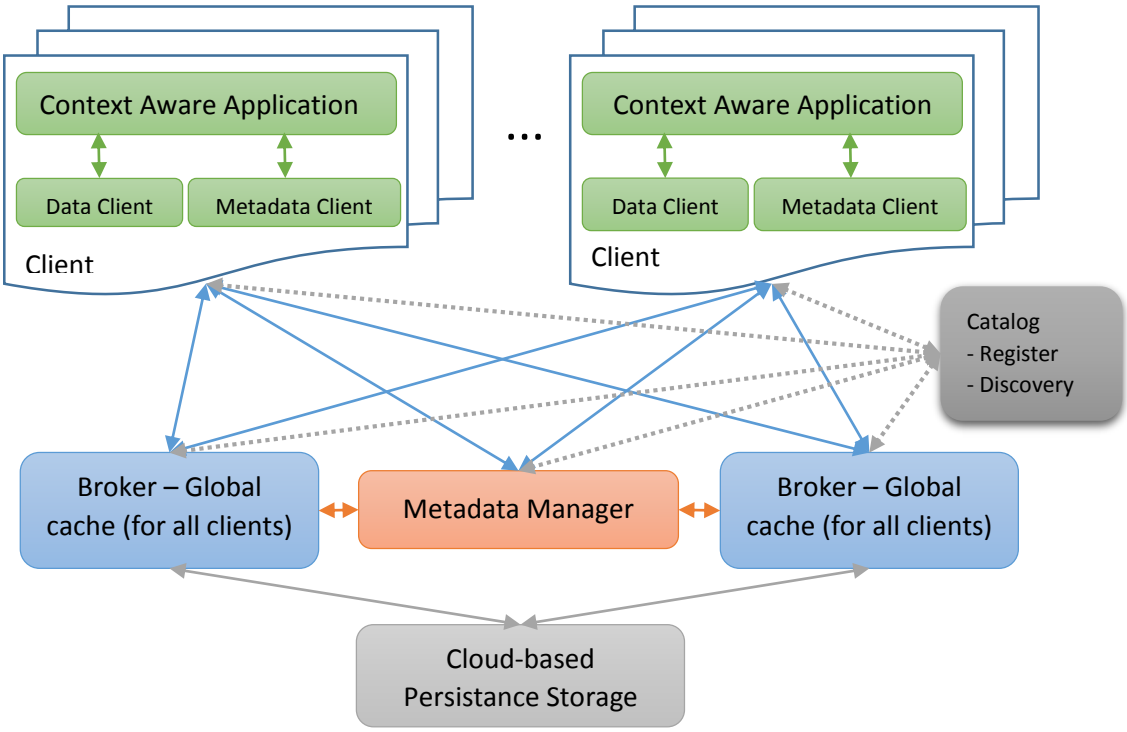


Figure 25. Architecture proposée [13]

Le ‘Metadata Client’ et les ‘Data Client’ connectent ‘Context Aware Application’ et les ‘Context Aware Framework’. Le ‘Metadata Client’ est chargé de créer et d'accéder à l'information sur les métadonnées qui décrit le schéma de données de contexte utilisé par une application particulière.

Le Client métadonnées est chargé de créer et d'accéder à l'information sur les métadonnées qui décrit le schéma de données de contexte utilisé par une application particulière.

Un client de données ‘Data client’ peut écrire des données, récupérer et stocker des données de contexte qui sont nécessaires à une application particulière sensible au contexte. Ici, ils supposent une relation one-to-one, chaque application étant servie par un client de données précis. Chaque Data Client est responsable de soutenir la mobilité de l'utilisateur, en soutenant un accès transparent au broker le plus proche.

Le Data Client fonctionne avec son propre cache local, utilisé pour les situations hors ligne, lorsque l'utilisateur ne peut pas se connecter au broker. Un service de découverte dédié est utilisé pour l'enregistrement et la découverte du gestionnaire de métadonnées existantes et des brokers. Dans l'architecture ils supposent l'existence d'un seul gestionnaire de métadonnées ‘Metadata Manager’, et de plusieurs brokers. Le service de découverte est aussi responsable de trouver le broker le plus commode pour un client particulier.

Le ‘Metadata Manager’ gère les connexions entre les méta-informations décrivant les données et les informations concernant le stockage de données physique réelle. Quand une nouvelle application sensible au contexte est inscrite pour la première fois, le data client se connecte au ‘Metadata Manager’ et écrit les méta-informations décrivant cette application particulière. L'information contient, entre autres, les formats de type de données pour décrire les données contextuelles collectées / stockées par l'application. Ensuite, lorsque le data client écrit les données de contexte, il se connecte au broker le plus proche. Les données de contexte est envoyé au broker, qui à son tour l'écrit dans la couche persistance.

L'architecture du Framework sensible au contexte apporte plusieurs fonctionnalités. Le Framework est conçu pour les applications sensibles au contexte qui travaillent avec des données représentées la plupart du temps en tant que série basée sur le temps, où les entrées sont sous la forme .timestamps, objet …

L'architecture prend en charge des applications évolutives. Une fois déployé, le système peut prendre en charge un grand nombre d'applications, impliquant potentiellement un grand nombre d'utilisateurs, chacun avec ses propres données de contexte. Ceci est parce que chaque application fonctionne dans un environnement distinct.

Le Framework sensible au contexte fournit une localité, Mobilité et des garanties d'accès en temps réel. Afin d'avoir un minimum en temps de réponse, un client se connecte au broker le plus proche avant de lancer une demande. Toutes les opérations de lecture sont mises en cache sur deux niveaux: l'un est sur le côté de Data Client et l'autre sur le côté du Broker. Si deux clients émettent la même demande, la réponse pour le second sera récupérée depuis le cache du Broker. Cela garantit aux deux clients un bon temps de réponse.

La Persistance est également supportée. Les clients écrivent leurs données, ce qui à son tour est enregistré dans le système de stockage. Plus tard, les clients peuvent demander des données, à travers des filtres de recherche complexes…

Annexe 3

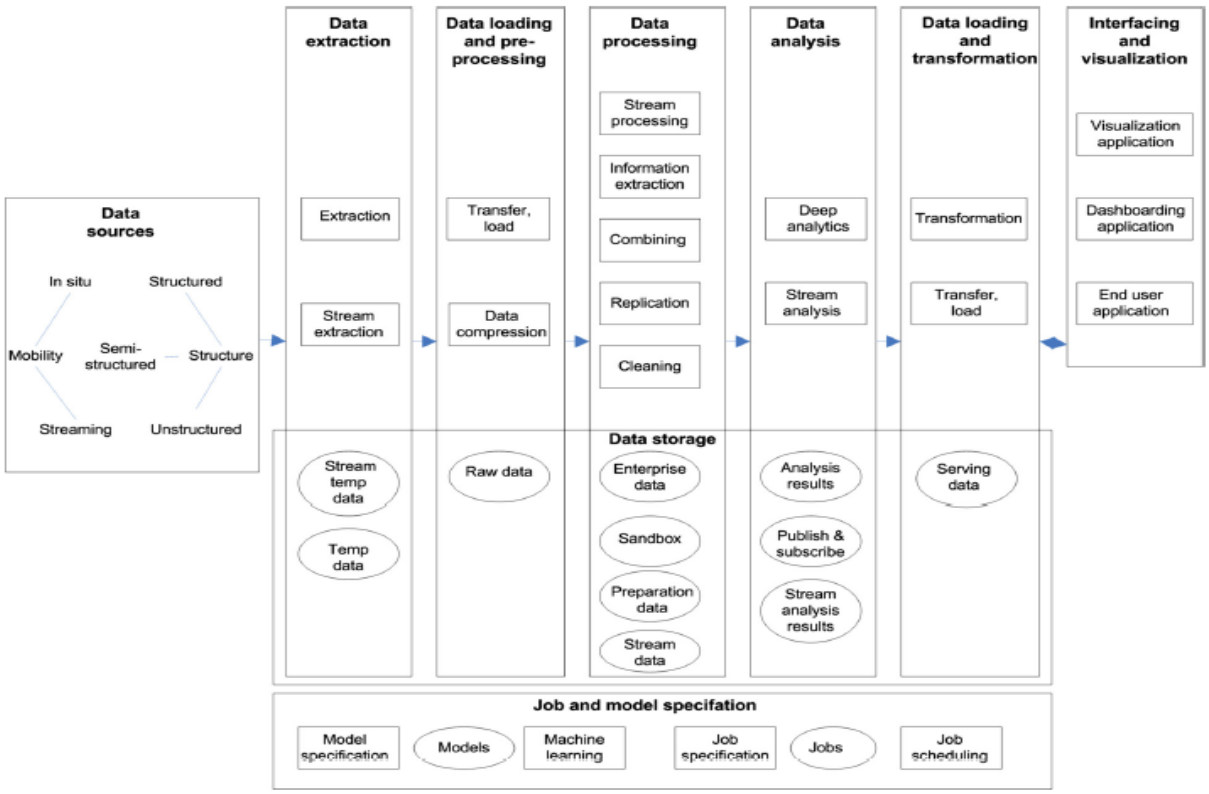


Figure 26. Conception haut niveau de l'architecture de référence [36]

- <u>Explication de l'architecture :</u>

Les fonctionnalités sont présentées en rectangle, les entrepôts de données en ellipse, les flux de données en flèche. Les fonctionnalités de traitement de données sont présentées en pipelines. Les fonctionnalités similaires sont regroupées dans les mêmes domaines. Les spécifications des jobs et modèles ont été illustré séparément des pipelines de data.

Pour la mobilité, les auteurs ont définis deux type, les data in situ et les data en streaming. In situ fait référence aux données qui ne doivent pas se déplacer. Les données en streaming font référence aux flux de données à traiter en temps réel.

Pour la structure des données, les données structurées, les données non structurées et les données semi-structurées. Le contenu des pages Web ou des images peuvent être considérées comme des données non structurées.

L'extraction réfère à l'entrée de données in situ dans le système. Lorsque des données in situ sont extraites, elles peuvent être stocké temporairement dans un entrepôt de données (temp data store) ou transféré, et chargé dans un entrepôt de données brutes.

Les données en streaming peuvent également être extraites et stockées temporairement (Streaming temp data store). L'efficacité peut être améliorée en utilisant la compression des données extraites avant les opérations de transfert et de chargement.

Le but de l'entrepôt de données brutes (Raw data store) est de contenir des données non traitées. Les données de l'entrepôt de données brutes peuvent être nettoyés ou combinés, et

enregistrées dans un nouvel entrepôt de données de préparation (preparation data store), qui détient temporairement les données traitées.

Le nettoyage et la combinaison référent à l'amélioration de la qualité des données brute, non transformés. Les données brutes et les données préparées peuvent être répliquées entre les entrepôts de données. En outre, de nouvelles informations peuvent être extraites à partir de l'entrepôt de données brutes (Raw data store) pour une analyse profonde.

L'extraction de l'information fait référence au stockage des données brutes dans un format structuré. L'entrepôt de données Enterprise (Entreprise data store) est utilisée pour le stockage des données nettoyées et traitées.

L'entrepôt Sandbox est utilisé pour contenir des données l'analyse de données à des fins expérimentales.

Les analyses profondes réfèrent à l'exécution des traitements par lots pour les données in situ. Les résultats de ces analyses peuvent être stockés de nouveau dans les entrepôts de données d'origine, dans un entrepôt d'analyse de résultats distinct (Analysis results store) ou dans un entrepôt de publication et souscription (Publish & subscribe store). L'entrepôt de publication et souscription (Publish & subscribe store) permet le stockage et la récupération des résultats d'analyse indirectement entre les utilisateurs et les éditeurs dans le système.

Le traitement en streaming (Stream processing) se rapporte au traitement des données extraites de streaming, qui peuvent être enregistrées temporairement avant être analysé.

L'Analyse en streaming (Steam analysis) réfère à l'analyse des données de streaming et sont sauvegardées dans l'entrepôt des résultats d'analyse en streaming (Stream analysis results) les résultats de l'analyse de données peuvent également être transférés en un entrepôt de données de service (serving data store), qui servent les interfaces et les applications de visualisation.

Les données analysées peuvent être visualisées de plusieurs manières :

- Application de Dashboarding (Dashboarding application) réfère à une interface utilisateur simple, où généralement des informations clés sont visualisées sans contrôle de l'utilisateur.
- L'application de visualisation (Visualization application) fournit des fonctions détaillées de visualisation et de contrôle, et est généralement réalisé avec un outil de Business Intelligence dans le domaine de l'entreprise.
- Application de l'utilisateur final (End user application) a un ensemble limité de fonctions de commande, et peut être réalisé sous forme d'une application mobile pour les utilisateurs finaux.

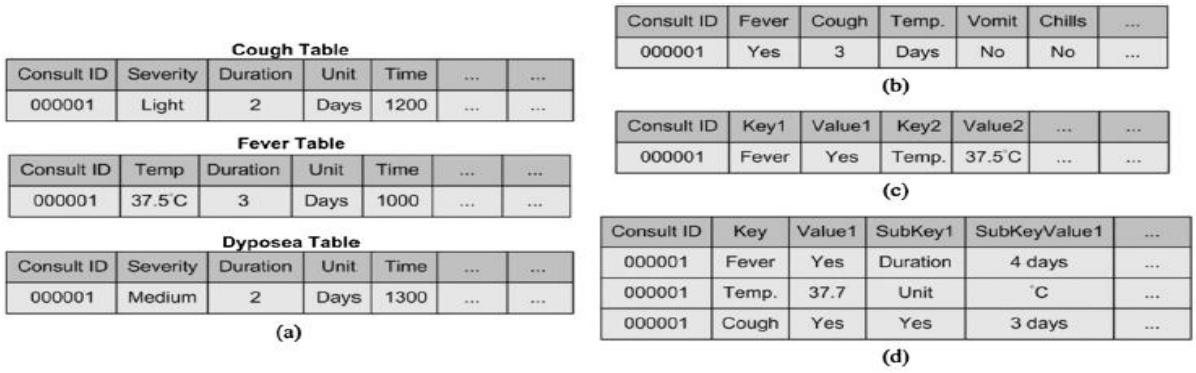
Les jobs de traitement par lots peuvent être spécifiés dans l'interface utilisateur. Les jobs peuvent être enregistrées et programmées avec des outils de planification des jobs. Les modèles/algorithmes peuvent aussi être spécifiées dans l'interface utilisateur (Model specification).

Annexe 4

Partie A :

De Différentes approches de conception de base de données sans schéma peuvent être utilisées pour mettre en œuvre les données cliniques. Ils comprennent, mais sans s'y limiter :

- Notion par table (basée sur les colonnes, plusieurs colonnes fixes) c'est un modèle de données simple. Comme le montre la Figure 26 (a), tous les concepts sont décomposés en tables individuelles. Cette approche est évidemment inadéquate, en raison de la nature inhérente du dossier médical qui pourrait impliquer une énorme quantité de concepts médicaux courants.
- Nom de colonne fixe (basée sur les colonnes, rangées non fixes, paire clé-valeur) cette approche est limitée à un champ d'application très spécifique d'un dossier médical. Tous les concepts médicaux sont prédéfinis dans une table de base de données sous forme de colonnes, et la capacité est limitée par le nombre maximal de colonnes souhaités par le système de base de données.
- Paire clé-valeur fixe et paire clé-valeur généralisée, Ce sont des techniques courantes dans la conception de base de données NoSQL.
  Les Bases de données mises en place dans l'approche paire clé-valeur fixe sont en mesure de fournir plus de flexibilité. Les valeurs sont stockées sous forme de paires clé-valeur, le rendant souple pour inclure autant de concepts que nécessaire. Cependant, le nombre de colonnes disponibles à partir d'une base de données peut facilement être épuisé par la quantité de concepts médicaux requis pour créer des notes cliniques.
  La flexibilité de bases de données développées avec l'approche paire clé-valeur généralisée est la meilleure entre les dernières approches. Dans cette approche, chaque concept est représenté comme une seule ligne dans la table de base de données, comme le montre la figure 26(d). L'inconvénient majeur, cependant, est que le nombre de lignes peut croître considérablement, même pour un petit nombre de dossiers. En outre, l'exécution de requêtes sur ces bases de données est très compliquée.



Cough Table

Consult ID	Severity	Duration	Unit	Time	...	...
000001	Light	2	Days	1200	...	...

Fever Table

Consult ID	Temp	Duration	Unit	Time	...	...
000001	37.5˚C	3	Days	1000	...	...

Dyposea Table

Consult ID	Severity	Duration	Unit	Time	...	...
000001	Medium	2	Days	1300	...	...

(a)

Consult ID	Fever	Cough	Temp.	Vomit	Chills	...
000001	Yes	3	Days	No	No	...

(b)

Consult ID	Key1	Value1	Key2	Value2	...	...
000001	Fever	Yes	Temp.	37.5˚C	...	...

(c)

Consult ID	Key	Value1	SubKey1	SubKeyValue1	...
000001	Fever	Yes	Duration	4 days	...
000001	Temp.	37.7	Unit	˚C	...
000001	Cough	Yes	Yes	3 days	...

(d)

Figure 27. Modélisation de base de données NoSQL: (a) Notion par table (trois concepts présentés), (b) Nom de colonne fixe, (c) paire clé-valeur fixe, et (d) paire clé-valeur généralisée [18]

Partie B :

La structure de données XML de base utilisé pour gérer les données cliniques structurés (figure 27) Au plus haut niveau on trouve l'élément "Record List" qui contient une liste de dossiers médicaux. Au niveau suivant on trouve les "Record" qui contiennent les éléments individuels des dossiers. Chaque dossier peut contenir des informations démographiques sur le patient comme le nom et le sexe, Au niveau plus bas on trouve le contenu des notes cliniques, à savoir une liste de concepts médicaux.

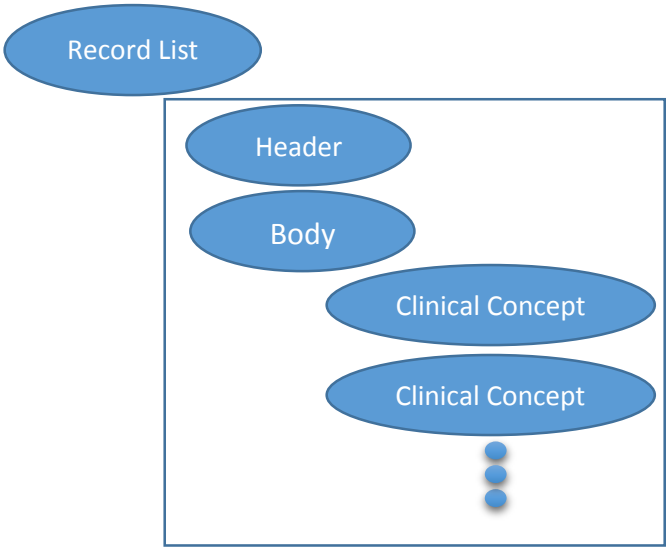


Figure 28. Structure de données XML [18]

Il y a deux implémentations de bases de données XML différentes pour le schéma XML décrit, à savoir, les bases de données compatibles XML et les bases de données XML native.

Consult ID	XML field	…
000001	<Data>	…
000002	<Data>	…
000003	<Data>	…

(a)

XML file list
Record000001.xml
Record000002.xml
Record000003.xml

(b)

Figure 29. (a) les données XML sont stockés avec un champ individuel dans la base de données compatible XML, (b) des fichiers XML contenant des données cliniques sont stockées sous forme d'une liste de fichiers dans la base de données XML native [18]

Annexe 5

- **Systèmes de recommandations en soins de santé :**

Dans cette partie, ils ont présenté trois systèmes de recommandation publiés :

- **Duan et all.** ont proposé un système de recommandation présentant un plan de soins infirmiers qui fournit un système d'aide à la décision, formation en soins infirmiers, et contrôle de qualité [20].

- **Hoens et all.** ont proposé un système de recommandation médical fiable à confidentialité préservée. Dans ce système médical, les patients peuvent contribuer avec leurs évaluations confidentielles sur les médecins à base de leurs satisfactions. Ce système peut recommander une liste de médecins qui convient le mieux aux besoins des patients [21].

- Wiesner et all. ont proposé un système de recommandation dans le contexte du système de dossier de santé personnel. Dans leur projet de système de Recommandation (Health Recommendation System : HRS), les articles sont non confidentiels, scientifiquement prouvé ou au moins des informations médicale acceptées. Le but de HRS est de fournir aux utilisateurs des informations médicales très pertinentes pour leur dossier de santé personnel [22].

- **Surveillance des épidémies sur internet**

Il existe plusieurs systèmes de surveillance des épidémies sur internet, mais le plus populaire est le 'Google Flu Trend'. Google fournit un outil appelé Google Flu Trends pour la surveillance en temps réel de grippe. Son hypothèse est que, lorsque le nombre de personnes présentant des symptômes de grippe, les recherches liés au sujets de grippe augmenteront. Par conséquent en se basant sur les recherches sur Internet, le nombre de personnes présentant des symptômes de grippe peut être estimé. Les prédictions faites par Google Flu Trends étaient 7-10 jours avant les réseaux officiels CDC et leurs résultats étaient cohérents.

Un autre géant du Web qui a lancé un outil de surveillance des épidémies en ligne est Twitter, Les publications sur Twitter sont appelées des tweets et ils reflètent les opinions et les jugements des gens au sujet de manifestation publique, en particulier les épidémies. Plusieurs méthodes ont été développées pour surveiller la réaction des gens à la flambée de l'épidémie et le syndrome de la maladie précoce, basé sur Twitter, qui aussi utilise les mots clés pour la détection des épidémies.

- **Capteur de surveillance de l'état de santé et du suivi de la sécurité alimentaire**

L'intégration de logiciels et de matériel, en particulier les différents capteurs, a créé des applications intéressantes qui surveillent l'état de santé et de sécurité alimentaire. Beaucoup d'entreprises de haute technologie ont lancé leurs produits, dont la plus connue est celle de Apple, la 'Apple Watch' qui est un boîtier bourré de capteurs enregistrant les mouvements (nombre de pas) et les paramètres vitaux au poignet du porteur : pression artérielle, le niveau d'hydratation, la fréquence cardiaque, des données comme la glycémie, cette montre peut être connectée avec plusieurs outils Apple où les données capturées et peuvent être analysé [23].

Annexe 6

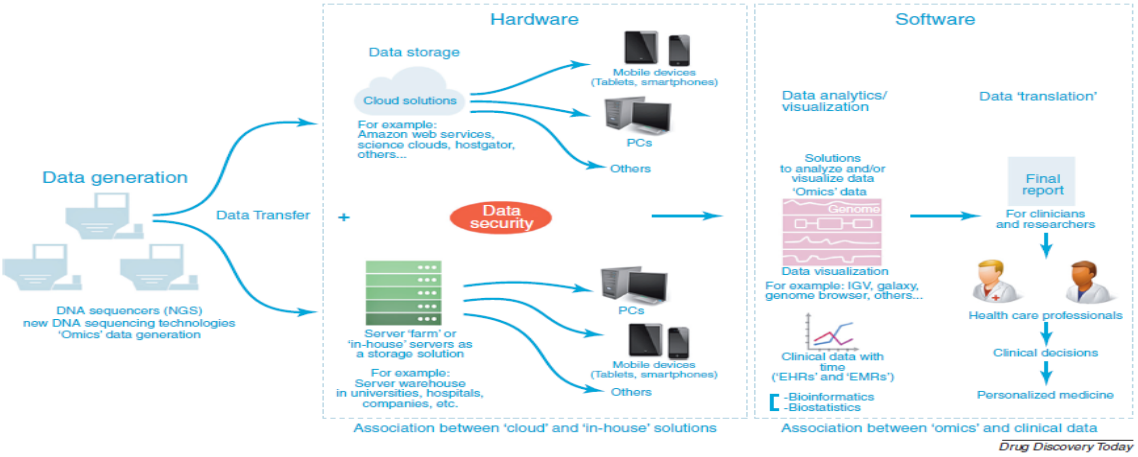


Figure 30. Pipeline d'analyse des Big Data en Biomédecine [24]

La figure ci-dessus est une représentation schématique d'un pipeline commençant à partir des données produites en utilisant le séquençage de prochaine génération (NGS), puis la «traduction» des données, et la génération d'un «rapport final» pour les cliniciens et les chercheurs.

Enfin, les données se traduisent par un bref rapport aux cliniciens et aux chercheurs après une analyse profonde de biomarqueurs et de cibles thérapeutiques liés à des phénotypes de maladies spécifiques et après des comparaisons avec les bases de données publiques ou privées sont effectuées.

Ce type de pipeline facilitera la mise en œuvre et l'application de la médecine personnalisée pour les cliniciens et à des fins de recherche. Entre le transfert de données, le stockage et la visualisation, les données du patient doivent être sécurisées par le cryptage d'information. Certaines solutions pour la sécurité des données médicales et scientifiques ont été développées récemment, mais puisque c'est un nouveau domaine d'étude en informatique biomédicale, de grands défis attendent la création de nouvelles possibilités sur le marché.

Dans cet article, ils ont aussi présenté des exemples d'entreprises et des institutions qui offrent des solutions pour générer, interpréter et visualiser les données omiques combinées et les données de santé clinique (table 1), des exemples de grandes entreprises qui offrent des solutions et des pipelines pour stocker, analyser et traiter les informations biomédicales complexes (table2), et des entreprises qui offrent une génétique personnalisées et des solutions omiques (table3).

Company or institution	Type of solution	Website
Appistry	High-performance big data platform that combines self-organizing computational storage with optimized and distributed high-performance computing to provide secure, HIPAA-complaint accurate on-demand analysis of omics data in association with clinical information	http://www.appistry.com
Beijing Genome Institute	This solution serves as a solid foundation for large-scale bioinformatics processing. The computing platform is an integrated service comprising versatile software and powerful hardware applied to life sciences	http://www.genomics.cn/en
CLC Bio	Utilizes proprietary algorithms, based on published methods, to accelerate successfully data calculations to achieve remarkable improvements in big data analytics	http://www.clcbio.com
Context Matters	Provides a comprehensive tool that empowers pharmaceutical and biotechnology companies to make better strategic decisions using web-based applications, and easy-to-use interface and visualization tools to deal with complex data sets	http://www.contextmattersinc.com
DNAnexus	Provides solutions for NGS by using cloud computing infrastructure with scalable systems and advanced bioinformatics in a web-based platform to solve data management and the challenges in analysis that are common in unified systems.	http://www.dnanexus.com
Genome International Corporation	Genome International Corporation (GIC) is a research-driven company that provides innovative bioinformatics products and custom research solutions for corporate, government, and academic laboratories in life sciences	http://www.genome.com
GNS Healthcare	A big data analytics company that has developed a scalable approach to deal with big data solutions that could be applied across the healthcare industry	http://www.gnshealthcare.com
NextBio	Big data technology that enables users to integrate and interpret systematically public and proprietary molecular data and clinical information from individual patients, population studies and model organisms applying omics data in useful ways both in research and in the clinic	http://www.nextbio.com
Pathfinder	Develops customized software applications, providing solutions in different sectors, including healthcare and omics, offering technologies that enable business breakthroughs and competitive advantages	http://www.pathfindersoftware.com

Table 1. Exemples de solutions proposées pour générer, interpréter et visualiser les données biomédicales [24]

Company	Solution(s)	Website
Amazon Web Services	Provides the necessary computing environment, including CPUs, storage, memory (RAM), networking, and operating system, for a hardware infrastructure as a service in the biomedical and scientific fields	http://aws.amazon.com
Cisco Healthcare Solutions	Offers different types of solution for the life sciences, including specific hardware and cloud computing for reliable and highly secure health data communication and sharing across the healthcare community	http://www.cisco.com/web/strategy/healthcare/index.html
DELL Healthcare Solutions	Connects researchers to the right technology and processes to create information-driven healthcare and accelerate innovation in life sciences with electronic medical record (EMR) solutions	http://www.dell.com/Learn/us/en/70/healthcare-solutions?c=us&l=en&s=hea
GE Healthcare Life Sciences	Provides expertise and tools for a wide range of applications, including basic research of cells and proteins, drug discovery research, as well as tools to support large-scale manufacturing of biopharmaceuticals	http://www3.gehealthcare.com/en/Global_Gateway
IBM Healthcare and Life Sciences	Provides healthcare solutions, technology and consulting that enable organizations to achieve greater efficiency within their operations, and to collaborate to improve outcomes and integrate with new partners for a more sustainable, personalized and patient-centric system	http://www-935.ibm.com/industries/healthcare
Intel Healthcare	Currently builds frameworks with governments, healthcare organizations, and technology innovators worldwide to build the health IT tools and services of tomorrow by combining different types of health information	http://www.intel.com/healthcare
Microsoft Life Sciences	Provides innovative, world-class technologies to help customers nurture innovation, improve decision-making and streamline operations	http://www.microsoft.com/health/en-us/solutions/Pages/life-sciences.aspx
Oracle Life Sciences	Delivers key functionalities built for pharmaceutical, biotechnology, clinical and medical device enterprises. Oracle maximizes the chances of discovering and bringing to market products that will help in treating specific diseases	http://www.oracle.com/us/industries/life-sciences/overview/index.html

Table 2. Exemples de grandes entreprises qui offrent des solutions et des pipelines pour stocker, analyser et traiter l'information biomédicale complexe [24].

Examples of companies that offer personalized genetics and omics solutions		
Company	**Applications and/or services**	**Website**
23andme	A DNA analysis service providing information and educational tools for individuals to learn and explore their DNA through personal genomics	http://www.23andme.com
Counsyl	Offers tests for gene mutations and variations in more than 100 inherited rare genetic disorders using a DNA biochip designed specifically to test for these disorders	http://www.counsyl.com
Foundation Medicine	A molecular information company at the forefront of bringing comprehensive cancer genomic analytics to routine clinical care	http://www.foundationmedicine.com
Knome	Analyzes whole-genome data using software-based tests to examine and compare simultaneously many genes, gene networks and genomes as well as integrate other forms of molecular and nonmolecular data	http://www.knome.com
Pathway Genomics	Incorporates customized and scientifically validated technologies to generate personalized reports, which address a variety of medical issues, including an individual's propensity to develop certain diseases	http://www.pathway.com
Personalis	A genome-scale diagnostics services company pioneering genome-guided medicine focused on producing the most accurate genetic sequence data from each sample, using data analytics and proprietary content to draw accurate and reliable biomedical interpretations	http://www.personalis.com

Table 3. Exemples de compagnies qui offrent la génétique personnalisées et des solutions omiques [24].

Pour conclure ils ont présenté les défis créés à cause des changements révolutionnaires dans génération de Big Data et l'acquisition de ces données créent des défis profonds pour le stockage, le transfert et la sécurité des informations. En effet, il peut maintenant être moins coûteux à produire des données qu'il en a pour les stocker, les sécuriser et les analyser. De plus, les données biologiques et médicales sont plus hétérogènes que d'informations de tout autre domaine de recherche.

Un autre défi consiste à transférer les données d'un endroit à un autre, parce que cela est principalement effectué par l'expédition des disques durs externes contenant les informations, ce qui n'est pas très pratique, pour eux une solution qui sera intéressante pour le transfert de données est l'utilisation de différents types de logiciels pour compresser les données sans perdre éléments d'information. La sécurité et la confidentialité des données des individus est également une préoccupation qui a besoin d'attention.

Bibliographie

1. Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*,*35*(2), 137-144.
4. Labrinidis, A., & Jagadish, H. V. (2012). Challenges and opportunities with big data. Proceedings of the VLDB Endowment, 5(12), 2032-2033.
5. Gonzalez, D. (2014). *Managing Online Risk: Apps, Mobile, and Social Media Security*. Butterworth-Heinemann.
6. Tufféry, S. (2005). *Data mining et statistique décisionnelle: l'intelligence dans les bases de données*. Editions Technip.
8. Pääkkönen, P., & Pakkala, D. (2015). Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems.*Big Data Research*.
9. Huang, T., Lan, L., Fang, X., An, P., Min, J., & Wang, F. (2015). Promises and Challenges of Big Data Computing in Health Sciences. *Big Data Research*,*2*(1), 2-11.
10. Blanke, T. (2014). *Digital Asset Ecosystems: Rethinking crowds and cloud*. Elsevier.
11. *Kshetri, N. (2014). Big data' s impact on privacy, security and consumer welfare. Telecommunications Policy, 38(11), 1134-1145. [12] Advancing Big Data for humanitarian needs*
13. Dobre, C., & Xhafa, F. (2014). Intelligent services for big data science. *Future Generation Computer Systems*, *37*, 267-281.
14. Jin, X., Wah, B. W., Cheng, X., & Wang, Y. (2015). Significance and challenges of big data research. *Big Data Research*, *2*(2), 59-64.
15. Sun, J., Xu, W., Ma, J., & Sun, J. (2015). Leverage RAF to find domain experts on research social network services: A big data analytics methodology with MapReduce framework. *International Journal of Production Economics*,*165*, 185-193.
16. Lafuente, G. (2015). The big data security challenge. *Network Security*,*2015*(1), 12-14.
17. Saraladevi, B., Pazhaniraja, N., Paul, P. V., Basha, M. S., & Dhavachelvan, P. (2015). Big Data and Hadoop-a Study in Security Perspective. *Procedia Computer Science*, *50*, 596-601.
18. Lee, K. K. Y., Tang, W. C., & Choi, K. S. (2013). Alternatives to relational database: comparison of NoSQL and XML approaches for clinical data storage.*Computer methods and programs in biomedicine*, *110*(1), 99-109.
19. Huang, T., Lan, L., Fang, X., An, P., Min, J., & Wang, F. (2015). Promises and Challenges of Big Data Computing in Health Sciences. *Big Data Research*,*2*(1), 2-11.
20. Duan, L., Street, W. N., & Xu, E. (2011). Healthcare information systems: data mining methods in the creation of a clinical recommender system. *Enterprise Information Systems*, *5*(2), 169-181.
21. Hoens, T. R., Blanton, M., Steele, A., & Chawla, N. V. (2013). Reliable medical recommendation systems with patient privacy. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *4*(4), 67.
22. Wiesner, M., & Pfeifer, D. (2014). Health recommender systems: Concepts, requirements, technical basics and challenges. *International journal of environmental research and public health*, *11*(3), 2580-2607.
24. Costa, F. F. (2014). Big data in biomedicine. *Drug discovery today*, *19*(4), 433-440.
25. Introduction aux systèmes NoSQL (Not Only SQL), Bernard ESPINASSE,Professeur à Aix-Marseille Université (AMU) Ecole Polytechnique Universitaire de Marseille
26. Prajapati, V. (2013). *Big data analytics with R and Hadoop*. Packt Publishing Ltd.
30. Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of "big data" on cloud computing: review and open research issues. *Information Systems*, *47*, 98-115.
36. Pääkkönen, P., & Pakkala, D. (2015). Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems.*Big Data Research*.

Webographie

2. *http://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/*
3. *http://fr.slideshare.net/BernardMarr/a-brief-history-of-big-data*
7. *http://harvardmagazine.com/2014/03/why-big-data-is-a-big-deal*
23. *http://www.sciencesetavenir.fr/sante/20140909.OBS8631/apple-iwatch-la-montre-connectee-au-service-de-la-sante.html*
27. *http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html*
28. *http://hortonworks.com/hadoop/oozie/*
29. *http://thebigdatablog.weebly.com/blog/the-hadoop-ecosystem-overview*
30. *http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html*
31. *http://hortonworks.com/hadoop/hcatalog/*
32. *http://hortonworks.com/hadoop/flume/*
33. *http://avro.apache.org/docs/1.7.7/*
34. *http://www.cloudera.com/content/cloudera/en/documentation/cdh4/v4-2-2/Hue-2-User-Guide/hue2.html*
35. *http://www.datadr.org/*