Table des Matières

Sommaire

Chapitre 1

Présentation du stage

1.1 Présentation de l'organisme d'accueil

Djezzy Opérateur de télécommunications algérien dont le siège est à Alger, il est historiquement le premier des trois opérateurs de téléphonie mobiles nationaux algériens, il a été créé le 11 juillet 2001, au mois de décembre 2016, l'entreprise compte plus de 16,5 millions d'abonnés et 4000 employés .
en 2015 l'entreprise a atteint un chiffre d'affaire de 2,1 milliards de dinars Le réseau 4G de Djezzy couvre aujourd'hui plus de 25% de la population à travers le territoire nationale et compte devenir le plus grand réseau 4G en Algérie avec un investissement de plus de 15 milliards de dinars en 2017.

la société s'est engagée dans un processus de transformation pour devenir une entreprise technologique en se focalisant sur les données pour exploiter au maximum le Big data En devenant une entreprise de données elle pourrait avoir des volumes plus importants issues de ses clients qu'ils lui permettront l'analyse et l'exploitation de ces données .

Djezzy fait partie du groupe VEON, un groupe d'entreprises néerlandais de classe mondial dans les télécommunications dont le siège est à Amsterdam. opérant sur 15 marchés,
il sert plus de 240 millions de clients en Internet fixe, data et services digitaux.
VEON prépare la Révolution Digitale en mettant en place avec ses opérations dans tous les pays où il est présent, le passage du modèle traditionnel d'opérateur de télécom à un modèle d'entreprise.

Figure 1.1- LOGO VEON et DJEZZY

1.2 Cadre du Stage et Objectifs de l'Entreprise

La numérisation du monde physique va impacter tous les secteurs économiques et faire émerger de nouveaux marchés. Cette « mise en données du monde » va aussi provoquer une vaste transformation dans les jeux d'acteurs et les modèles économiques, et accélérer l'émergence de gisements de données qui concernent de très nombreux domaines.

Djezzy va jouer un rôle structurant en tant que médiateur de ces données, en assurant la collecte, le traitement et leur exposition auprès des acteurs marché tout en garantissant le respect de la vie privée. cela permet a l'entreprise d'optimiser leurs processus et de prendre un avantage compétitif non négligeable.

Notre stage à Djezzy nous a permit dans un cadre opérationnel de mettre en pratique l'ensemble des compétences que nous avons appris au cours de notre cursus universitaire, dans le cadre réel de notre futur métier, afin de favoriser notre insertion professionnelle. il s'agit dans un premier temps de :

monter en compétence sur les technologies Big Data et de réaliser un démonstrateur Big Data en s'intéressant aux offres et cas d'utilisation au sein de Djezzy.
Comprendre le métier de Big data Engineer, S'initier au domaine du Big data et se familiariser avec les différents outils de l'écosystème Hadoop (Apache Ignite ,Apache NiFi, Cassandra ,Apache Kafka ...).

Finalement notre objectif serait d'étudier la localisation en temps réel des clients Djezzy, basé sur des données venant du MSC Huawei, passant par la collection, le traitement et le stockage, dans ce scénario, un abonné entre dans une certaine zone géographique et peut recevoir un SMS opportun ou non facturé ou une bannière correspondant au média social avec une publicité ou une promotion (basée sur le comportement et les préférences de son client) d'un marchand local.

Chapitre 2

Etat de l’art

2.1 Introduction au Big Data

Pour les organisations de toutes tailles, la gestion des données est passée d'une compétence importante à un différenciateur essentiel, capable de déterminer les gagnants du marché. Ces organisations définissent de nouvelles initiatives et réévaluent les stratégies existantes pour examiner comment elles peuvent transformer leurs activités en utilisant le Big Data. Ce faisant, ils apprennent que le Big Data n'est pas une technologie, une technique ou une initiative unique. Il s'agit plutôt d'une tendance dans de nombreux domaines des affaires et de la technologie.

Plus précisément, le Big Data se est utilisé pour définir la complexité de la gestion et de la gestion de grandes quantités de données générées de nos jours. Le Big Data fait référence aux technologies et aux initiatives qui impliquent des données trop variées, en mutation rapide ou volumineuses pour que les technologies, les compétences et les infrastructures classiques puissent être traitées efficacement.
Autrement dit, le volume, la vélocité ou la variété des données est trop important. Ce sont les fameux quatre v:

Volume : énormes volumes de données provenant de journaux, de tables, de fichiers, etc.

Vitesse : les données doivent être traitées rapidement et efficacement

Variété : nous n’avons plus que des tables dans une base de données comme par le passé. Nous avons maintenant des fichiers audio non structurés. fichiers structurés, texte, etc. Le coût de la transformation de ce type de données en un format structuré dans un entrepôt est énorme, ce qui explique pourquoi le traitement des données volumineuses est en train de passer de l' ETL (entrepôt de données traditionnel) à l' ELT (lacs de données modernes). C'est-à-dire, déplacez l'entreprise vers les données plutôt que les données vers l'entreprise.

Véracité : les données proviennent de différentes sources

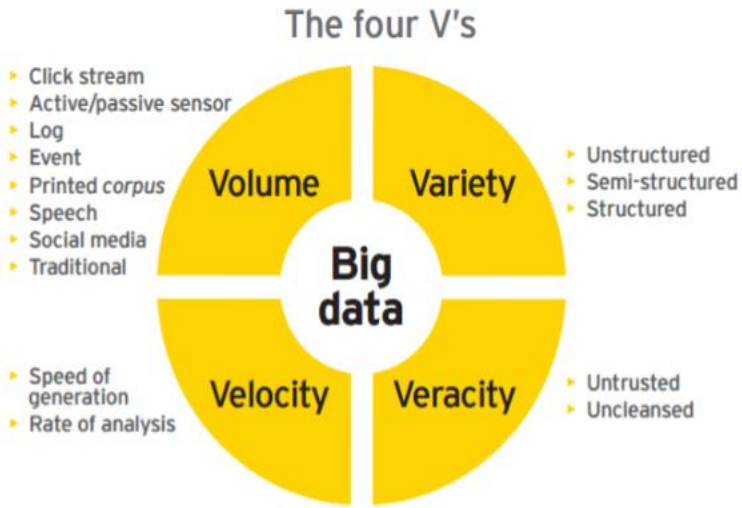

Figure 2.1 : The four V's of Big Data

2.1.1 Batch Big data

le traitement par lots est un moyen efficace de traiter de gros volumes de données. Il est traité, en particulier lorsqu'un groupe de transactions est collecté sur une période donnée. Dans ce processus, Au début, les données sont collectées, entrées et traitées. Ensuite, cela produit des résultats par lots. Nous pouvons dire que Hadoop fonctionne sur le traitement de données par lots. En outre, cela signifie qu'on compte sur la capacité du système. Nous pouvons dire, le système de traitement par lots :

Traitement par lots, accès à toutes les données. Cela pourrait calculer quelque chose de grand et de complexe .En règle générale, il est très préoccupé par le débit. Plutôt que la latence des composants individuels du calcul. Le traitement par lots a une latence mesurée en minutes ou plus.

2.1.2 Traitement en temps réel

Le traitement en temps réel implique une entrée, un traitement et une sortie continus des données. Par conséquent, il traite dans un court laps de temps. Certains programmes utilisent ce type de traitement de données. Par exemple, les guichets automatiques bancaires, les services clients, les systèmes radar et les systèmes de point de vente. Chaque transaction est directement reflétée dans le fichier principal, avec ce traitement de données. Donc, ce sera toujours à jour. Si on souhaite obtenir des résultats d'analyse en temps réel, le traitement Spark Real-Time est essentiel. Nous pouvons alimenter les données en outils d'analyse, en créant des flux de données, dès leur génération.

De plus, pour des tâches telles que la détection de fraude, le traitement en temps réel est très utile. Fondamentalement, si vous traitez des données de transaction, nous pouvons détecter cette fraude en temps réel. En outre, peut arrêter les transactions frauduleuses avant qu'elles ne se produisent, par le traitement en temps réel. Nous pouvons dire, le système de traitement en temps réel

Le traitement en temps réel permet de calculer une fonction d'un élément de données. En outre, peut dire qu'il calcule une petite fenêtre de données récentes. Le traitement en temps réel calcule quelque chose de relativement simple Bien que nous ayons besoin de calculer en temps quasi réel, quelques secondes tout au plus, nous passons au traitement en temps réel.

Dans le traitement en temps réel, les calculs sont généralement indépendants. Ils sont de nature asynchrone. Cela signifie qu'une source de données n'interagit pas directement avec le traitement du flux.

2.2 Business Intelligence vs Big Data

La Business Intelligence est le processus qui permet de prendre des décisions commerciales viables à partir de la manipulation analytique et de la présentation de données dans les limites d'un environnement d'affaires.

Si le Big Data était un morceau de bois, la Business Intelligence pourrait être la hache qui l'a coupé et l'artiste qui l'a réduit à une figurine. BI est l'action. Cela signifie que vous vous

engagez avec vos informations, qu'elles soient de taille régulière ou de grande taille et que vous y fassiez quelque chose de significatif. BI implique l'organisation et l'analyse de données brutes pour obtenir de précieuses informations commerciales. C'est une pierre de Rosette qui traduit vos informations à partir de symboles insignifiants, de chaînes de zéros et

d'uns en une carte qui mène à un trésor d'affaires : de meilleures décisions, une plus grande efficacité et des profits plus élevés.

Les environnements d'analyses Big data ne visent pas à remplacer la BI / data warehouse traditionnels mais à les compléter, ils doivent être totalement intégrés en permettant de faire émerger des phénomènes depuis des données brutes.

2.3 Big Data Use Cases

2.3.1 Vue à 360 ° du client

De nombreuses entreprises utilisent le Big Data pour créer une application de tableau de bord offrant une vue à 360 ° du client. Ces tableaux de bord rassemblent des données provenant de diverses sources internes et externes, les analysent et les présentent au personnel du service clientèle, des ventes et / ou du marketing de manière à les aider à effectuer leur travail.

Toute cette information aiderait évidemment à préparer le personnel de l'entreprise à interagir avec le client, mais les tableaux de bord les plus sophistiqués ne s'arrêtent pas là. S'il utilise des outils d'analyse avancés ou d'apprentissage automatique, le tableau de bord laisse deviner le motif d'un appel client. Cela pourrait suggérer des opportunités de vente croisée ou de vente incitative de clients sur les produits, ou, s'il détectait qu'un client risquait de passer à un concurrent, il pourrait suggérer des remises potentielles pouvant abaisser le taux du client. Certains outils peuvent même analyser le langage des clients pour détecter leurs émotions actuelles et suggérer des réponses appropriées aux agents des ventes ou du service clientèle.

2.3.2 Optimisation des prix

Les entreprises grand public (B2C) et interentreprises (B2B) utilisent également l'analyse de données volumineuses pour optimiser les prix facturés à leurs clients. Pour toute entreprise, l'objectif est de fixer les prix de manière à maximiser leurs revenus. Si le prix est trop élevé, ils vendront moins de produits, ce qui réduira leur rendement net. Mais si le prix est trop bas, ils peuvent laisser de l'argent sur la table.

Les analyses de données volumineuses permettent aux entreprises de déterminer les prix les plus avantageux dans l'ensemble des conditions de marché historiques. Les entreprises plus sophistiquées en matière d'analyse des prix peuvent également utiliser des stratégies de prix variables ou dynamiques. Ils utilisent leurs solutions Big Data pour segmenter leur clientèle et créer des modèles qui montrent combien différents types de clients seront disposés à payer dans différentes circonstances. Les entreprises B2C qui ont essayé cette approche ont eu des résultats mitigés, mais il s'agit plutôt d'une norme parmi les entreprises B2B.

2.3.3 Reducing Customer Churn

La perte de clientèle fait partie intégrante du secteur des télécommunications, car le marché est beaucoup plus saturé aujourd'hui que par le passé. Dans ces conditions, le seul moyen d'augmenter les revenus consiste à créer davantage de services pour lesquels les clients paient davantage et à dépenser des sommes importantes pour que les clients changent d'opérateur.

<< It costs hundreds of dollars to acquire each new customer — so losing one costs carriers not only the expected revenue, but also the money they spent acquiring the customer. >>

2.3.4 Analyse et réponse des médias sociaux

Le flot de publications qui circulent dans les médias sociaux tels que Facebook, Twitter, Instagram et autres est l'un des exemples les plus évidents du Big Data. Aujourd'hui, les entreprises sont censées surveiller ce que les gens disent à leur sujet dans les médias sociaux et y répondre de manière appropriée. Sinon, elles perdent rapidement des clients.

En conséquence, de nombreuses entreprises investissent dans des outils pour les aider à surveiller et à analyser les plateformes sociales en temps réel. Parfois, il s'agit de produits de médias sociaux autonomes, alors que d'autres fois, ils font partie d'une solution plus vaste d'intelligence marketing ou d'analyse de données volumineuses.

2.3.5 Djezzy Use Cases (Triggers)

Actuellement les réseaux de télécoms acheminent le trafic numérique mondial, les opérateurs ont une position unique en termes de données.

l'Operateur Djezzy saisi des données granulaires sur les clients et leurs comportements, les expériences de service, les emplacements. En analysant toutes ces données ensemble - en une fois , ils peuvent obtenir des informations décisives pour un réel avantage concurrentiel.

A Djezzy, On appelle les Processus temps réel qui génèrent une action qui communique avec les clients : 'Trigger'.

Le Processus qu'on a réalisé est une partie intégrale dans le Trigger présenté dans la problématique. Mais il sert aussi d'un bas pour d'autres triggers utilisant les données de géolocalisation qui peuvent être réalisé.

Chez Djezzy il existe 7 Triggers en fonctionnement parmi eux :

- 70% Consommation Data
- 0% Crédit

Chapitre 3

Environnement technique

3.1 Environnement technique de Djezzy

Djezzy dispose de 40 serveurs dédier Big Data, en route de faire un Upgrade à 80 serveurs. L'environnement Logiciel utilise la plateforme Big-Data de Horton-Works , ou ils utilise des images d'HDF (Horton-Works Data Flow) et HDP (Horton-Works Data Platform), HDP est basée Sur Apache Hadoop, Apache Hive, Apache Spark, HDF est basée sur Apache NiFi, Apache Storm, Apache Kafka .

L'environnement total comprend :

1. Apache Hadoop
2. HDFS (Hadoop Distributed File System)
3. YARN (Hadoop Job Tracker)
4. Hadoop Map Reduce (Hadoop JobTracker , Gestion du execution de Map-Reduce)
5. Apache Hive (Un Data Warehouse crée au-dessus de Hadoop, il fournit une interface SQL pour interroger Hadoop).
6. Apache Spark 2 (Un Framework de calcule distribuer sur mémoire, comporte plusieurs outils, SQL, Machine Learning, Streaming, GraphX pour traiter les Graphs).
7. Apache Ambari (Plateforme de gestion et monitoring pour les applications Big Data)
8. Apache Kafka (Système de manipulation de flux de Données Temps Réel à latence faible)
9. Apache Strom (Système traitement de flux distribués)
10. Apache Zookeeper (Un logiciel de gestion de configuration pour systèmes distribués)
11. Apache Ranger (Un Framework permettant d'activer, de surveiller et de gérer une sécurité complète des données sur la plate-forme Hadoop)

12. Apache Knox (Une passerelle fournissant un point d'authentification et d'accès unique aux services Apache Hadoop dans un cluster)
13. Kerebros server (Un protocole d'authentification réseau qui repose sur un mécanisme de clés secrètes (chiffrement symétrique) et l'utilisation de tickets)
14. Apache Ignite (une base de Donnée In Memory distribuée, de mise en cache et de traitement distribuée open-source conçue pour stocker et calculer de gros volumes de données sur un cluster de nœuds)
15. Apache NiFi (Un système de gestion de flux de données. Il permet de gérer et d'automatiser des flux de données entre plusieurs systèmes informatiques, à partir d'une interface web et dans un environnement distribué.

3.2 Architecture générale Big data

Le processus Big-Data est utilisé se compose de 3 étape :

- **L'Ingestion** de donnée où on récupère les données des sources différentes on utilise apache NiFi et Apache Kafka dans cette étape.
- **Le Processing** où on transforme et agrège les données on Utilise Apache NiFi, Apache Spark, Apache Storm pour le Traitement et Apache Hive, Apache Ignite, Apache Cassandra pour le Stockage.
- **Le Reporting** où on génère des rapports et de chartes graphiques présentant les résultats qui peuvent être utilisé par management pour la prise de décisions.

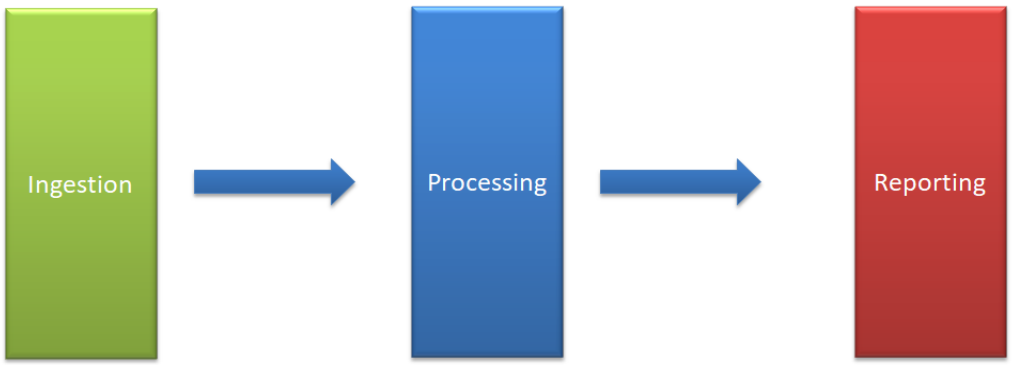

Figure 3.2: Architecture Big data

3.3 Architecture Big Data chez Djezzy

Djezzy ont organisé leur plateforme selon le plan dans la figure suivante pour l'ingestion des données et la DMP (Data Management Platform):

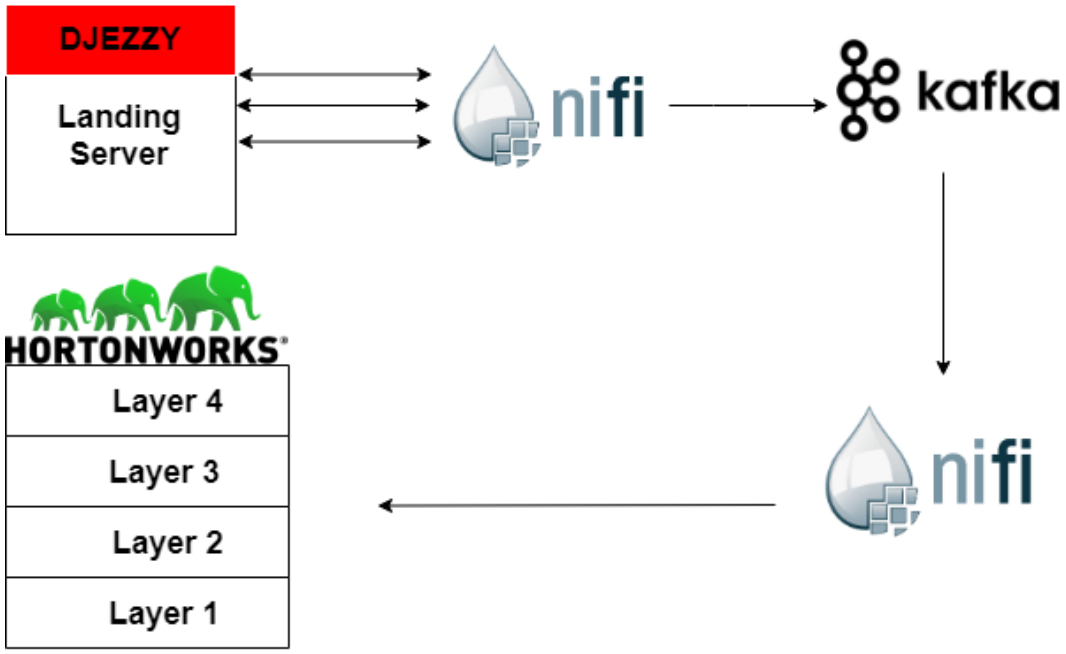

Figure 3.1: Djezzy Data Management Platform

Les différentes sources de données (Equipment, systèmes, CDRs) sont récupérées sur un seul serveur qu'ils appellent 'Landing Server' utilisant le Logiciel de NiFi. Puis ils utilisent puis sont publier sur Kafka. Ces Données sont consommées par les différentes applications est pipelines mis sur Hadoop et Hive. Ils traversent plusieurs couches de donnée où les couches supérieurs contient des données agrégée et transformer, pour les applications temps réel les donnée sont publier sur Kafka pour qu'ils soient récupérer par autres systèmes

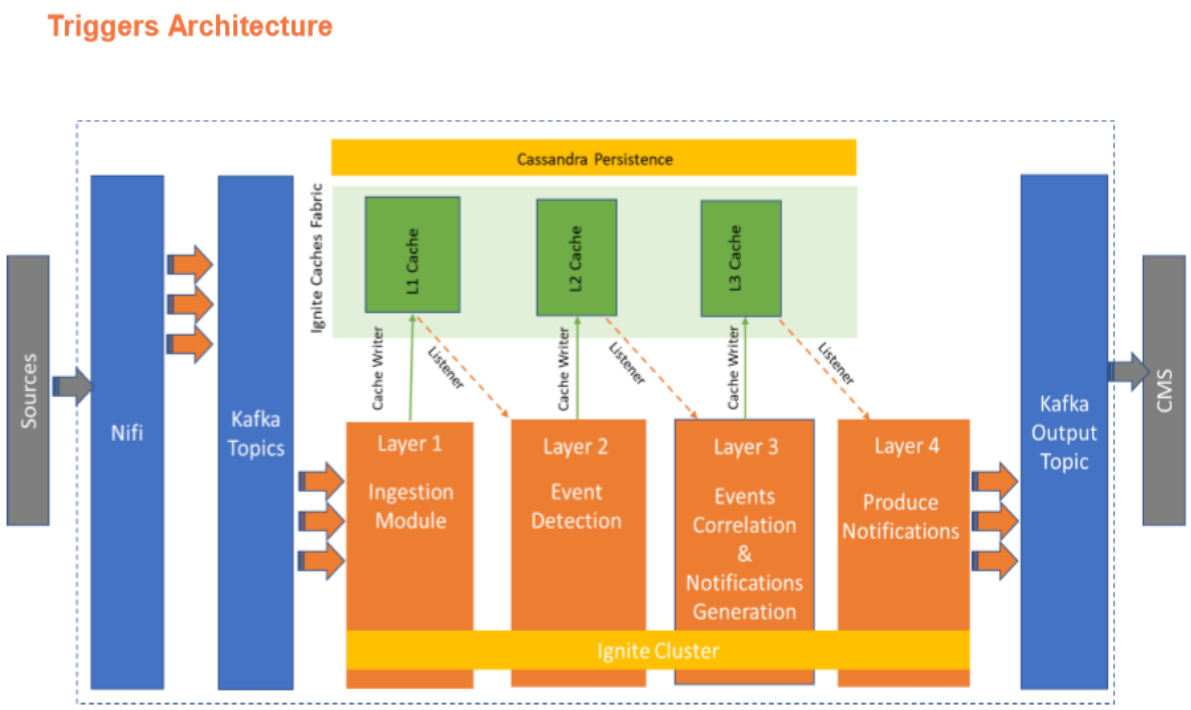

Figure 3.3: Triggers Architecture

Le System de Trigger Temps Réel suit cette Architecture :

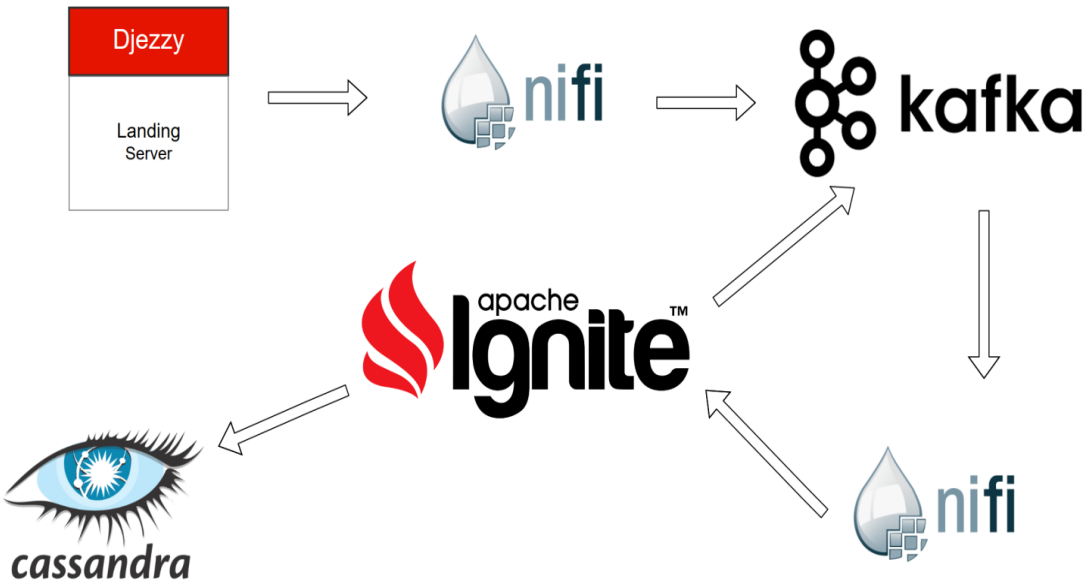

Figure 3.3: Triggers Architecture

Comme Le system de Batch Précèdent il intègre les Données utilisant NiFi et Kafka ces Données sont puis consommées une autre fois par NiFi transformées, agrégées , puis Stockées sur Apache Ignite Une BD in-Memory, la donnée pour un Access Rapide, ces Données sont persisté sur Apache Cassandra.

Chapitre 4

Réalisation du projet

4.1 Ecosystem Solution:

4.1.1 HortonWorks Data Platform:

HortonWorks Data Platform (HDP) est une plate-forme massivement évolutive et 100% open source pour le stockage, le traitement et l'analyse de gros volumes de données. C'est un acteur clé dans la transformation de Hadoop en un état «prêt à l'entreprise», ce qui stimulera encore davantage l'adoption par des organisations qui doutent actuellement des coûts informatiques liés à l'exploitation des clusters Hadoop..

• HortonWorks Data Platform fournit une plate-forme ouverte, stable et hautement extensible qui facilite l'intégration d'Apache Hadoop aux architectures de données existantes et optimise la valeur des données.

• L'architecture MapReduce de nouvelle génération (également appelée YARN) ajoute des avancées en matière d'évolutivité, de performances et de haute disponibilité, dissocie MapReduce de l'architecture de gestion des ressources et permet aux nouveaux types d'applications de s'intégrer à Hadoop, notamment le traitement de flux, le traitement des graphes, le traitement synchrone en masse et les messages. passer l'interface (MPI).

• HDFS Federation, qui permet aux nœuds Name de fonctionner de manière indépendante et sans coordination, la haute disponibilité du nœud de nom HDFS améliore l'intégrité des données et prend en charge plusieurs options de basculement..

• Pour les opérateurs, Pour explorer de nouveaux types de données et des ensembles de données volumineux qui étaient auparavant trop volumineux pour être capturés, stockés et traités, les analystes commerciaux utilisent largement HDP. Cela fournit des informations sur le flux de clics, la géolocalisation, le capteur, le journal du serveur, les données sociales, les données textuelles et vidéo.

4.1.2 Apache NiFi :

Apache NIFI prend en charge des graphiques dirigés puissants et évolutifs de logique de classement, de transformation et de médiation système. Parmi les fonctionnalités de haut niveau d'Apache NiFi, on peut citer une interface utilisateur Web. Expérience transparente entre la conception, le contrôle, le retour d'information et la surveillance, l'origine des données, SSL, SSH, HTTPS, le contenu crypté, etc., l'authentification / autorisation connectable basée sur des rôles. Apache NIFI est hautement configurable avec tolérance aux pertes et livraison garantie. faible latence vs débit élevé. la priorisation dynamique, le flux peut être modifié au moment de l'exécution.

Dans notre cas, On utilise Apache NiFi pour :

- lire les fichiers MSC Huawei en temps réel.
- Traiter les fichiers ,prendre chaque ligne individuellement .
- extraire les champs qui nous intéresses.
- transformer les champs en JSON.
- Crée une liaison avec Apache Kafka et Publier dans un Topic spécifique.

4.1.3 Apache Kafka:

Apache Kafka est un projet de courtier de messages à code source ouvert destiné à fournir une plate-forme unifiée, à haut débit et à faible temps de latence pour la gestion des flux de données en temps réel. Kafka est un service de journal de validation distribué, partitionné et répliqué. Il fournit les fonctionnalités d'un système de messagerie, mais avec un design unique. Kafka a une conception moderne, centrée sur les clusters, qui offre une forte durabilité et des garanties de tolérance aux pannes. Kafka est conçue pour permettre à un cluster unique de servir de base de données centralisée pour une grande entreprise. Il peut être étendu de manière transparente et élastique sans temps d'arrêt. Les flux de données sont partitionnés et répartis sur un groupe de machines afin de permettre des flux de données supérieurs à la capacité de n'importe quel ordinateur et de permettre à des groupes de consommateurs coordonnés.

Apache Kafka est largement adopté pour les cas d'utilisation allant de la collecte de données sur l'activité des utilisateurs, des journaux, des métriques d'application. Sa principale force réside dans sa capacité à rendre disponibles des volumes de données volumineux sous forme

de flux en temps réel destinés à être consommés dans des systèmes aux exigences très différentes, des systèmes par lots tels que Hadoop aux systèmes en temps réel nécessitant un accès à faible temps de latence, afin de gérer en continu les moteurs de traitement qui transforment le flux de données comme ils arrivent.

4.1.4 Apache Cassandra:

Apache Cassandra, un projet Apache de niveau supérieur né sur Facebook et basé sur Dynamo d'Amazon et BigTable de Google, est une base de données distribuée permettant de gérer de grandes quantités de données structurées sur de nombreux serveurs de base, tout en fournissant un service hautement disponible et sans point de défaillance unique. Apache Cassandra offre des fonctionnalités incomparables entre bases de données relationnelles et autres bases de données NoSQL, telles que: disponibilité continue, performances d'échelle linéaire, simplicité opérationnelle et distribution aisée des données sur plusieurs centres de données et zones de disponibilité dans le Cloud.

Relational Database	**NoSql Database**
Supports powerful query language.	Supports very simple query language.
It has a fixed schema.	No fixed schema.
Follows ACID (Atomicity, Consistency, Isolation, and Durability).	It is only "eventually consistent".
Supports transactions.	Does not support transactions.

Figure 4.1.4 : Relational vs NoSQL Database

Cassandra est devenue si populaire en raison de ses caractéristiques techniques exceptionnelles. Ci-dessous quelques-unes des fonctionnalités de Cassandra:

Elastic scalability : Cassandra est hautement évolutif; cela permet d'ajouter plus de matériel pour accueillir plus de clients et plus de données selon les besoins..

Always on architecture : Cassandra n'a pas de point de défaillance unique et est disponible en permanence pour les applications critiques pour l'entreprise qui ne peuvent se permettre une panne.

Fast linear-scale performance : Cassandra est évolutif de manière linéaire, c'est-à-dire qu'il augmente votre débit lorsque vous augmentez le nombre de nœuds dans le cluster. Par conséquent, il maintient un temps de réponse rapide.

Flexible data storage : Cassandra accepte tous les formats de données possibles, y compris: structuré, semi-structuré et non structuré. Il peut s'adapter de manière dynamique aux modifications apportées à vos structures de données en fonction de vos besoins.

Easy data distribution : Cassandra offre la possibilité de distribuer les données là où vous en avez besoin en les répliquant sur plusieurs centres de données.

Transaction support : Cassandra prend en charge des propriétés telles que l'atomicité, la cohérence, l'isolation et la durabilité (ACID).

Fast writes : Cassandra a été conçue pour fonctionner avec du matériel de base bon marché. Il effectue des écritures extrêmement rapides et peut stocker des centaines de téraoctets de données, sans sacrifier l'efficacité de la lecture.

4.1.5 Apache Ignite:

Apache Ignite est la principale plate-forme informatique en mémoire open source. Il s'agit d'une plate-forme en mémoire hautes performances, intégrée et distribuée, permettant de calculer et de traiter en temps réel des ensembles de données à grande échelle. Il exécute des ordres de grandeur plus rapidement que ce n'est possible avec les technologies traditionnelles à disque ou Flash. En tant que couche logicielle de gestion de données en mémoire, elle se situe entre les applications et diverses sources de données et ne nécessite pas d'extraire ou de remplacer les bases de données existantes.

Apache Ignite comprend, dans un cadre bien intégré, un ensemble de fonctionnalités clés en mémoire, notamment:

- An in-memory data grid
- An in-memory compute grid
- An in-memory service grid
- In-memory streaming processing
- In-memory acceleration for Hadoop

Ignite peut être exécuté en tant qu'application autonome ou au sein de l'application qui l'utilise. En tant qu'application autonome, il doit être exécuté en mode serveur. Cela signifie qu'il sera responsable du stockage et de la gestion des données. Cependant, lorsqu'il est exécuté dans une autre application, il peut être démarré en mode serveur ou client. En mode client, l'application n'est pas responsable du stockage ou de la transmission des données.

4.2 Problématique **Géo localisation des Clients Djezzy en Temps Réel**

L'Operateur Djezzy prend une position unique en termes de données. l'exploitation de ces données est nécessaires pour crée de nouvelle valeur dans le marché, ce qui nous mènes a l'exploitation et croisement de données marketing qui propose des offres aux clients et D'autres Tables contenants des données sur les cellules se trouvant dans les zones Economiques et Commerciales. afin de mieux cibler les utilisateurs sous forme de publicité géo-clôturée ou localisée.

4.3 Contribution Technique

Même avec tous ces processeurs et services de contrôleur disponibles dans Apache NiFi, il existe de nombreuses situations dans lesquelles on aura besoin d'1un service de processeur ou de contrôleur personnalisé

Notre contribution dans ce stage concerne le développement d'un Custom Processor sur l'outil Apache NiFi , afin de pouvoir écrire les données dans Apache Ignite .

Propriétés du Custom Processor:

Langage utilisé : Java

Le processeur a pour rôle de traiter un fichier JSON en entrée, et écrire les données dans une table de base de données avec les configurations suivante comme le montre la figure:

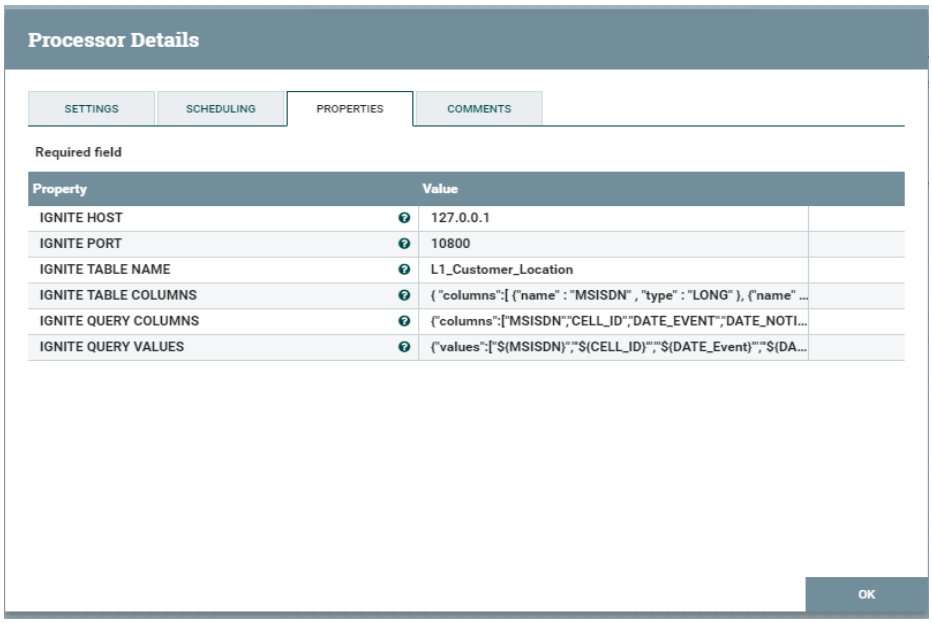

Figure 4.3 Processor Properties

4.3.1 Description technique du Trigger :

Les Donnée de Géo localisation qu'on a utilisés sont générer par les équipements de MSC-HUAWEI, cet équipement génère des CDR (Call Détail Record) lors d'un évènement d'appel voix ou envoi de SMS.

Les CDR fournie une localisation jusqu'au niveau de précision d'une cellule, elle ne fournit pas des coordonnées de longitude et latitude, mais ce niveau de précision est suffisant pour notre Use Cas.

Ces CDR sont envoyer sur le 'Landing Server' ou ils sont ingérer par un data flow sur NiFi, au niveau de ce data flow on devise les ficher on lignes individuelle et chaqu'une devient un message, on enrichit ces messages avec des Méta données (temps d'ingestion pour mesure de performance, id pour tracer les messages individuelles et un autre pour tracer le groupe de message 'Le Fichier').

Ces messages sont ensuit publier vers un 'Topic' sur Kafka, cela achevé la partie de l'ingestion des donnés.

Pour le Processing on a créé un autre data flow sur NiFi, il commence par consommer les donnée depuis le ‘Topic’ Kafka créé, puis on extrait les informations nécessaires depuis les flow file :

- Le MSCSDN (représente le numéro de l’abonné)
- Le Cell_Id (représente un identifiant de la cellule du réseau mobile)
- La date d’évènement (représente le temps où l’évènement s’est produit)

On ajoute le temps où on va écrire ces données sur La BD Ignite (Elle pour mesurer les performances en la comparant avec le temps d’ingestion).

Ces données sont ensuite écrites sur la base de données Ignite utilisant un custom process. Et Persisté sur la Base de données Apache Cassandra.

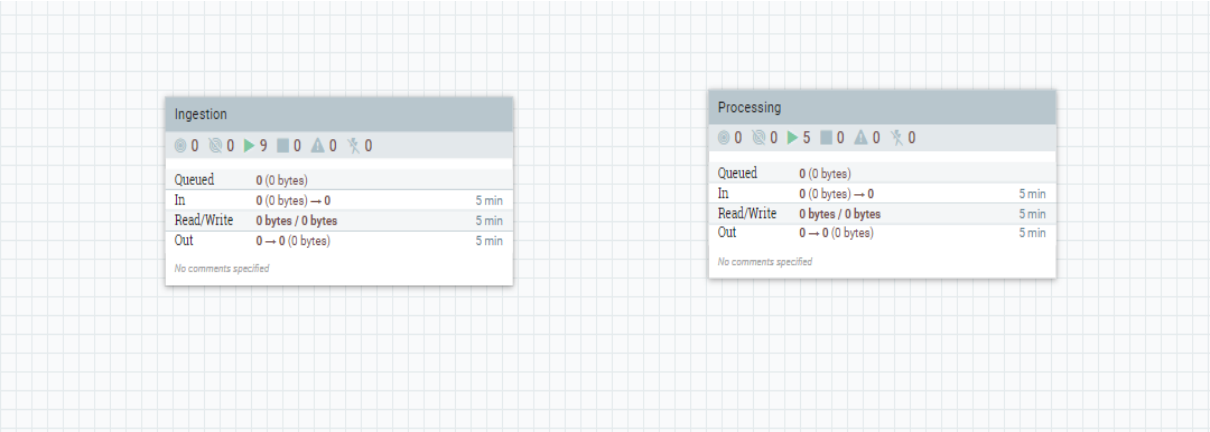

Figure 4.3.1 : NiFi trigger Overview

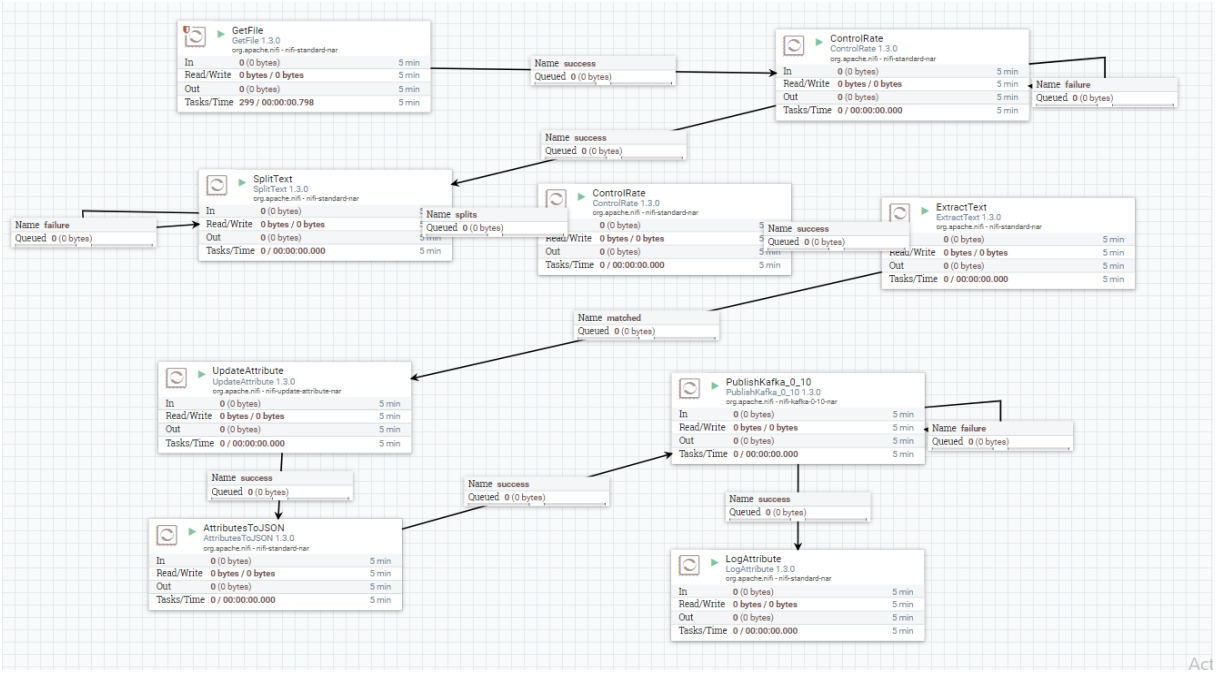

Figure 4.3.1(a) : Ingestion Process

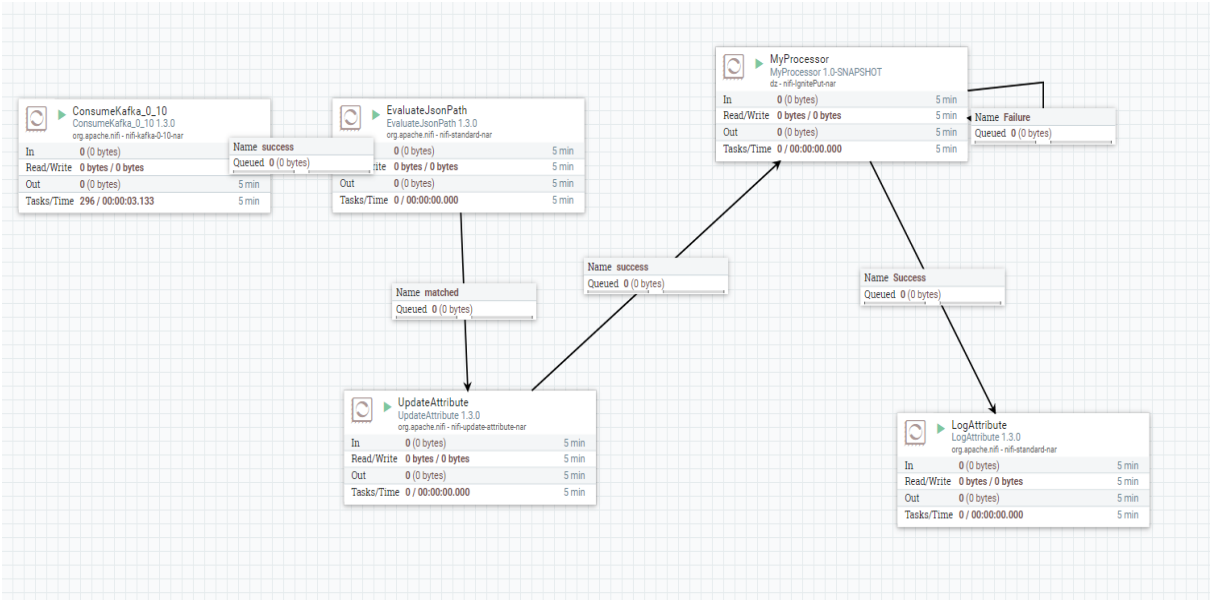

Figure 4.3.1(b) : Processing Process