



Rapport de Stage de deuxième année

Déploiement d'Apache Hadoop Hive comme service cloud dans un environnement Dockerisé

Réalisé par :

YACHAOUI Ayman

EL MZOURI Hayat

Encadré par :

M. GUEDIRA Khalid

Remerciements

Ce rapport est le résultat d'une période d'immersion dans le milieu professionnel. En préambule, nous souhaitons adresser nos remerciements aux personnes que nous avons côtoyées durant cette période et qui ont ainsi contribué à l'élaboration de ce rapport.

Tout d'abord de grands remerciements à Monsieur Khalid GUEDIRA, notre tuteur d'entreprise, pour son aide et pour le temps qu'il a bien voulu nous consacrer.

Nous remercions également Monsieur Bouchaib BELHADJ et Monsieur Mehdi ABDALLAH qui ont su nous aiguiller dans le cadre de notre travail ; nous les remercions également pour leur réactivité.

Nous aimerions également remercier toutes les personnes de l'open space que nous avons rencontrées lors de la période de stage et plus particulièrement, Madame Asmaa Charraf pour ses connaissances de notre structure d'accueil et son amabilité.

Nous exprimons notre sincère gratitude à mademoiselle Angeline KONE de l'INSA Lyon qui a eu l'amabilité de nous fournir les données de test dérivées de son stage en SI chez Capgemini et de nous accorder le droit de reprendre sa synthèse de l'environnement Hadoop.

Nous sommes particulièrement fiers de notre formation au sein de l'école et reconnaissants envers l'ensemble du corps professoral de l'ENSIAS qui nous ont armés de connaissances solides et de réflexes de résolution de problèmes que les membres d'IB-Maroc ont salué.

Enfin, nous adressons nos sincères remerciements à nos parents, nos fratries et à tous nos proches et amis, qui nous ont encouragés, aidés et soutenus durant cette année de formation.

Que ceux-ci et ceux dont les noms n'y figurent pas, puissent trouver l'expression de leur encouragement dans ce travail.

I. Résumé

Notre stage d'été effectué au sein d'IB- Maroc durant environ 7 semaines s'inscrit dans le cadre d'un projet de recherche & développement et veille technologique portant sur les tendances et concepts technologiques Big Data. Les Big Data sont des méthodes et des technologies pour des environnements évolutifs, pour l'intégration, le stockage et l'analyse des données multi-structurées (structurées, semi structurées et non structurées). L'objectif de ce stage a consisté à monter en compétence sur les technologies Big Data afin d'élaborer une présentation commerciale de ces technologies, puis de faire des tests techniques des composants de l'écosystème Hadoop (framework Java libre, destiné à faciliter la création et le déploiement sur cluster d'applications distribuées et scalables) et réaliser un démonstrateur Big Data illustrant des cas d'utilisation du Big Data avant d'implémenter une instance Amazon EC2 exposant le service Hadoop Hive sous forme de conteneur Docker générique.

Ce rapport a pour but de donner un aperçu du stage que nous avons effectué au sein d'IB-Maroc. On y trouve, dans une première partie, une présentation de l'entreprise permettant de dégager des informations sur l'environnement du travail au sein de cette société. Dans la seconde partie, avant de présenter les travaux effectués, la mission du stage et l'environnement technique de travail sont présentés. Enfin la dernière partie est consacrée au bilan du stage et à la conclusion.

Contenu

I. Résumé.....	4
II. Introduction générale	8
III. Présentation de l'entreprise et du contexte de stage	9
III.1 Présentation de L'entreprise.....	9
III.1.1 Historique et secteur d'activité.....	9
III.1.2 Valeurs	10
III.1.3 Métiers et secteurs d'activité	10
III.1.4 Organisation.....	11
III.2 Contexte du stage	12
IV. Présentation de la mission lors du stage	13
IV.1 Collecte et analyse de la documentation	13
IV.2 Tests techniques et montée en compétence opérationnelles	13
IV.3 Réalisation de la plateforme de démonstration.....	14
IV.4 Création de l'image Docker à même d'exposer Apache Hive comme Cloud service.....	14
V. Les Big Data.....	14
V.1 Présentation	14
V.2 Caractéristiques des Big Data	14
V.3 Les Big Data en chiffres.....	15
V.4 Intérêts des Big Data.....	15
V.5 Paysage technologique des Big Data	15
V.5.1 Intégration	16
V.5.2 Stockage.....	16
V.5.3 Analyse.....	17
V.5.4 Restitution	17
V.6 Ecosystème Hadoop	17
V.6.1 Hadoop kernel	18
V.6.2 Composants Apache Hadoop.....	22
V.7 Solutions Big Data sur le marché	24
VI. Solution mise en place.....	25
VI.1 Choix de la solution et tests techniques réalisés.....	25

VI.1.1 Architecture du cluster mise en place	25
VI.1.2 Architectures des composants des machines.....	26
VI.1.3 Déploiement de la plateforme.....	27
VI.1.4 Tests techniques réalisés	28
VI.2 Démonstrateur Big Data	30
VI.2.1 Les jeux de données de test.....	30
VI.2.2 Scénario d'exécution des cas d'utilisation	32
VI.2.3 Restitution des données via Hive	32
VI.2.4 Restitution des données via QlikView	33
VI.3 Outils utilisés.....	35
VII. Déploiement sur Cloud	35
VII.1 Qu'est ce que docker.	35
VII.2 Quelle différence avec la virtualisation ?	36
VII.3 Les principaux avantages de Docker comparé à la virtualisation ?	36
VII.4 Construction de l'image du conteneur : les Dockerfiles.....	36
VII.4.1 Les Dockerfiles	37
VII.4.2 Construction de l'image des conteneurs	37
VIII. Conclusion.....	38

Liste des abréviations

ESN : Entreprise de Services du Numérique

ERP : Enterprise Resource Planning (Progiciel de gestion intégré)

RFID : Radio Frequency Identification (radio identification)

ETL : Extract, Transform and Load

EAI : Enterprise Application Integration (Intégration d'applications d'entreprise)

EII : Enterprise Information Integration

CRM : Customer Relationship Management (gestion des relations avec les clients)

DSI : Direction des Systèmes d'Information

SQL : Structured Query Language

HDFS : Hadoop Distributed File System

SPOF : Single Point Of Failure

UDF : User Defined Function

UDAF : User Defined Aggregate Function

UDTF : User Defined Table Function

JDBC : Java DataBase Connectivity

ODBC : Open Database Connectivity

CDH : Cloudera's Distribution including apache Hadoop

SI : Système d'Information

R&D : Recherche et Développement

II. Introduction générale

Vers la fin de l'année 2003, 5 Exaoctets (1 exaoctet (Eo) = 1018 octets) de données ont été générées. Avec l'avènement du Web 2.0, la prolifération des réseaux sociaux, et la multiplication des applications mobiles stockant sur serveur ; cette quantité de données était générée chaque 2 jours. Actuellement, il suffit de 10 minutes pour en produire autant. D'après les prévisions, d'ici 2020 cette croissance sera supérieure à 40 Zettaoctets (1 zettaoctet (Zo) = 1021 octets) et le volume des données produites, diffusées et consommées en l'espace d'une année, doublera tous les 2 ans.

Cette croissance exponentielle ne pose pas simplement des problématiques quant au stockage de celles-ci mais –le but ultime derrière tout stockage- le recyclage de ces données dans des processus visant à valoriser les informations « cachées » au sein de cette foule de data. Tout l'enjeu, pour les entreprises et les administrations, consiste à savoir reconnaître la valeur ajoutée potentielle au sein des données engendrée par ces processus internes et celles disponibles dans les dépôts de data publique. C'est là qu'intervient la technologie du "Big Data", qui repose sur une analyse très fine de masses de données.

Sans les outils, les technologies à même d'apporter des solutions durables, fiables et exhaustives à ce challenge, le manque à gagner entre information potentiellement utilisable, et effectivement valorisée ne peut que s'accroître. Comme l'énergie atomique, avant sa découverte, à l'heure actuelle, les technologies du Big Data nous apprennent à sortir l'information en sommeil dans l'énorme potentiel économique enfoui dans des milliards d'enregistrements superflus.

Les ESN les plus attentives aux signes de la prochaine révolution numérique et qui entendent en tirer profit au moyen de leur expertise, s'engagent déjà dans une veille technologique continue et encouragent des projets de prototypes de solution Big Data. C'est dans ce contexte que notre stage s'inscrit.

Dès nos premiers jours au sein d'IB-Maroc, l'objectif était bien défini : En premier lieu, monter en compétences quant aux innovations et techniques de traitement massif de données et en sortir avec des livrables utiles aux présentations clients ainsi qu'aux formations internes ; pour ensuite implémenter une plateforme de démonstration pratique mettant en application les connaissances théoriques acquises et de réaliser un prototype fonctionnel d'un service Big Data de stockage et d'interrogation de base de données.

III. Présentation de l'entreprise et du contexte de stage

III.1 Présentation de L'entreprise

III.1.1 Historique et secteur d'activité

1997: Création d'IB Sud filiale marocaine d'IB Group, société holding de droit français à la tête de plusieurs sociétés opérant dans l'intégration globale des infrastructures informatiques (IB Solution) et l'achat et vente de matériels informatiques (IB Remarketing et Red Systems).

1999: Acquisition de DIGITEM par IB Sud, étoffant ainsi la présence d'IB Group auprès des grands comptes et apportant une nouvelle activité dans le domaine des services (maintenance et intégration), fondamentale pour le développement du groupe au Maroc. La société s'appellera DIGITEM GROUPE IB et deviendra par la suite IB Maroc.

2001: IB Maroc est introduite en bourse, elle demeure la première société marocaine cotée, dans le secteur des nouvelles technologies.

2003: IB Maroc lance sur le marché marocain IB formation, structure spécialisée dans la formation informatique high-tech.

2004: IB Maroc se lance dans le domaine de la télé administration.

2006: Le 10 août 2006, MIBS Infrastructure et Services et IB Group ont cédé leurs participations, qui s'élèvent à 53,96% du capital, dans IB Maroc à Sophia Invest Holding. Du 30 octobre au 03 novembre 2006, et suite à la détention par Sophia Invest de 53,96% du capital et 64,21% des droits de vote d'IB Maroc, une offre publique d'achat a été initiée et lancée par sophia Invest. En date du 17 novembre 2006, Sophia Invest Holding a pu acquérir 11.604 actions suite à l'offre publique d'achat lancée entre le 30 octobre 2006 et le 03 novembre 2006. Le nombre total des actions détenues par Sophia Invest Holding a ainsi été porté à 236.878 actions, représentant 56,74% du capital social d'IB Maroc.



Figure 1 : Implantation d'IB-Maroc dans le monde

III.1.2 Valeurs

Capital humain.

La notion de Capital Humain apparaît au premier rang dans les facteurs clés de succès au sein d'IB Maroc. Grâce à la connaissance rare et l'expertise du capital humain individuel ainsi que la synergie de du capital humain collectif, le groupe a instauré une culture spécifique focalisée sur l'investissement personnel, l'évolution et l'adaptation rapide aux changements du marché.

Prestation d'excellence.

IB-Maroc ont toujours été précurseurs dans l'introduction des dernières technologies sur leurs marchés. Leur métier ne s'improvise pas, pour cela ils appliquent une seule recette pour la réussite des projets : une conjugaison entre la mise en œuvre selon les meilleures pratiques et l'appropriation des usages clients.

Partenaire stratégique de confiance.

La stratégie IB-Maroc est centrée sur le partenariat de confiance sur le long terme avec leurs clients et les leaders technologiques mondiaux afin de mettre à disposition les meilleures solutions.

III.1.3 Métiers et secteurs d'activité

Consulting.

IB-Maroc met à disposition de ses clients son expertise et son savoir-faire pour les accompagner dans la transformation de leurs systèmes d'information tout en optimisant

leurs infrastructures et services applicatifs. IB-Maroc plaide qu'il n'existe pas de solution universelle, et s'efforcent à la place de cultiver une parfaite connaissance des différentes technologies disponibles, des solutions éprouvées à l'état de l'art.

Data center et cloud.

IB-MAROC a développé une méthodologie afin que les DSI appréhendent les différentes options d'une infrastructure Cloud Privé, l'adéquation avec leurs besoins opérationnels et métiers et la définition d'une stratégie pour tirer profit du modèle cloud.

ERP et BI.

Les entreprises et organisations ont besoin de solutions pour capter des informations, piloter, comprendre, interpréter, prévoir et décider. A tous les niveaux il est nécessaire que les décisions soient rapides, adaptées et cohérentes. Ces informations constituent le premier capital en matière de processus de décision. IB-Maroc prête toute son attention à ce segment du marché.

Mobilité et Apps.

La mobilité est sur le point de transformer radicalement le paysage des technologies de l'information et de la communication dans l'entreprise. Une (r)évolution qui ouvre de réelles opportunités, mais pose aussi des questions de sécurité. IB-Maroc accompagne ses clients pour leur faire bénéficier des premiers enseignements émergent pour mieux gérer les risques et les coûts associés à ce défi organisationnel.

Sécurité.

IB-Maroc propose des prestations de services pour sécuriser les données des Systèmes d'Information des organisations. Le périmètre de sécurité peut couvrir l'intégralité du SI : Sécurité d'accès, échanges et réseaux, accès distants, VPN, solutions anti-spam et anti-malwares, études et audits réseaux, etc.

Formation.

IB formation a développé un savoir-faire et une expérience incomparables, au service du client. Plusieurs organisations s'appuient sur IB formation pour les aider à identifier les compétences dont elles ont besoin, afin de supporter les systèmes et technologies qu'elles utilisent au quotidien.

III.1.4 Organisation

Le contrôle de la gestion de la société est actuellement confié à un Conseil d'Administration composé de trois membres choisis parmi les actionnaires. Ils sont nommés et révocables par l'Assemblée Générale Ordinaire des actionnaires et sont toujours rééligibles.

Suite à la cession des participations d'IB Group et MIBS à Sophia Invest Holding, celle-ci a été nommée membre du Conseil d'administration, représentée par M. Abdellatif HADEF, en tant qu'Associé Gérant, en remplacement de M. Loïc VILLERS. M. Loïc VILLERS était le Président Directeur Général de IB Group et de MIBS Infrastructure et Services. M. Abdellatif HADEF est actuellement Président Directeur Général de IB Maroc.com et de ses deux filiales, IB Libya détenue à hauteur de 51% et IB Continuité détenue à hauteur de 82,2%.

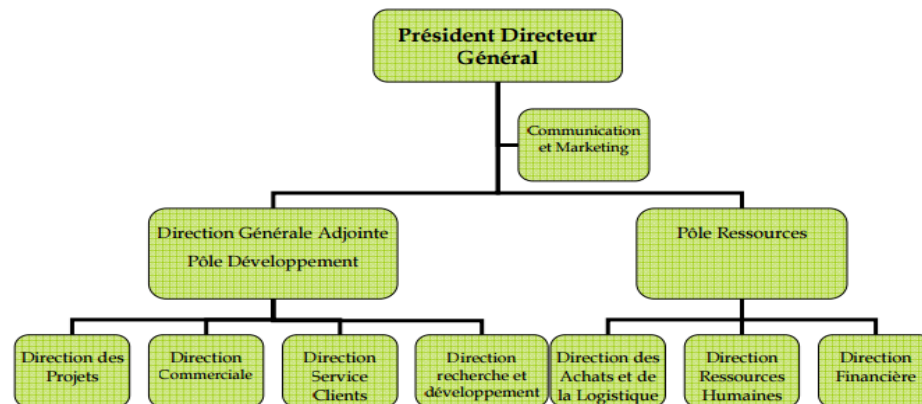


Figure 2 : Organigramme d'IB-Maroc

III.2 Contexte du stage

L'outil informatique apporte foule de solution pour répondre à un besoin grandissant des entreprises à assainir leurs processus métiers des déchets d'activité et impureté liée aux activités de support dénuées de l'agrément technologique convenable. Au début du siècle, Frederick Taylor révolutionnait déjà l'espace de travail par ses idées sur l'organisation des ateliers, la segmentation des tâches, et l'évaluation des performances des travailleurs. Son objectif était d'augmenter la productivité des entreprises d'une ère qui appelait à penser différemment les processus métiers. A l'aube de ce 21^e siècle, deux nouveaux outils sont en train de révolutionner le monde du travail comme le Taylorisme l'a fait jadis. Nous entendons par là les nouvelles technologies de l'information et de la communication - les possibilités offertes par les ordinateurs, les outils logiciels et les moyens de télécommunication - et les méthodes de re-design des "business process" - l'analyse et la modélisation des flux d'activités au sein des organisations. Il faut prendre de l'avance afin d'élaborer les solutions innovantes pouvant répondre aux besoins sans cesse croissants de la clientèle, dans sa quête perpétuelle de la performance. IB-Maroc, en tant qu'un des leaders nationaux et africains dans le domaine des services du numérique, anticipe sur les éventuels besoins clients en mettant en œuvre des technologies innovantes. C'est ainsi que les nouvelles solutions passent par une étape de veille technologique où on s'informe sur les technologies les plus récentes et la mise en œuvre commerciale. Les Big Data font

donc partie des technologies récentes en cours de gestation.

IV. Présentation de la mission lors du stage

La mission du stage a débuté le 17 Juillet 2015 pour prendre fin le 28 Aout 2015. Celle-ci consistait en la réalisation des objectifs suivants :

- Réaliser un travail de R&D et veille technologique des tendances et concepts technologiques Big Data.
- Les documents techniques des composants déployés et des tests techniques réalisés.
- La réalisation d'une plateforme de démonstration, mettant en œuvre des cas d'utilisation du Big Data.
- L'implémentation d'un prototype sous forme de conteneur Docker prêt à être déployé sur une instance Amazon EC2.

La veille technologique consiste à garder un fil d'Ariane ancré aux dernières innovations, idées et implémentations des solutions techniques les plus récentes et surtout de leur potentiel commercial.

La production de nos livrables est passée dans sa préparation par les étapes suivantes :

IV.1 Collecte et analyse de la documentation

Cette étape a été l'une des expériences les plus enrichissantes durant notre stage. Aiguillés par nos encadrants externes, nous digérions le maximum d'articles universitaires, de pages de blog techniques sur le net et de livres de référence quand la clarté et la précision étaient de mise. Nous devions – une fois un premier tout d'horizon des technologies et pratiques Big Data – focaliser notre attention plus précisément sur l'écosystème Hadoop qui s'avérait être de plus en plus le candidat idéal pour le socle de la réalisation des tests techniques. Une attention particulière de notre part a été dédiée aux tests réalisés et aux critiques quant aux versions des produits que nous passions en revue. La documentation et l'analyse ont donc suivi un cycle récursif définissant au fur et à mesure nos priorités quant aux prochains points d'intérêt.

IV.2 Tests techniques et montée en compétence opérationnelles

Durant cette étape, une configuration en local utilisant les machines d'IB-Maroc allait implémenter la solution de stockage et de requête sur des données de nature Big Data pour tester, la fiabilité de notre configuration mais aussi prévoir les contraintes physiques pour garantir la réussite de la plateforme de démonstration qui allait être le

prochain nœud dans le cheminement de notre mission.

IV.3 Réalisation de la plateforme de démonstration

Cette étape – que nous avons, sous le stress du moment, appelée le moment de vérité – allait être la validation de la pertinence de la solution que nous avons choisi ainsi que le premier résultat concret de notre travail aux côtés de l'équipe d'IB-Maroc.

IV.4 Création de l'image Docker à même d'exposer Apache Hive comme Cloud service

Forts et confiants des résultats en local, la prochaine étape était la conception de l'image du conteneur virtuel Docker capable d'exposer le service Hive prototype en cloud chez Amazon. Nous parlerons en plus amples détails de cette technologie de virtualisation et des atouts qui nous ont incités à la choisir pour notre déploiement.

V. Les Big Data

V.1 Présentation

Big Data est un terme qui décrit le grand volume de données - à la fois structurées et non structurées - qui inonde une entreprise jour après jour. Mais il serait très réducteur de limiter le concept au seul volume de données qui est important. Big Data c'est aussi et surtout comment les organisations traitent cette masse – souvent non structurée – de données. Une fois analysées, ces données donnent vie à des idées qui mènent à de meilleures décisions et motivent des choix business plus intelligents.

V.2 Caractéristiques des Big Data

Bien que le terme "Big Data" est relativement nouveau, l'acte de recueillir et de stocker de grandes quantités d'informations pour l'analyse éventuelle ne l'est pas pour autant. Le concept pris de l'ampleur dans le début des années 2000 quand l'analyste industriel Doug Laney a articulé la définition désormais populaire du Big Data fondée sur les 3 Vs:

- **Volume.** Les Organisations collectent des données à partir d'une variété de sources, y compris les transactions commerciales, les médias sociaux et les informations provenant de capteur ou des données d'interfaces machine-machine. Dans le passé, le stockage constituait un problème - mais les nouvelles technologies (telles que les systèmes de fichiers distribués inspirés de Google BigTable) ont allégé le fardeau.
- **Vitesse.** Les flux de données connaissent des débits de plus en plus croissants et doivent être traités en temps synchrone. Les étiquettes RFID, capteurs et

compteurs intelligents motivent la nécessité de digérer des torrents de données en temps quasi-réel.

- Variété. Les données proviennent de tous les types de formats - à partir de données numériques structurées, des bases de données traditionnelles, de documents non structurés de texte, email, vidéo, audio, données de transactions financières, etc.

A ces « 3V », s'ajoutent les challenges de visualisation et de « vulgarisation » managériale pour les premiers bénéficiaires de la valeur ajoutée du Big Data : les managers potentiellement néophytes aux sciences du traitement informatique.

V.3 Les Big Data en chiffres

- ✓ Nous créons chaque jour 2.500.000.000.000.000.000 (2.5 Quintillions) d'octets de données. De quoi remplir 10 millions de disques Blu-Ray.
- ✓ 118 milliards d'emails sont envoyés chaque jour.
- ✓ Les données stockées augmentent 4x plus vite que l'économie mondiale.
- ✓ Twitter génère 7 téraoctets de données par jour.
- ✓ Facebook génère 500 téraoctets de données par jour.

Nous nageons dans la donnée, et le niveau de la mer monte très rapidement !

V.4 Intérêts des Big Data

L'importance du Big Data ne tourne pas autour de la quantité de données qu'une entreprise possède, mais de ce qu'elle en fait. Des données de toute source peut être analysée pour trouver des réponses qui permettent 1) des réductions de coûts, 2) des gains en temps, 3) le développement de nouveaux produits et d'offres personnalisées, et 4) la prise de décision intelligente. Lorsque qu'on les grandes données avec l'analyse à forte puissance de calcul, l'entreprise peut:

- Déterminer les causes profondes des échecs, des problèmes et des défauts en temps quasi-réel.
- Génération de coupons aux points de vente basés sur les habitudes d'achat des clients.
- Recalculer le risque total de portefeuilles financiers en quelques minutes.
- Détecter les comportements frauduleux avant qu'ils n'affectent l'organisation.

V.5 Paysage technologique des Big Data

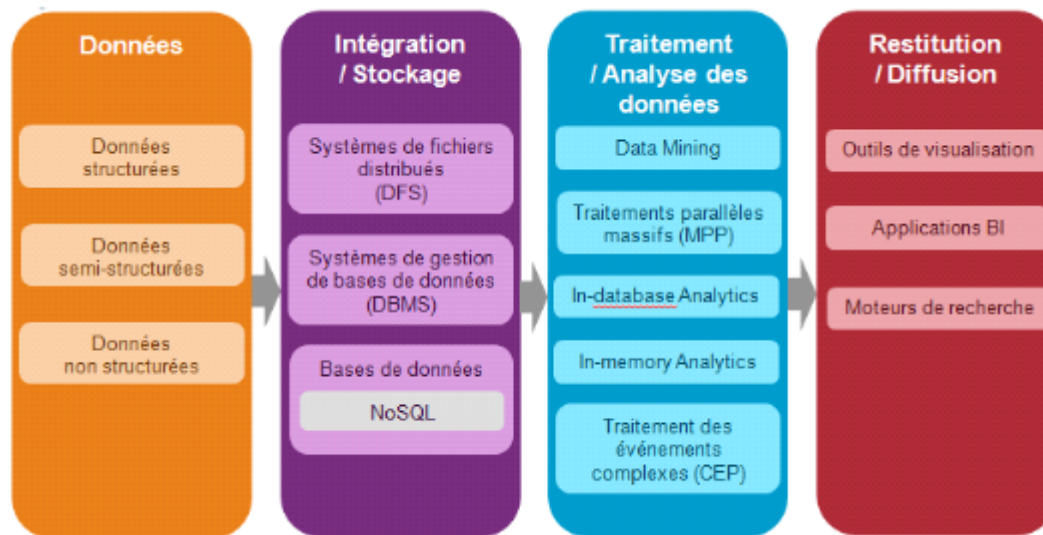


Figure 3 : Paysage technologique Big Data, (Bermond, 2013)

Le Big Data repose sur plusieurs technologies, qui sont utilisées pour exploiter les gigantesques masses de données.

V.5.1 Intégration

Dans le contexte Big Data, l'intégration des données s'est étendue à des données non structurées (données des capteurs, journaux Web, réseaux sociaux, documents). Hadoop utilise le scripting via MapReduce ; Sqoop et Flume participent également à l'intégration des données non structurées. Ainsi, certains outils d'intégration comprenant un adaptateur Big Data existe déjà sur le marché ; c'est le cas de Talend Enterprise Data Integration - Big Data Edition. Pour intégrer des gros volumes de données issus de briques fondatrices du système d'information des entreprises (ERP, CRM, Supply Chain (gestion de la chaîne logistique)), les ETL, les EAI, les EII sont toujours utilisés.

V.5.2 Stockage

Le premier élément structurant dans le contexte Big Data est le socle de stockage des données. Anciennement, la solution était les DataWarehouse (entrepôts de données), qui ont évolué pour supporter de plus grandes quantités de données et faire porter par le stockage, une capacité de traitement étendue. Les solutions de DatawareHouse ont toutes en commun un modèle de données profondément structuré (schéma, base de données, tables, types, vues, etc.) et un langage de requête SQL.

Le Big Data vient rompre cette approche ; l'approche du Big Data consiste en 2 grands principes.

Premièrement, le principe de la scalabilité (horizontale) des clusters de traitement. Puis

deuxièmement, on peut s'affranchir de certaines contraintes inhérentes aux bases de données relationnelles traditionnelles et qui ne sont pas forcément nécessaires pour le Big Data. C'est le cas de l'ACIDité (Atomicité, Cohérence, Isolation et Durabilité).

Pour mettre en œuvre cette approche avec une infrastructure simple, scalable (mot utilisé pour indiquer à quel point un système matériel ou logiciel parvient à répondre à une demande grandissante de la part des utilisateurs, il traduit aussi la capacité de montée en charge), du matériel à bas coût, le framework Hadoop est utilisé pour la gestion du cluster, l'organisation et la manière de développer. La solution la plus emblématique de cette approche est Hadoop et son écosystème.

V.5.3 Analyse

C'est bien de pouvoir stocker les données, mais faut-il pouvoir également les rechercher, les retrouver et les exploiter : c'est l'analyse des données. Elle est naturellement l'autre volet majeur du paysage technologique du Big Data. En la matière, une technologie qui s'impose, Hadoop. Ce projet applicatif fait l'unanimité même s'il est loin d'être stable et mature dans ses développements. Hadoop utilise MapReduce pour le traitement distribué.

MapReduce est un patron d'architecture de développement informatique introduit par Google qui permet de réaliser des calculs parallèles de données volumineuses (supérieures à 1 téraoctet). Les calculs peuvent être distribués sur plusieurs machines ce qui permet de répartir les charges de travail et d'ajuster facilement le nombre de serveurs suivant les besoins.

Plusieurs implémentations de MapReduce existent dont les plus connues sont l'implémentation réalisée par Google nommée Google MapReduce et l'implémentation Apache MapReduce. L'implémentation de Google est propriétaire alors qu'Apache MapReduce est open source.

Apache MapReduce est un des deux composants majeurs du framework open source Apache Hadoop qui permet la gestion des Big Data.

V.5.4 Restitution

Le Big data met aussi l'accent sur l'importance de restituer efficacement les résultats d'analyse et d'accroître l'interactivité entre utilisateurs et données. Ainsi, des produits comme QlikView, Tableau (de Tableau Software), PowerView, SpotFire proposent des visualisations graphiques innovantes.

V.6 Ecosystème Hadoop

Pour certains, l'avenir appartiendra à ceux qui seront capable d'analyser les vastes volumes de données qu'ils ont collectés.

En écologie, un écosystème est l'ensemble formé par une association ou communauté d'êtres vivants et son environnement biologique, géologique, édaphique, hydrologique, climatique, etc. l'écosystème Hadoop est l'ensemble des projets Apache ou non, liés à Hadoop et qui sont appelés à se cohabiter.

Hadoop désigne un framework Java libre ou un environnement d'exécution distribuée, performant et scalable, dont la vocation est de traiter des volumes des données considérables. Il est le socle d'un vaste écosystème constitué d'autres projets spécialisés dans un domaine particulier parmi lesquels on compte les entrepôts de données, le suivi applicatif (monitoring) ou la persistance de données.

Le schéma ci-après présente les différents éléments de l'écosystème Hadoop en fonction du type d'opération.

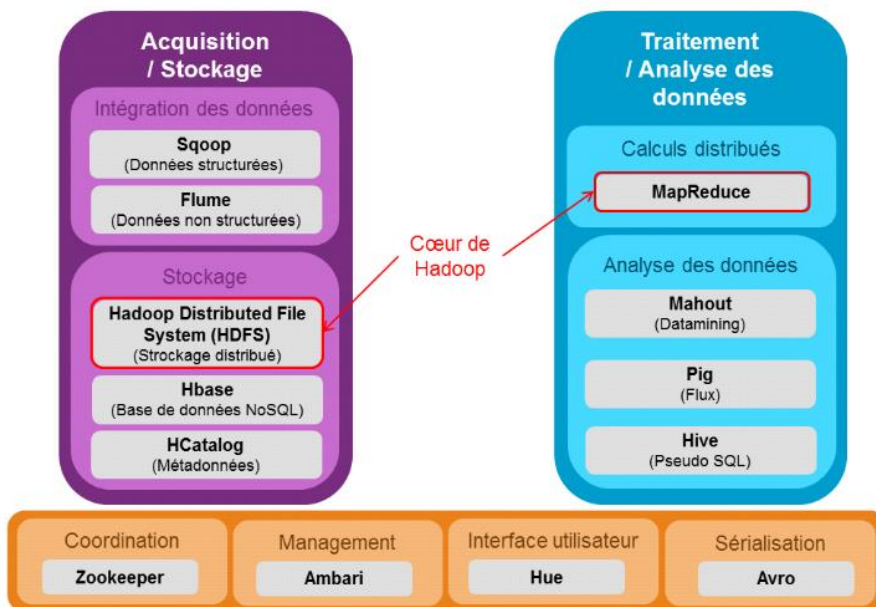


Figure 4 : Ecosystème Hadoop

V.6.1 Hadoop kernel

Hadoop Kernel est le cœur de l'écosystème Hadoop. Ce framework est actuellement le plus utilisé pour faire du Big Data. Hadoop est écrit en Java et a été créé par Doug Cutting et Michael Cafarella en 2005. Le noyau est composé de 2 composants. Hadoop présente l'avantage d'être issu de la communauté open source, de ce fait un porte un message exprimant une opportunité économique. En revanche, il affiche une complexité qui est loin de rendre accessible au commun des DSI.

Nous continuerons par une présentation de quelques grands concepts d'Hadoop.

V.6.1.1 Architecture Hadoop

Le schéma ci-dessous présente l'architecture distribuée dans le contexte Hadoop.

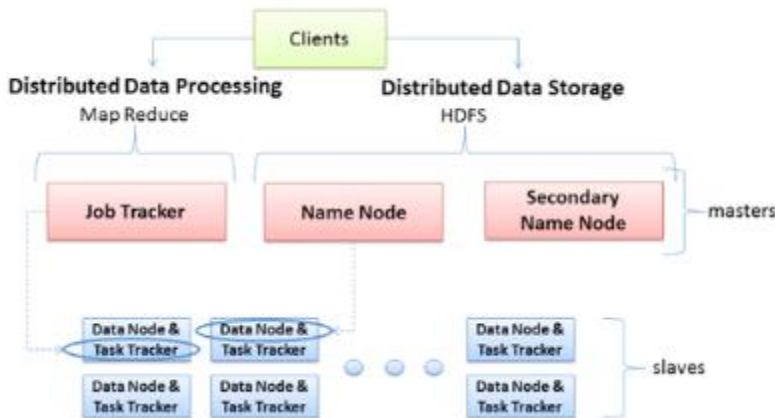


Figure 5 : Architecture Hadoop avec les principaux rôles des machines

Il est primordial de savoir qu'une architecture Hadoop est basée sur le principe maître/esclave, représentant les deux principaux rôles des machines. Les sous rôles relatifs au système de fichiers et à l'exécution des tâches distribuées sont associés à chaque machine de l'architecture.

Les machines maîtres ont trois principaux rôles qui leur sont associées :

- **JobTracker** : c'est le rôle qui permet à la machine maître de lancer des tâches distribuées, en coordonnant les esclaves. Il planifie les exécutions, gère l'état des machines esclaves et agrège les résultats des calculs.
- **NameNode** : ce rôle assure la répartition des données sur les machines esclaves et la gestion de l'espace de nom du cluster. La machine qui joue ce rôle contient des métadonnées qui lui permettent de savoir sur quelle machine chaque fichier est hébergé.
- **SecondaryNameNode** : ce rôle intervient pour la redondance du NameNode. Normalement, il doit être assuré par une autre machine physique autre que le NameNode car il permet en cas de panne de ce dernier, d'assurer la continuité de fonctionnement du cluster.

Deux rôles sont associés aux machines esclaves :

- **TaskTracker** : ce rôle permet à un esclave d'exécuter une tâche MapReduce sur les données qu'elle héberge. Le TaskTracker est piloté par JobTracker d'une machine maître qui lui envoie la tâche à exécuter.
- **DataNode** : dans le cluster, c'est une machine qui héberge une partie des données. Les noeuds de données sont généralement répliqués dans le cadre d'une architecture Hadoop dans l'optique d'assurer la haute disponibilité des données.

Lorsqu'un client veut accéder aux données ou exécuter une tâche distribuée, il fait appel à la machine maître qui joue le rôle de JobTracker et de Namenode.

Maintenant que nous avons vu globalement l'articulation d'une architecture Hadoop, nous allons voir deux principaux concepts inhérents aux différents rôles que nous avons présentés.

V.6.1.2 HDFS

HDFS est le système de fichiers Java, permettant de gérer le stockage des données sur des machines d'une architecture Hadoop. Il s'appuie sur le système de fichier natif de l'OS (unix) pour présenter un système de stockage unifié reposant sur un ensemble de disques et de systèmes de fichiers.

La consistance des données réside sur la redondance ; une donnée est stockée sur au moins n volumes différents.

Le NameNode rendrait le cluster inaccessible s'il venait à tomber en panne, il représente le SPOOF (maillon faible) du cluster Hadoop. Actuellement, la version 2.0 introduit le failover automatisé (capacité d'un équipement à basculer automatiquement vers un équipement alternatif, en cas de panne). Bien qu'il y ait plusieurs NameNodes, la promotion d'un NameNode se fait manuellement sur la version 1.0.

Dans un cluster les données sont découpées et distribuées en blocks selon la taille unitaire de stockage (généralement 64 ou 128 Mo) et le facteur de réplication (nombre de copie d'une donnée, qui est de 3 par défaut).

Un principe important de HDFS est que les fichiers sont de type « write-once » ; ceci est lié au fait que lors des opérations analytiques, la lecture des données est beaucoup plus utilisée que l'écriture.

V.6.1.3 MapReduce

MapReduce qui est le deuxième composant du noyau Hadoop permet d'effectuer des traitements distribués sur les noeuds du cluster. Il décompose un job (unité de traitement mettant en œuvre un jeu de données en entrée, un programme MapReduce (packagé dans un JAR (Java Archive : fichier d'archive, utilisé pour distribuer un ensemble de classes Java)) et des éléments de configuration) en un ensemble de tâche plus petites qui vont produire chacune un sous ensemble du résultat final ; ce au moyen de la fonction Map.

L'ensemble des résultats intermédiaires est traité (par agrégation, filtrage), ce au moyen de la fonction Reduce.

Le schéma ci-dessous présente le processus d'un traitement MapReduce.

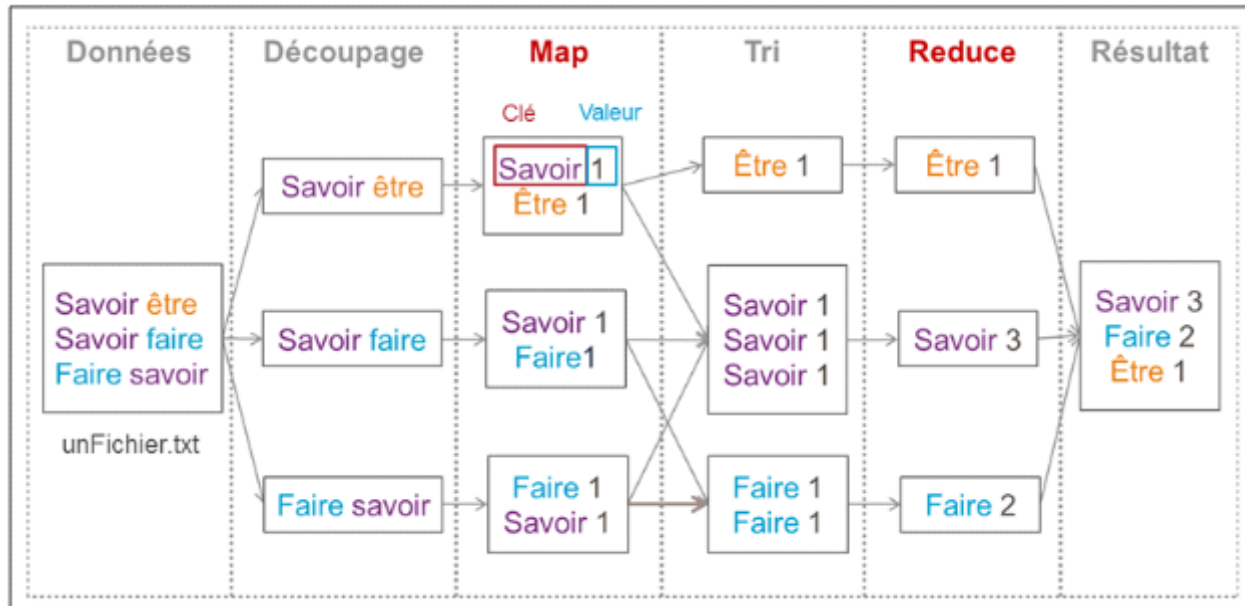


Figure 6 : Processus d'un traitement MapReduce.

Le MapReduce présenté sur le schéma permet de trouver le nombre d'occurrence des mots d'un fichier nommé ici « unFichier.txt ».

Durant la phase de « Découpage », les lignes du fichier sont découpées en blocs.

Puis lors de la phase « Map », des clés sont créées avec une valeur associée. Dans cet exemple une clé est un mot et la valeur est 1 pour signifier que le mot est présent une fois.

Lors du « Tri », toutes les clés identiques sont regroupées, ici ce sont tous les mots identiques.

Ensuite lors de la phase « Reduce » un traitement est réalisé sur toutes les valeurs d'une même clé. Dans cet exemple on additionne les valeurs ce qui permet d'obtenir le nombre d'occurrence des mots.

V.6.1.4 Modes d'utilisation

Hadoop peut être sous trois modes différents :

- Mode Standalone : ce mode a pour objectif de tester le fonctionnement d'une tâche MapReduce. Ici, la tâche est exécutée sur le poste client dans la seule machine virtuelle Java (JVM), pas besoin d'une configuration particulière car c'est la mode de fonctionnement de base de Hadoop.
- Mode Pseudo distributed : ce mode permettra de tester l'exécution d'une tâche MapReduce sur une seule machine tout en simulant le fonctionnement d'un

cluster Hadoop. Le job est exécuté sur la machine et les opérations de stockage et de traitement du job seront gérées par des processus Java différents. L'objectif de ce mode est de tester le bon fonctionnement d'un job sans besoin de mobiliser toutes les ressources du cluster.

- Mode Fully distributed : c'est le mode réel d'exécution d'Hadoop. Il permet de mobiliser le système de fichier distribué et les jobs MapReduce sur un ensemble de machines ; ceci nécessite de disposer de plusieurs postes pour héberger les données et exécuter les tâches.

Dans la suite, d'autres composants qui entrent dans l'écosystème Hadoop sont présentés.

V.6.2 Composants Apache Hadoop

V.6.2.1 Hbase

Hbase est un système de gestion de bases de données non relationnelles distribuées, écrit en Java, disposant d'un stockage structuré pour les grandes tables. C'est une base de données NoSQL, orientée colonnes. Utilisé conjointement avec HDFS, ce dernier facilite la distribution des données de Hbase sur plusieurs noeuds. Contrairement à HDFS, HBase permet de gérer les accès aléatoires read/write pour des applications de type temps réel.

V.6.2.2 HCatalog

HCatalog permet l'interopérabilité d'un cluster de données Hadoop avec d'autres systèmes (Hive, Pig, ...). C'est un service de gestion de tables et de schéma Hadoop. Il permet :

D'attaquer les données HDFS via des schémas de type tables de données en lecture/écriture.

D'opérer sur des données issues de MapReduce, Pig ou Hive.

V.6.2.3 Hive

Hive est un outil de requêtage des données, il permet l'exécution de requêtes SQL sur le cluster Hadoop en vue d'analyser et d'agréger les données. Le langage utilisé par Hive est nommé HiveQL. C'est un langage de visualisation uniquement, raison pour laquelle seules les instructions de type « Select » sont supportées pour la manipulation des données.

Hive proposent des fonctions prédéfinies (calcul de la somme, du maximum, de la moyenne), il permet également à l'utilisateur de définir ses propres fonctions qui peuvent être de 3 types :

- UDF (User Defined Function) : qui prennent une ligne en entrée et retournent une ligne en sortie. Exemple : mettre une chaîne de caractère en minuscule et inversement
- UDAF (User Defined Aggregate Function) : qui prennent plusieurs lignes en entrée et retournent une ligne en sortie. Exemple : somme, moyenne, max....
- UDTF (User Defined Table Function) : qui prennent une ligne en entrée et retournent plusieurs lignes en sortie. Exemple : découper une chaîne de caractère en plusieurs mots.

Hive utilise un connecteur jdbc/odbc, ce qui permet de le connecter à des outils de création de rapport comme QlikView.

V.6.2.4 Pig

Pig est une brique qui permet le requêtage des données Hadoop à partir d'un langage de script (langage qui interprète le code ligne par ligne au lieu de faire une compilation). Pig est basé sur un langage de haut niveau appelé PigLatin. Il transforme étape par étape des flux de données en exécutant des programmes MapReduce successivement ou en utilisant des méthodes prédéfinies du type calcul de la moyenne, de la valeur minimale, ou en permettant à l'utilisateur de définir ses propres méthode appelées User Defined Functions (UDF).

V.6.2.5 Sqoop

Sqoop est une brique pour l'intégration des données. Il permet le transfert des données entre un cluster et une base de données relationnelles.

V.6.2.6 Flume

Flume permet la collecte et l'agrégation des fichiers logs, destinés à être stockés et traités par Hadoop. Il s'interface directement avec HDFS au moyen d'une API native.

V.6.2.7 Oozie

Oozie est utilisé pour gérer et coordonner les tâches de traitement de données à destination de Hadoop. Il supporte des jobs MapReduce, Pig, Hive, Sqoop, etc.

V.6.2.8 Zookeeper

Zookeeper est une solution de gestion de cluster Hadoop. Il permet de coordonner les tâches des services d'un cluster Hadoop. Il fournit au composants Hadoop les fonctionnalités de distribution.

V.6.2.9 Ambari

Ambari est une solution de supervision et d'administration de clusters Hadoop. Il

propose un tableau de bord qui permet de visualiser rapidement l'état d'un cluster. Ambari inclut un système de gestion de configuration permettant de déployer des services d'Hadoop ou de son écosystème sur des clusters de machines. Il ne se limite pas à Hadoop mais permet de gérer également tous les outils de l'écosystème.

V.6.2.10 Mahout

Mahout est un projet de la fondation Apache visant à créer des implémentations d'algorithmes d'apprentissage automatique et de datamining.

V.6.2.11 Avro

Avro est un format utilisé pour la sérialisation des données.

Le caractère open source de Hadoop a permis à des entreprises de développer leur propre distribution en ajoutant des spécificités.

V.7 Solutions Big Data sur le marché

Sur le marché, on retrouve une panoplie de solutions, chacune avec ses particularités.

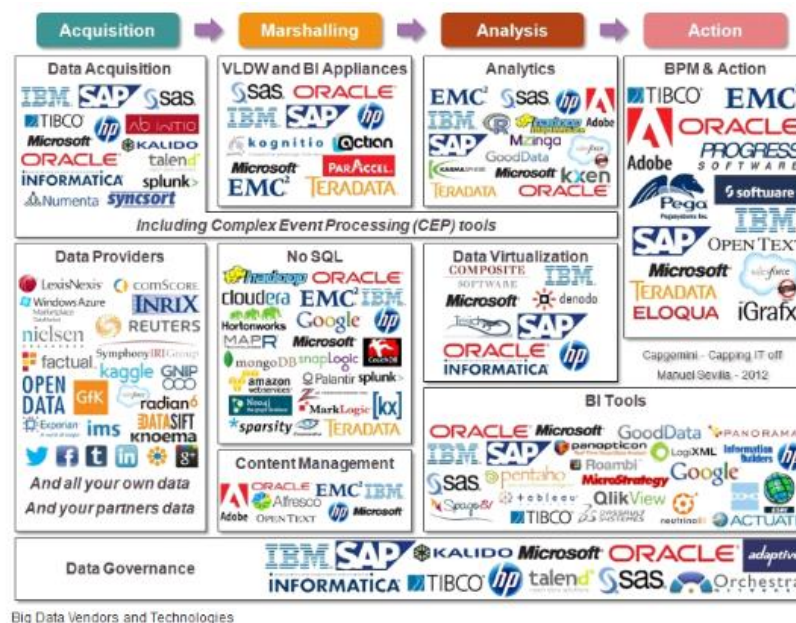


Figure 7 : Solutions Big Data sur le marché.

Parmi cette panoplie, trois se distinguent par le développement d'une distribution Hadoop :

- Cloudera : c'est le leader, ce qui lui donne une légitimité avec un nombre de clients supérieur à celui de ses concurrents. Le fait de disposer du créateur du

framework Hadoop dans ces rangs est un grand avantage.

- MapR : cette distribution offre une solution un peu éloignée d'Apache Hadoop car elle intègre sa propre vision de MapReduce et HDFS. Elle vient juste après Cloudera.
- Hortonworks : cette distribution est l'unique plateforme entièrement Hadoop. Sa stratégie est de se baser sur les versions stables de Hadoop plutôt que sur les dernières versions.

Cloudera est la solution qui a été retenue pour la réalisation des tests.

VI. Solution mise en place

VI.1 Choix de la solution et tests techniques réalisés

La distribution Cloudera a été utilisée pour plusieurs raisons. Tout d'abord le fait que Cloudera propose une version open source qui utilise les principaux composants de Hadoop. Ensuite, la distribution de Cloudera est la plus mature sur le marché avec déjà la quatrième version nommée CDH5.

Cloudera existe en trois versions : Free Edition, Standard et Enterprise. Nous avons décidé d'utiliser la version Enterprise (car elle était gratuite pour une période de 60 jours et passait en version Standard si l'on arrivait au terme de la période d'essai sans s'être procuré d'une licence) afin d'explorer les fonctionnalités qu'elle offre vu que celles-ci sont adaptées pour un contexte d'entreprise. Cloudera propose un outil pour superviser et automatiser le déploiement des clusters Hadoop nommé Cloudera Manager. C'est ce composant que nous avons utilisé pour installer le cluster Hadoop.

Les fonctionnalités clés de Cloudera sont les suivantes :

- Gestion du cluster : elle permet de déployer, configurer et exploiter facilement des clusters de façon centralisée, avec une administration intuitive pour tous les services, les hôtes et les workflows.
- Monitoring du cluster : elle permet de maintenir une vue centralisée de toutes les activités de la grappe (noeuds du cluster), ses contrôles proactifs et des alertes.
- Diagnostic du cluster : cette fonctionnalité permet de diagnostiquer et résoudre facilement les problèmes avec l'aide des rapports opérationnels et des tableaux de bord, des événements, de l'affichage des journaux, des pistes d'audit.
- Intégration : cette fonctionnalité permet d'intégrer les outils de surveillance existants (SNMP, SMTP) avec Cloudera Manager.

VI.1.1 Architecture du cluster mise en place

Le schéma ci-dessous présente l'architecture du cluster Hadoop que nous avons mis

en place en local chez IB-Maroc.

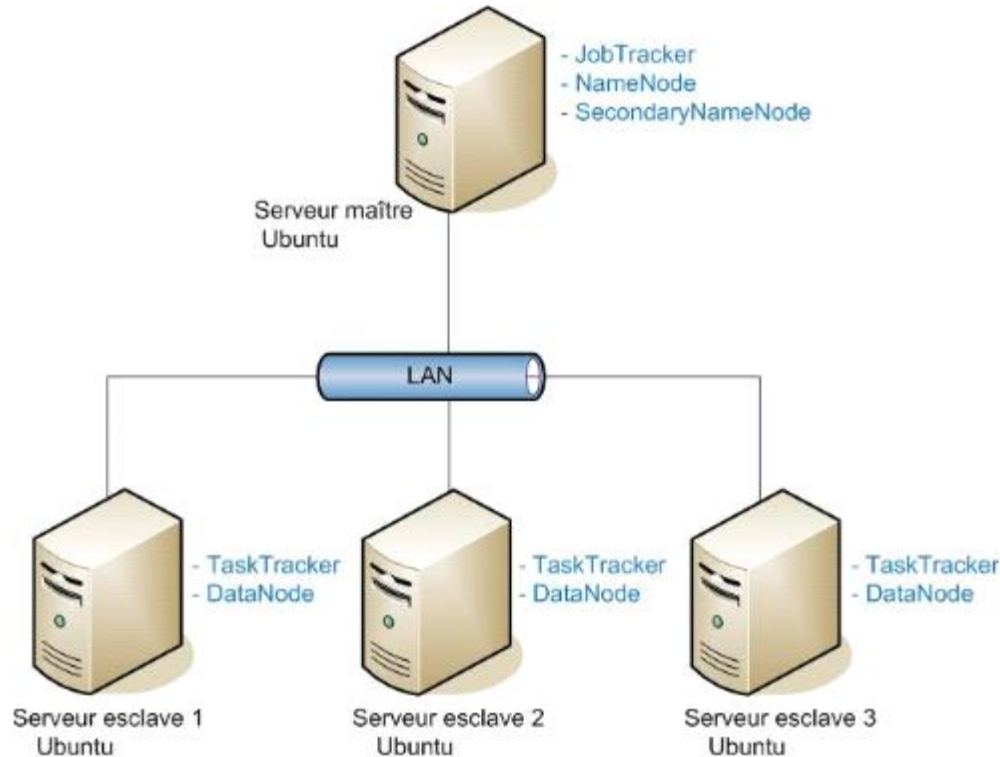


Figure 8 : Architecture du cluster Hadoop mis en place

Ce cluster est constitué de postes standards équipés de système d'exploitation Ubuntu (version 14.04). Ce schéma présente les différentes machines (maître et esclave) du cluster et les rôles qui leurs sont associés dans le cadre d'une architecture Hadoop.

VI.1.2 Architectures des composants des machines

Ces architectures présentent les composants qui sont déployés sur chaque machine. Sur les machines, plusieurs composants de l'écosystème Hadoop sont installés, à savoir les composants du kernel Hadoop, les composants Apache Hadoop et des composants spécifiques à Cloudera.

Les composants propres à Cloudera sont les suivants :

- Cloudera Manager : il permet l'installation automatisée des composants de la plateforme sur une machine. Il permet également de centraliser la configuration des composants du cluster, d'effectuer un contrôle d'intégrité des composants à la fin de l'installation.
- Hue : c'est un outil qui fournit des interfaces utilisateur pour les applications Hadoop.
- Impala : c'est moteur temps réel de requêtage SQL parallélisé de données

stockées dans HDFS ou HBase. Il n'utilise pas MapReduce ce qui lui permet d'une part d'optimiser le temps d'exécution des requêtes et d'autre part, une consultation des données de façon interactive.

- Solr (Cloudera search), c'est un outil qui n'est pas propre à Cloudera mais qu'il intègre pour effectuer la recherche sur les données du cluster.

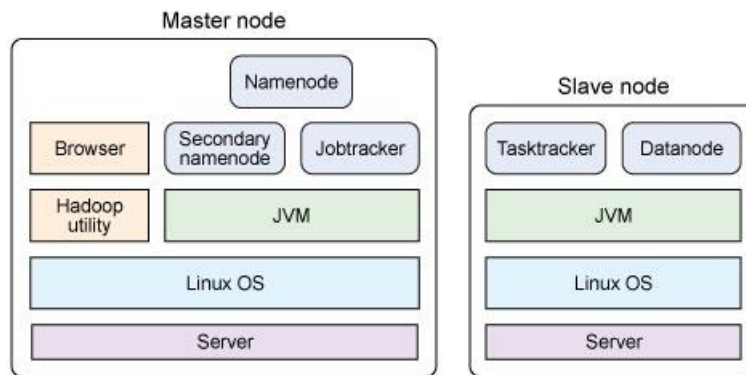


Figure 9 : Architecture des composants

VI.1.3 Déploiement de la plateforme

Le déploiement du cluster peut se faire soit manuellement (utilisation des packages), soit automatique avec Cloudera Manager. Nous avons fait une installation automatique pour limiter des éventuelles erreurs d'installation et aussi compte tenu du temps que nous voulions préserver pour les étapes prochaines du cheminement de notre stage.

La figure ci-dessous présente la page d'accueil de la console web de Cloudera Manager après l'installation du cluster, listant les différents composants installés sur le cluster Hadoop.

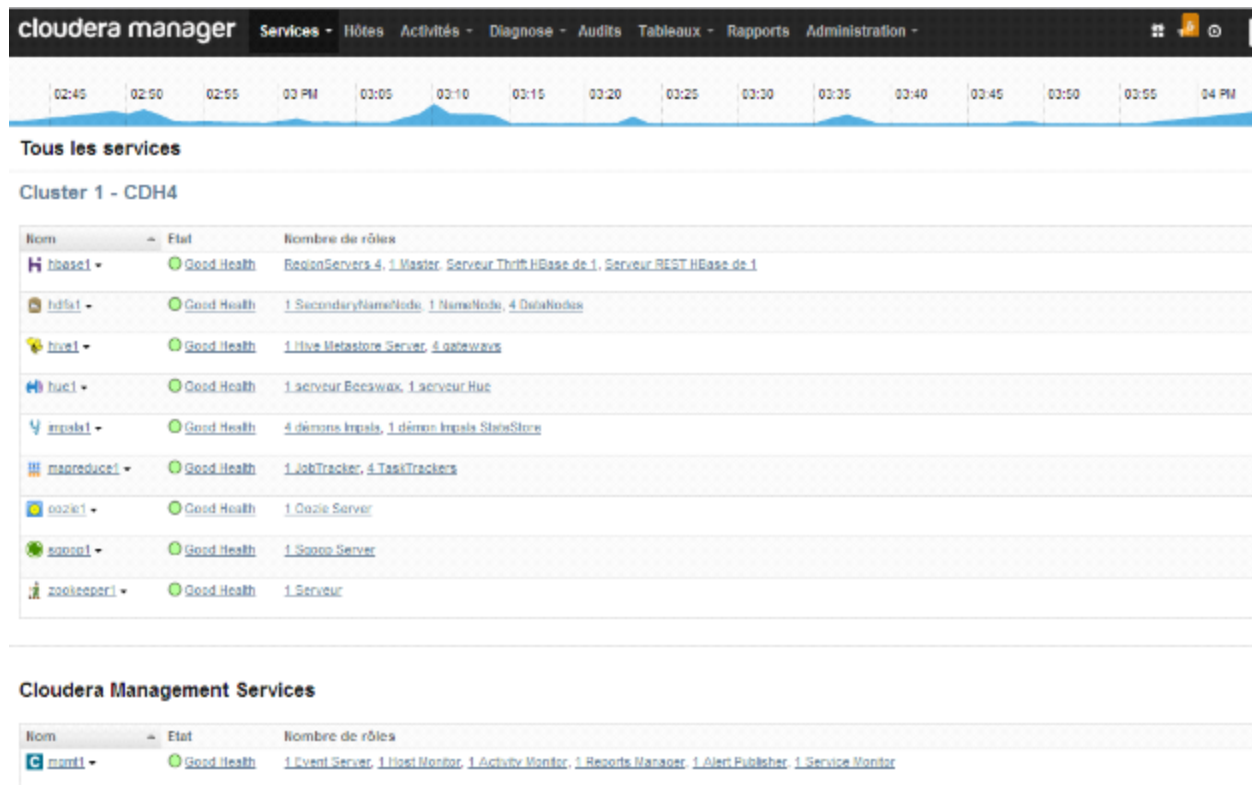


Figure 10 : Page d'accueil de Cloudera Manager après l'installation du cluster

VI.1.4 Tests techniques réalisés

VI.1.4.1 But des tests

Lorsqu'on réalise un test, il est important de d'utiliser une démarche précise et de définir les objectifs à atteindre à la fin du test, ce qui permettra de faire un contrôle à la fin, par rapport aux résultats obtenus pour savoir si le test a été satisfaisant ou non.

La démarche a consisté à tester les composants séparément ou conjointement, à identifier les fonctionnalités de chacun.

Les objectifs des différents tests étaient :

- D'acquérir une meilleure connaissance de chaque composant.
- Pouvoir présenter le fonctionnement de chaque composant.
- Faire des recommandations par rapport à l'utilisation d'un composant.
- Montrer la mise en œuvre de certains concepts.

VI.1.4.2 Test de l'environnement existant

VI.1.4.2.1 Traitement de données avec MapReduce et HDFS

Lorsque les données sont collectées sans un outil spécifique, elles sont d'abord stockées sur le système de fichier local, puis l'utilisateur les déplace sur le système de fichier HDFS. HDFS repartit et réplique les données sur les différents noeuds du cluster en tenant compte du facteur de réplication pour assurer une certaine tolérance à d'éventuelles pannes. Lors du traitement, toutes les données concernées sont mobilisées quel que soit leur emplacement.

La problématique sous-jacente est celle de savoir comment les traitements sont réalisés sur des données dispersées dans le cluster.

Ce sont les composants du noyau Hadoop qui participent au traitement : MapReduce et HDFS. Pour réaliser le traitement, il faut développer un programme java composé de deux fonctions principales : map et reduce.

Pour le test, nous allons vérifions que nous pouvons faire toutes les opérations du système de fichiers d'habitude telles que la lecture de fichiers, création de répertoires, déplacement des fichiers, la suppression de données, et le listing de répertoires.

Nous créons en local le fichier charabia.txt dont nous remplissons quelques lignes à partir de /dev/random, nous faisons subir à charabia.txt un hashage md5 avant de le transférer sur HDFS puis le copier depuis HDFS vers local sous le nom charabia2.txt que nous soumettons à son tour à un hashage md5 pour vérifier que le fichier a bel et bien survécu à son aller-retour sur HDFS

```
bash-4.3$ touch charabia.txt
bash-4.3$ cat /dev/random >> charabia.txt
^C

bash-4.3$ hadoop fs -copyFromLocal charabia.txt hdfs://localhost/user/ibmrc/hdfs_charabia.txt
bash-4.3$ hadoop fs -copyToLocal hdfs_charabia.txt charabia2.txt
bash-4.3$ md5sum charabia.txt
72d05974bbe0fb2c4a5523608e88b6cc charabia.txt
bash-4.3$ md5sum charabia2.txt
72d05974bbe0fb2c4a5523608e88b6cc charabia2.txt
```

VI.1.4.3.2 Restitution des données

La restitution des résultats des traitements dans Hadoop n'est pas conviviale à cause des propriétés intrinsèques de HDFS qui est un système de fichier et non outil de présentation de données. De plus, les résultats de traitement ne sont pas assez structurés pour faciliter leur lisibilité.

La problématique qui se pose est de savoir comment faciliter la lisibilité des résultats des traitements stockés dans HDFS.

Dans l'écosystème Hadoop, plusieurs solutions sont possibles (utilisation de Sqoop, Pig), Nous avons opté pour l'utilisation de Hive car elle offre déjà la possibilité de

connexion à un outil de visualisation de données.

Requêtage des données via Hive

Le composant Hive permet de créer une structure des données à partir des données stockées dans HDFS. L'objectif était donc d'appréhender concrètement comment Hive améliore la restitution des informations provenant de HDFS.

Les requêtes étant réalisées avec HiveQL, langage similaire au langage SQL, l'appropriation du langage est plus rapide.

Nous avons testé des requêtes sur des données stockées dans HDFS et visualisé les résultats depuis la console Hive et l'interface web. Les données sont stockées sous forme de tables. Les exemples sont présentés dans la suite du document.

Hive se base sur des traitements MapReduce pour réaliser les requêtes ce qui suscite des temps de réponse assez long. Pour pallier ce problème, il est possible de partitionner les tables pour réaliser les requêtes sur certaines parties de la table au lieu de la table entière. Cela permet également de ne recharger qu'une partie des données en cas de modification ou de suppression ce qui est important dans le cas de grands volumes de données.

VI.2 Démonstrateur Big Data

Le démonstrateur Big Data est le livrable qui présente la mise en œuvre des cas d'utilisation du Big data. C'est un livrable qui pourra être utilisé à des fins de présentation à un client. Le démonstrateur déroule les cas d'utilisation de la création des jeux de données pour les tests à la restitution des données via QlikView (outil de visualisation de données), en passant par les traitements et la structuration des données.

En terme d'architecture réseau et architecture des composants, nous avons utilisé l'architecture présentées plus haut dans ce document.

Les différents cas d'utilisations traités sont les suivants :

- Cas d'utilisation 1 : Les taux d'adolescents et d'adultes sur quelques réseaux sociaux
- Cas d'utilisation 2 : Les taux d'utilisation de certaines activités sur internet via PC et Mobile

La suite présente la création des données des tests.

VI.2.1 Les jeux de données de test

Pour la mise en œuvre des cas d'utilisation de test, à défaut de disposer des données

réelles, nous avons sollicité l'aide de mademoiselle Angeline KONE qui – en plus de nous avoir fourni en documentation sur l'écosystème Hadoop et nous a permis d'utiliser sa synthèse dérivée de son stage chez Capgemini – nous a fourni en jeux de données en entrée aux fins de test de notre configuration.

VI.2.1.1 Cas d'utilisation 1

Réseau social / Tranche	Effectif	Fichier en entrée
Facebook /Adolescent	164	TauxAdoFacebook.csv
Facebook / Adulte	164	TauxAdultesFacebook.csv
Twitter /Adolescent	395	TauxAdoTwitter.csv
Twitter / Adulte	542	TauxAdultesTwitter.csv
Pinterest /Adolescent	319	TauxAdoPinterest.csv
Pinterest / Adulte	542	TauxAdultesPinterest.csv
Instagram /Adolescent	318	TauxAdoPinterest.csv
Instagram / Adulte	532	TauxAdultesPinterest.csv
Tumblr / Adulte	318	TauxAdoTumblr.csv
Tumblr / Adulte	542	TauxAdultesTumblr.csv

Format d'une ligne d'un fichier

id,nom_profil,nom,prenom,age,sexe,email,ville,pays,numero_de_rue,rue,code_postal,t
aux_Ado_facebook

VI.2.1.3 Cas d'utilisation 2

Effectif	Fichier en sortie
1000	TempsPCMobileSurInternet.csv

Format d'une ligne du fichier TempsPCMobileSurInternet.csv

id,maps,weather,music,social_network,sport,retail,newspaper,game,email,portal

Exemple de ligne du fichier TempsPCMobileSurInternet.csv (chaque mot correspond à une activité)

16,Mobile,Mobile,Mobile,PC,PC,Mobile,Mobile,Mobile,PC,PC

VI.2.2 Scénario d'exécution des cas d'utilisation

Le schéma ci-dessous présente les enchaînements à suivre pour dérouler complètement un cas d'utilisation.

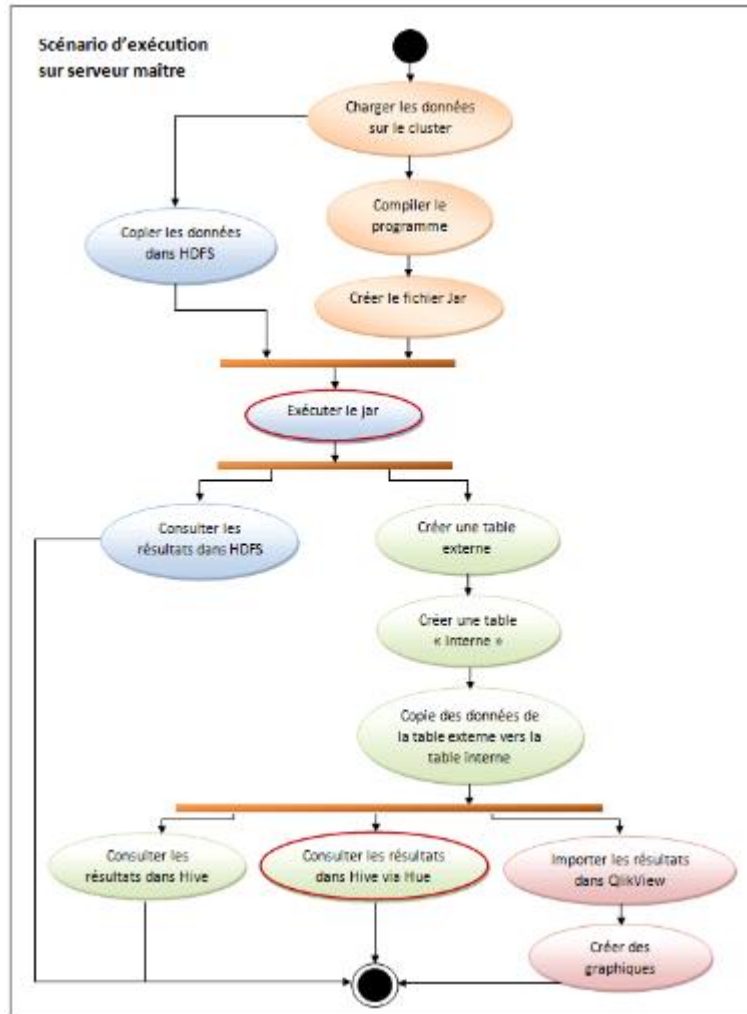


Figure 16 : Scénario d'exécution des cas d'utilisation

VI.2.3 Restitution des données via Hive

Comme mentionné un peu plus haut dans ce document, pour améliorer la lisibilité des données dans HDFS, nous avons importé les résultats des traitements dans Hive.

Dans Hive, lorsqu'une table est créée avec des données stockées dans HDFS, les données sont déplacées de HDFS vers Hive : aucune copie sur HDFS n'est préservée pour d'éventuelles utilisations ultérieures. Pour éviter que les données ne soient supprimées de HDFS, il faut créer une table externe qui pointe vers les données présentes sur HDFS et une table simple ; copier les données de la table externe vers la

table simple, ceci permettra de pouvoir réutiliser les données présentes sur HDFS.

Exemple du cas d'utilisation 1

Création de la table externe

```
CREATE EXTERNAL TABLE types_utilisateurs_ex (id int,nom_profil string ,nom string ,prenom string ,age int ,sexe char ,email string ,ville string ,pays string,numero_de_rue string,rue string,code_postal string,taux_Ado_facebook float) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION 'hdfs://localhost:8020/user/admin/typesutilisateurs_rs/output';
```

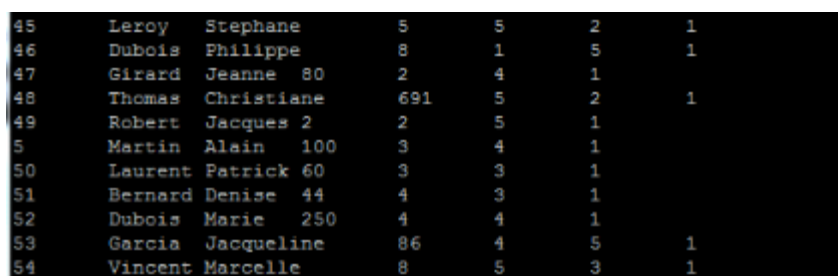
Création de la table simple

```
CREATE TABLE types_utilisateurs_in (id int,nom_profil string ,nom string ,prenom string ,age int ,sexe char ,email string ,ville string ,pays string,numero_de_rue string,rue string,code_postal string,taux_Ado_facebook float);
```

Copie des données de la table externe vers la table simple

```
FROM types_utilisateurs_ex ue INSERT OVERWRITE TABLE types_utilisateurs_in SELECT ue.id, ue.nom_profil, ue.nom, ue.prenom, ue.age, ue.sexe, ue.email, ue.ville, ue.pays, ue.numero_de_rue, ue.rue, ue.code_postal, ue.taux_Ado_facebook;
```

Avec Hive, il est possible de visualiser les contenus des tables soit en sur la console, soit sur l'interface graphique. Le schéma ci-dessous présente un extrait du résultat de la requête « *select * from utilisateurs_in;* » sur l'invite de commande.



45	Leroy	Stephane		5		5		2		1		
46	Dubois	Philippe		8		1		5		1		
47	Girard	Jeanne	80	2		4		1				
48	Thomas	Christiane	691	5		2				1		
49	Robert	Jacques	2	2		5		1				
5	Martin	Alain	100	3		4		1				
50	Laurent	Patrick	60	3		3		1				
51	Bernard	Denise	44	4		3		1				
52	Dubois	Marie	250	4		4		1				
53	Garcia	Jacqueline	86	4		5				1		
54	Vincent	Marcelle		8		5		3		1		

Figure 18 : Résultat de requête sur l'invite de commande de Hive

Nous avons traité tous les résultats présents dans HDFS de la même façon pour améliorer leur visualisation dans Hive.

Les données ainsi structurées, sont importées dans un outil de visualisation qui va faciliter leur analyse et la prise de décision.

VI.2.4 Restitution des données via QlikView

QlikView est un outil du monde de la Business Intelligence (BI), utilisé dans le cadre de la visualisation des données. Il dispose d'une interface utilisateur conviviale et utilise la technologie in-memory (technologie permettant d'effectuer des traitements en mémoire vive sur de grandes quantités de données) pour une interaction plus rapide avec les données. La plateforme Big Data mise en place avec cluster Hadoop est indépendante des plateformes de Business Intelligence des entreprises. Néanmoins, ces deux mondes partagent un même objectif qui est d'extraire de la valeur des données dont disposent les entreprises afin de faciliter la prise de décision.

A la différence de la BI qui est plus mature et qui fournit des outils efficaces et ergonomiques, les plateformes Big Data ne permettent pas encore une restitution de données conviviale.

Pour répondre à la problématique de visualisation, le monde du Big Data s'appuie actuellement sur les outils du monde de la BI. Il est donc judicieux d'utiliser les outils de la BI déjà connus et adoptés par les utilisateurs.

Pour exploiter cet outil, premièrement, nous avons installé un connecteur ODBC pour connecter Hive à QlikView et tester la connexion à Hive à partir de la machine physique. En second lieu, nous avons importé les données via un script QlikView puis créé les graphiques des différents cas d'utilisation.

Les schémas ci-dessous présentent les résultats des différents Cas d'utilisation s via des graphiques créés sur QlikView.

VI.2.4.1 Cas d'utilisation 1

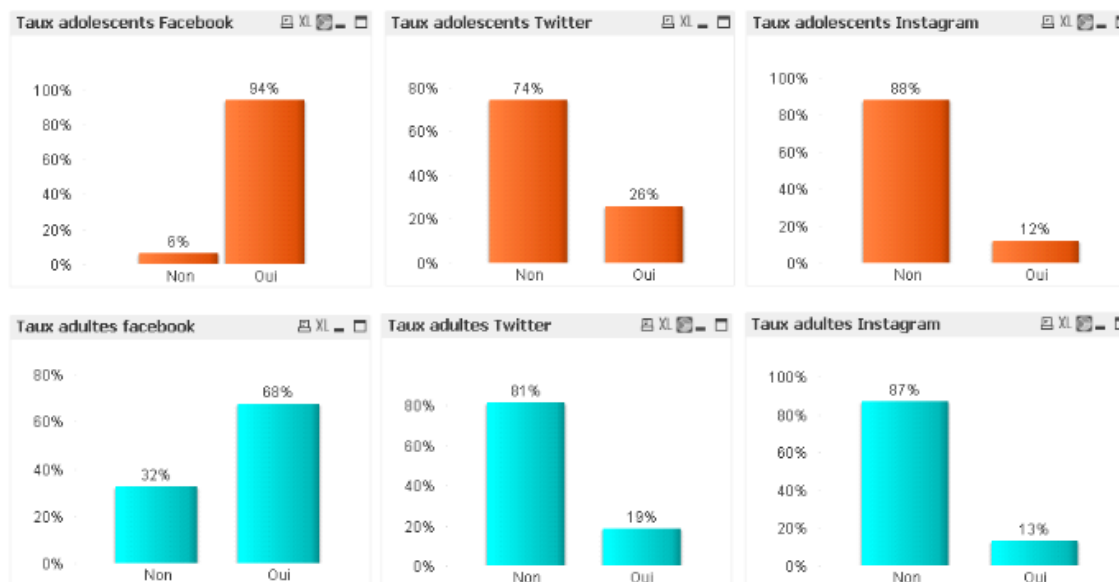


Figure 20 : Taux d'adolescents et d'adulte sur les réseaux sociaux

VI.2.5.2 Cas d'utilisation 2

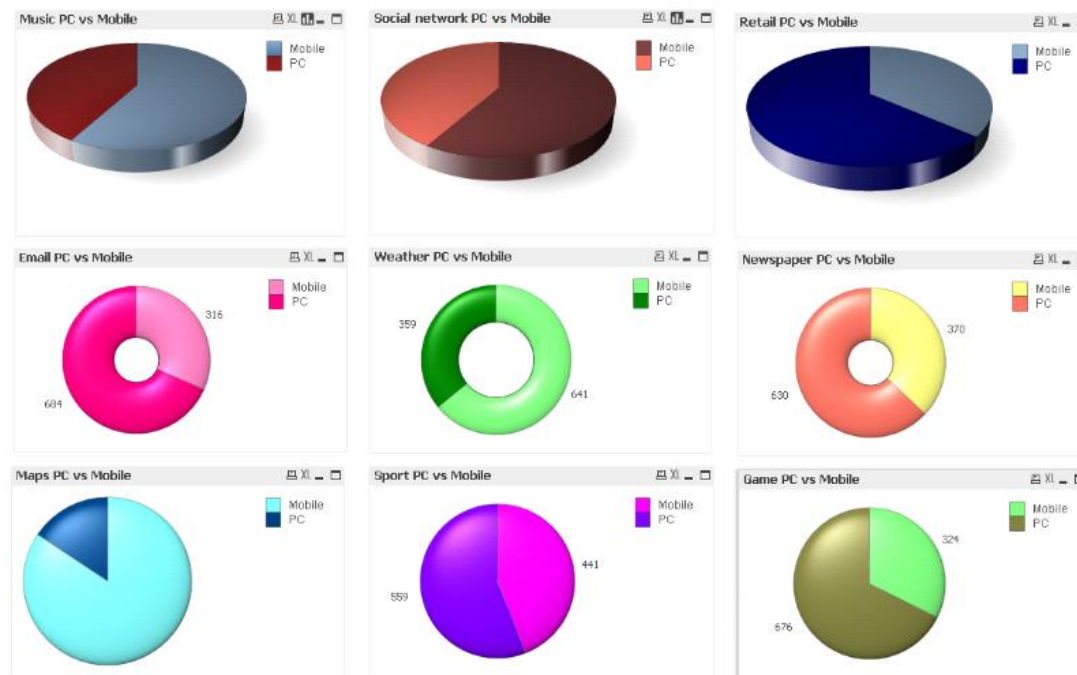


Figure 21 : Activités sur internet PC vs Mobile

VI.3 Outils utilisés

En dehors des composants de l'écosystème Hadoop, du service de conteneurisation Docker et des composants propres à Cloudera, nous avons utilisé :

- Putty pour l'accès aux machines via ssh,
- Dropbox pour le partage des fichiers,
- NodePad++ et NetBeans pour le développement des programmes,
- QlikView pour la réalisation de graphiques.

VII. Déploiement sur Cloud

VII.1 Qu'est ce que docker.

Docker permet d'embarquer une application dans un container virtuel qui pourra s'exécuter sur n'importe quel serveur (Linux et bientôt Windows). C'est une technologie qui a pour but de faciliter les déploiements d'une application, et la gestion du dimensionnement de l'infrastructure sous-jacente. Cette solution est proposée en open source (sous licence Apache 2.0) par une société américaine, également appelée

VII.2 Quelle différence avec la virtualisation ?

La virtualisation permet, via un hyperviseur, de simuler une ou plusieurs machines physiques, et les exécuter sur un serveur machine. Cette machine physique intègre elle-même un OS sur lequel ses applications sont exécutées. Ce n'est pas le cas du container. Le container fait en effet directement appel à l'OS de la machine hôte pour réaliser ses appels système. Historiquement, Docker repose sur le format de containers Linux, alias LXC. Il l'étend par le biais d'une API dans l'optique d'exécuter les applications dans des containers standards portables d'un serveur Linux à l'autre.

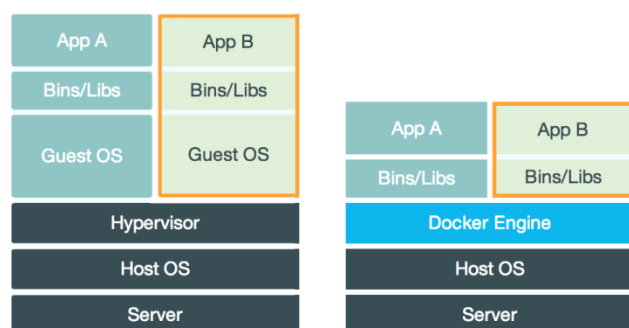


Figure 20 : Différence entre la virtualisation VM et Docker

VII.3 Les principaux avantages de Docker comparé à la virtualisation ?

Comme le container n'embarque pas d'OS supplémentaire, à la différence de la machine virtuelle, il est par conséquent beaucoup plus léger que cette dernière. Il n'a pas besoin en effet d'activer un second système pour exécuter ses applications. Cela se traduit par un lancement beaucoup plus rapide, mais aussi par la capacité à migrer beaucoup plus facilement un container d'une machine physique à l'autre, du fait de son plus faible poids. Typiquement, une machine virtuelle pourra peser plusieurs Go, alors qu'un container ne représentera, lui, quelques Mo.. Mais Docker présente un autre gros avantage. Grâce à leur légèreté, les containers sont portables, et ce, y compris entre les clouds ayant implémenté cette technologie... Et les plus grands clouds l'ont déjà intégré. C'est le cas d'Amazon sur AWS, Microsoft sur Azure, et Google sur Google Compute, mais aussi d'OVH et DigitalOcean, etc.

VII.4 Construction de l'image du conteneur : les Dockerfiles

VII.4.1 Les Dockerfiles

Les Dockerfiles sont des fichiers qui permettent de construire une image Docker adaptée à nos besoins, étape par étape. La première chose à faire dans un Dockerfile est de définir quelle image nous souhaitons hériter.

```
FROM ubuntu:trusty
```

Par la suite, comme sur un terminal bash, nous entrons séquentiellement les manipulations qu'on souhaiterait faire sur notre serveur étape par étape.

```
RUN apt-get update
```

```
RUN apt-get install -y curl tar openssh-server openssh-client rsync python-software-properties apt-file
```

```
[...]
```

```
ENV HADOOP_VERSION 2.5.2
```

```
RUN curl -s http://www.us.apache.org/dist/hadoop/common/hadoop-${HADOOP_VERSION}/hadoop-${HADOOP_VERSION}.tar.gz | tar -xz -C /usr/local/
```

```
RUN cd /usr/local && ln -s ./hadoop-${HADOOP_VERSION} Hadoop
```

```
[...]
```

Pour des raisons de présentabilité, nous ne présenterons pas l'intégralité du DockerFile mais nous ne manquerons pas de le détailler durant notre soutenance.

VII.4.2 Construction de l'image des conteneurs

Sur un terminal bash, la commande « build » permet de construire séquentiellement l'image ainsi décrite sur le DockerFile.

```
docker build -t Hadoop-sur-docker .
Sending build context to Docker daemon 4.381 MB
Sending build context to Docker daemon
Step 0 : FROM ubuntu:trusty
--> bf84c1d84a8f
Step 1 : RUN apt-get update && apt-get install -y curl && rm -rf
/var/lib/apt/lists/*
--> Running in 93258459a279
```

Nous obtenons ainsi une image minimale de notre serveur Linux paramétré pour exposer Apache Hadoop Hive comme un service et déployable sur le cloud public. Notamment Amazon EC2 dont nous exposerons les détails durant notre soutenance, inshallah.

VIII. Conclusion

Les technologies du Big Data s'inscrivent dans une évolution continue compte tenu du fait qu'elles sont jeunes et pas encore stables, ce qui leur vaut la réticence des certaines entreprises. Actuellement, le virage technologique est d'ores et déjà annoncé. Le Big Data s'impose tout doucement, mais certains aspects ne sont pas encore à la hauteur des attentes, certaines pistes sont à explorer profondément avant l'intégration dans les systèmes d'information :

- La sécurité : elle est encore balbutiante malgré quelques initiatives comme Apache Knox (système qui fournit un point unique d'authentification et d'accès pour les services Apache Hadoop dans un cluster. Le but est de simplifier la sécurité Hadoop pour les utilisateurs (qui ont accès aux données du cluster et exécutent des jobs) et les opérateurs (qui contrôlent les accès et de gèrent le cluster).
- L'intégration avec le système d'information (SI), une plate forme Hadoop isolée et non intégrée au système d'information ne sera plus possible dans le futur (en tout cas certains besoins exigeront une interaction plus grande). Cette intégration entraînera une modification des processus et par conséquent des besoins de formation des ressources humaines.
- Les ressources compétentes : actuellement les compétences ne sont pas encore assez poussées dans le domaine
- Protection de la vie privée : la manipulation à grande échelle de des données pose aussi le problème de la vie privée. Trouver l'équilibre entre le respect de son intimité et les bénéfices tirés du big data n'est pas simple. Les utilisateurs des réseaux sociaux ignorent souvent que leurs données privées sont accessibles par un grand public, beaucoup reste à faire afin de garantir la protection des utilisateurs.

« L'avenir appartiendra à ceux qui seront capable d'analyser les vastes volumes de données qu'ils ont collectés ».

Références Bibliographiques

- Mathieu Millet, L. G. (2013, Février 3). *Quel est le paysage technologique du Big Data*. sur site web www.solucominsight.fr
- *'Programming Hive : Data Warehouse and Query Language for Hadoop'* d' Edward Capriolo, Dean Wampler et Jason Rutherglen. O'reilly Media – Septembre 2012.
- *'QlikView 11 for Developers'* de Barry Harmsen et Miguel Garcia. PaperIT – Novembre 2012.
- *'Hadoop: The Definitive Guide, 4th Edition Storage and analysis at Internet scale'* de Tom White. O'reilly Media – Janvier 2015.