Université Mohammed V - Rabat

Faculté des sciences Rabat

Master Traitement Intelligente des Systèmes

Rapport de Projet du BIG DATA

Par

Abouche Hind

Spark Streaming : Analyse de sentiments sur Twitter

Table des matières

Chapitre 1

Introduction au Big Data

Plan

Introduction

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

1.1 Qu'est-ce que le Big Data

— Big Data est un terme générique qui désigne les datasets qui ne peuvent pas être gérés par des serveurs et des outils classiques en raison de leur volume, de leur vélocité et de leur variété. Le concept de Big Data fait aussi référence aux technologies et aux stratégies mises en œuvre pour gérer ce type de données.

— Le Big Data peut être défini comme une collection de données qui peuvent être structurées ou non structurées et si grandes et compliquées qu'il peut devenir complexe de les traiter à l'aide de systèmes simples ou d'applications de traitement de données traditionnelles.

- Volume(Volume) : Un système Big Data se caractérise d'abord par le volume de données en jeu. Un système Big Data traite un volume de données largement supérieur à ce que traitent les bases de données traditionnelles. Ce qui pose un défi technologique.
- Variété(Variety) :Les données sont en grand nombre et circulent vite dans le système. Mais ce n'est pas tout. Le Big Data se caractérise aussi par l'immense variété des données traitées. Dans le Big Data, les données sont dans leur majorité non-structurée ou semi-structurée. Et pour cette raison, elles doivent être travaillées, longuement préparées.
- Vitesse(Velocity) :La vitesse pour générer de nouvelles données et se déplacer est indiquée comme Velocity. Par exemple, les messages ou vidéos sur les réseaux sociaux qui deviennent viraux en une minute et détectent rapidement l'activité suspecte sur les transactions par carte de crédit. Les systèmes conçus pour gérer les mégadonnées peuvent gérer et traiter des millions de lignes

par seconde, ce qui facilite l'obtention des informations souhaitées à partir de ces données.

- Véracité (Veracity) : La variété des sources et la complexité des traitements peuvent poser des problèmes en ce qui concerne l'évaluation de la qualité des données. La problématique de la Data Quality est structurante dans n'importe quel projet Big Data.
- Valeur(Value) : Le défi ultime du Big Data est de créer de la valeur. Or, parfois, les systèmes et les procédures en place sont si complexes qu'il devient difficile d'extraire de la valeur des données à disposition (d'en dégager des insights). La valeur rappelle la finalité business de tout projet Big Data.

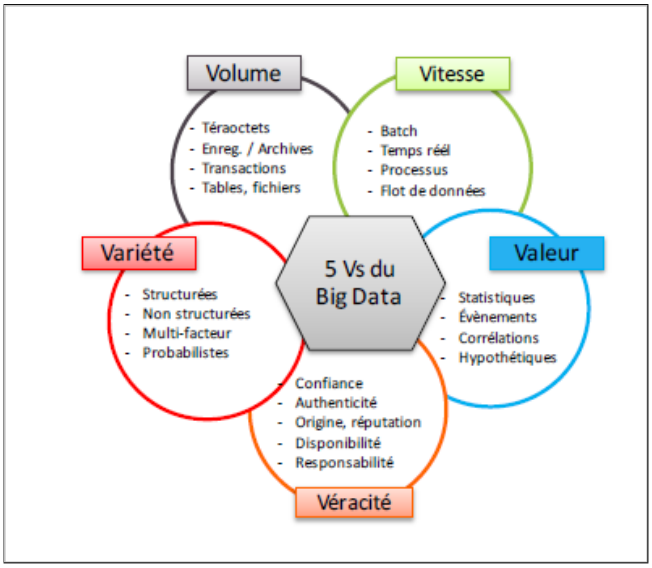


Figure 1.1: les 5 Vs de BIG DATA

1.2 Sources of Big Data

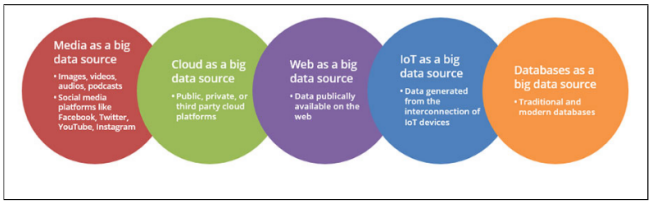


Figure 1.2: les sources de BIG DATA

Le Big Data s'appuie sur cinq sources de données :

- Media as a Big data source : Les médias sont la source la plus populaire de mégadonnées, car ils fournissent des informations précieuses sur les préférences des consommateurs et l'évolution des tendances. Les médias incluent les médias sociaux et les plateformes interactives, comme Google, Facebook, Twitter, YouTube, Instagram, ainsi que les médias génériques comme les images, les vidéos, les audios etc

- Cloud as a Big data source : Aujourd'hui, les entreprises ont devancé les sources de données traditionnelles en déplaçant leurs données sur le Cloud. Le stockage Cloud héberge des données structurées et non structurées et fournit aux entreprises des informations en temps réel et des informations à la demande. Le principal attribut du Cloud computing est sa flexibilité et son évolutivité.
- Web as a Big data source : Le web public constitue un Big data largement répandu et facilement accessible. Les données sur le Web ou «Internet» sont couramment disponibles pour les particuliers et les entreprises. De plus, les services Web tels que Wikipédia fournissent des informations gratuites et rapides à tout le monde.
- IOT As a Big data source : Le contenu généré par la machine ou les données créées à partir de l'IoT constituent une source précieuse de données volumineuses. Ces données sont généralement générées à partir des capteurs connectés à des appareils électroniques. La capacité d'approvisionnement dépend de la capacité des capteurs à fournir des informations précises en temps réel.
- Database as a big data source : Les entreprises préfèrent aujourd'hui utiliser une fusion de bases de données traditionnelles et modernes pour acquérir des mégadonnées pertinentes. Cette intégration ouvre la voie à un modèle de données hybride et nécessite de faibles investissements et des coûts d'infrastructure informatique.

1.3 Pourquoi avons-nous besoin du Big Data ?

- Le Big Data représente une grande quantité d'ensembles de données qui est collectée et stockée pour être analysée afin que les entreprises et les organisations puissent les utiliser pour prendre de meilleures décisions et améliorer les affaires.
- La clé du succès avec le Big Data ne réside pas dans la quantité de données collectées par une entreprise, mais dans la capacité de traitement et la manière dont elle utilise réellement ces données collectées.
- Le Big Data est le terme désignant des ensembles de données volumineux et complexes qu'il devient difficile pour l'entrepôt de données traditionnel de stocker, d'analyser, de gérer, de traiter et de travailler dessus et de visualiser.

1.4 Qu'est-ce que le Big Data Analytics ?

- Le Big Data Analytics est le processus d'analyser de grands ensembles de données contenant une variété de types de données, comme les mégadonnées, pour découvrir des modèles cachés, des corrélations inconnues, les tendances du marché, les préférences des clients et d'autres informations

commerciales utiles.

- Grâce à l'analyse de Big Data, des informations importantes peuvent être obtenues. Pour ce projet, nous utilisons Big Data Analytic pour obtenir des informations à partir des données Twitter.

Conclusion

Dans ce chapitre, nous avons essayé de parler en général du big data, d'où cela vient-il, des caractéristiques du big data et enfin pourquoi nous avons besoin du big data dans les entreprises.

Chapitre 2

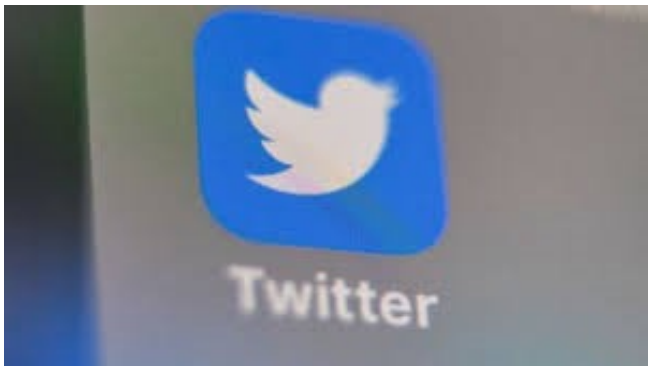
APERÇU DES TECHNOLOGIES ET DE LA PLATEFORME UTILISÉES

Plan

Introduction

Afin de réaliser Notre projet nous avons utilisé quelques technologies et plateformes,et c'est ce qu'on va voir dans le chapitre suivant :

2.1 Twitter :



- Twitter est un réseau social de microblogage géré par l'entreprise Twitter Inc. Il permet à un utilisateur d'envoyer gratuitement de brefs messages, appelés Tweets, sur internet, par messagerie instantanée ou par SMS. Ces messages sont limités à 280 caractères (140 caractères jusqu'en novembre 2017). les gens peuvent se suivre pour recevoir les Tweets d'une autre personne. Ils peuvent en étiqueter d'autres à l'aide du symbole @.
- Comme tous les forums de médias sociaux, ces Tweets peuvent inclure des URL, des photographies et des Hashtags. Une personne peut partager le Tweet d'une autre personne (re_tweet) si elle suit la dernière. Hashtag (#) est utilisé comme pionnier dans Twitter et permet de rechercher facilement des informations sur un sujet particulier lors de la collecte de données sur Twitter.
- Twitter compte plus d'un milliard de comptes d'utilisateurs enregistrés et environ 317 millions d'utilisateurs Twitter actifs par mois. Il contient une énorme quantité de données et comprenait des utilisateurs de tous les domaines comme les stars de cinéma, les critiques de marques, les sportifs, les gens ordinaires, les politiciens, etc.

2.2 Apache Spark

- Apache Spark est un moteur de traitement de données rapide dédié au Big Data. Il permet d'effectuer un traitement de larges volumes de données de manière distribuée.Très en vogue depuis maintenant quelques années, ce Framework est en passe de remplacer Hadoop.
- Son principal avantage est sa vitesse, puisqu'il permet de lancer des programmes 100 fois plus rapidement que Hadoop MapReduce in-memory, et 10 fois plus vite sur disque. Son moteur

d'exécution DAG avancé supporte le flux de données acyclique et le computing in-memory. Il est également facile à utiliser, et permet de développer des applications en Java, Scala, Python et R. Son modèle de programmation est plus simple que celui d' Hadoop.

- Spark utilise RDD (resilient distributed dataset) pour distribuer les éléments sur le cluster. Outre l'étincelle de RDD, il existe également des DataFrames qui sont des structures de type table utilisées par Spark pour stocker des données au format tableau. Les RDD sont les éléments constitutifs de toute application Spark.

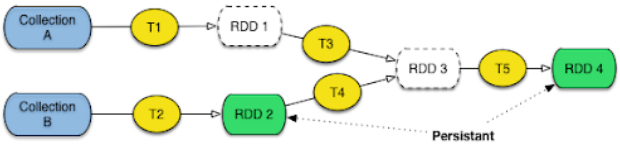


Figure 2.1: RDD persistants et transitoires dans Spark

- Les RDD représentent les collections obtenues au cours des différentes étapes d'une chaîne de traitement.La différence essentielle est que dans Spark, les RDD peuvent être marquées comme étant persistants car ils peuvent être réutilisés dans d'autres chaînes. Spark fait son possible pour stocker un RDD persistant en mémoire RAM, pour un maximum d'efficacité.

2.2.0.1 Caractéristiques de Spark

Apache Spark est une open source cluster computing framework pour le traitement des données en temps réel. La principale caractéristique d'Apache Spark est son calcul en cluster en mémoire qui augmente la vitesse de traitement d'une application :

- Vitesse(Speed) : Spark fonctionne jusqu'à 100 fois plus rapidement que Hadoop MapReduce pour le traitement de données à grande échelle. Il est également capable d'atteindre cette vitesse grâce à un partitionnement contrôlé.
- Mise en cache puissante (Powerful Caching) : une couche de programmation simple offre de puissantes capacités de mise en cache et de persistance du disque.

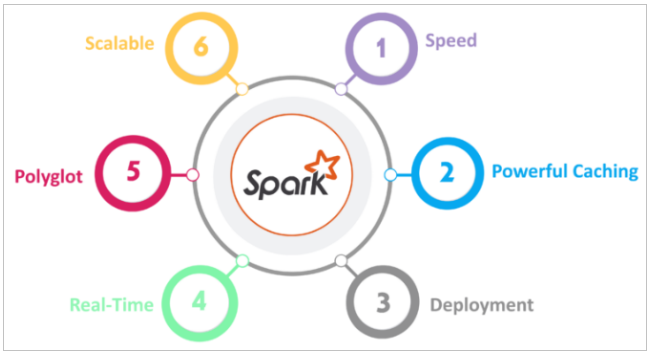


Figure 2.2: les Caractéristiques de Spark

- Déploiement(Deployment) : il peut être déployé via Mesos, Hadoop via YARN ou le propre gestionnaire de cluster de Spark.
- Temps réel (Real-Time) : il offre un calcul en temps réel et une faible latence en raison du calcul en mémoire.
- Polyglot : Spark fournit des API de haut niveau en Java, Scala, Python et R. Le code Spark peut être écrit dans l'un de ces quatre langages. Il fournit également un shell en Scala et Python.

2.2.0.2 Fonctionnement de l'architecture Apache Spark

La structure de Spark peut être comprise à partir du diagramme ci-dessous :

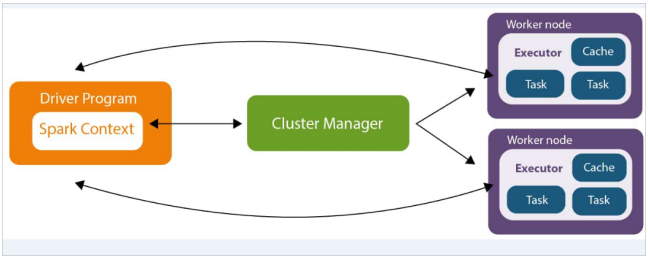


Figure 2.3: Architecture Spark

- Driver Program : gère l'allocation des Tasks aux Worker nodes. Il est également appelé Master. Le Driver fait la tâche d'écouter les Worker nodes pour d'éventuels messages entrants. Il garde également la tâche isolée afin qu'il n'y ait pas de fuite de données entre les tâches.
- Cluster Manager : distribue la tâche à différents Worker nodes. C'est comme l'intermédiaire entre le Driver program et les Worker nodes qui gère le cluster lors de sa distribution.
- Worker node : chaque Worker node a un exécuteur qui peut effectuer de nombreuses tâches. Chaque Worker dispose d'une mémoire cache allouée qui est configurable. Chaque executer teste la tâche afin qu'il n'y ait aucune fuite de mémoire entre les multiples tâches soumises. La seule façon de partager la mémoire entre deux tâches est de l'écrire dans une mémoire externe.
- Executor est le processus initié pour l'exécution d'une application. Chaque application exécutée

sur le cluster a ses propres exécuteurs. L'Executer est responsable de la conservation des données et des opérations d'entrée-sortie entre les applications.

- Task : est une unité de travail assignée par un Executer

Un Spark Cluster peut être configuré avec différents réglages de paramètres. Le nombre d'Executerr, de Worker nodes, d'allocation de mémoire, de mémoire cache, de Worker core, d'Executer core, tout peut être configuré en fonction des exigences du système.

Apache spark fournit diverses autres fonctionnalités comme :

2.2.1 Spark Streaming :

- Spark Streaming est la partie traitement en pseudo temps réel d'Apache Spark. La brique Streaming est basé sur Spark Core et traite la donnée sous forme de mini-batchs espacés par un instant T.
- Spark streaming utilise « discretized streams (DStreams) », qui est exprimé par une séquence de RDDs et représente un flux continu de données (c'est-à-dire une série continue de RDD) mais les traitements qui seront faits sur les Dstreams sont identiques à ceux de Spark Core classique.
- Spark Streaming est un module complémentaire de l'API Spark de base qui est évolutif et à haut débit. Il offre des fonctionnalités de traitement des données en direct ruisseaux. Pour le streaming, Apache Spark peut avoir un canal, HDFS, apache Kafka, Twitter, sources de données kinesis. Ces données peuvent ensuite être nettoyées et structurées dans l'étincelle elle-même et utilisées pour poursuivre le traitement.

Concrètement, Spark streaming utilise des micro-batchs. Cela signifie que, Spark streaming reçoit les données et les divise en plusieurs mini batch RDDs qui sont à leur tour traités par « Spark Engine » pour générer le flux des résultats en batch

2.2.2 Spark MLlib

C'est une bibliothèque d'apprentissage automatique, apparu dans la version 1.2 de Spark, qui contient tous les algorithmes et utilitaires d'apprentissage classiques, comme la classification, la régression, le clustering, le filtrage collaboratif et la réduction de dimensions, en plus des primitives d'optimisation sous-jacentes. On parle souvent d'une fouille de données par apprentissage statistique.

Conclusion

Dans ce deuxième chapitre ,on a parlé sur les utiles qu'on a utilisé dans notre projet de spark streaming, dans le dernier chapitre nous allons discuter la partie de la réalisation du projet Conclusion partielle ayant pour objectif de synthétiser le chapitre et d'annoncer le chapitre suivant.

Chapitre 3

Implémentation Twitter-Sentiment-Analysis

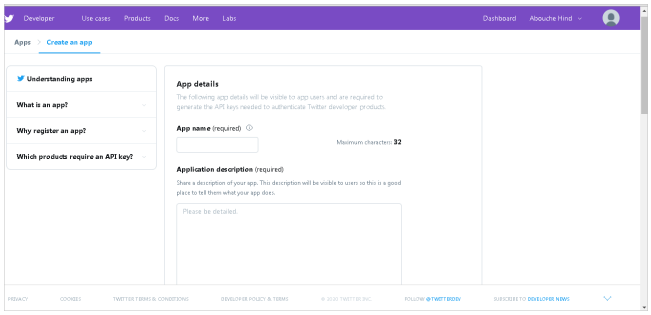
Introduction

dans ce chapitre ,on va découvrir les étapes pour utiliser Spark Streaming afin d'analyser les sentiments des tweets 8

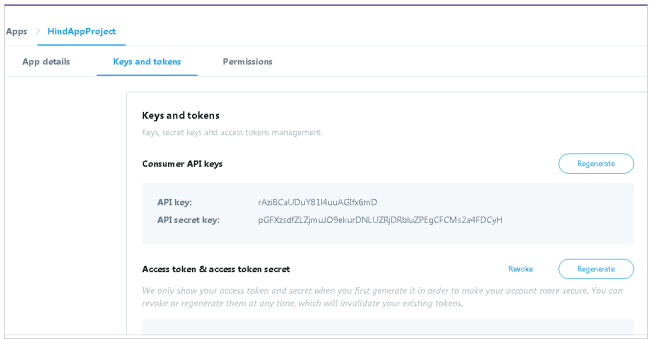
3.1 Création de nos propres informations d'identification pour les API Twitter

Dans notre projet, Nous utilisons Spark Streaming pour diffuser des données de Spark en direct. Pour charger des données Twitter dans Apache Spark, Twitter fournit aux développeurs une interface qui peut être utilisée pour accéder aux données de Twitter.

Tout d'abord, nous connectons à partir de notre compte Twitter et passons aux applications Twitter, pour créer une nouvelle application afin d'obtenir les clés : Consumer Key, Consumer Secret, Access Token and Access Token Secret (on va les utiliser pour connecter avec notre Twitter API). Apres la création de notre



application, on peut accéder à Keys and Token, pour obtenir nos clés pour les utiliser dans la connexion avec notre Twitter API.



3.2 Comment utilisons-nous l'analyse des sentiments Twitter en Python :

- Matplotlib : est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques.
- Tweepy : est le client python de l'API Twitter officielle.
- TextBlob : est la bibliothèque python pour le traitement des données textuelles.

Ce script peut vous dire les sentiments des gens concernant tous les événements qui se produisent dans le monde en analysant les Tweets liés à cet événement. Il recherchera des Tweets sur n'importe quel sujet et analysera chaque Tweet pour voir à quel point son émotion est positive ou négative.

- Importer les libraries et Authentification :

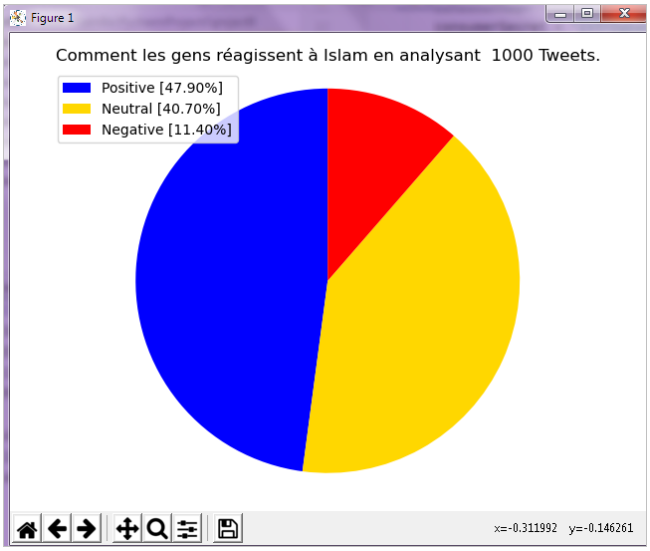
```
# ******************** créer par ************************
################ HIND ABOUCHE ************************
#*************BIG DATA*********************************
#************2020*************************************
import tweepy,csv,re
from textblob import TextBlob
import matplotlib.pyplot as pl
class SentimentAnalysis:
    def __init__(self):
        self.tweets = []
        self.tweetText = []
    def get_tweets(self):
        consumerKey = "WDnCulflaJEnf7YYHl7N5HMMP"
        consumerSecret = "ZV5YDgbaG9aaN3QbXt1nHjYTLKBtFBfAqDamOG2w9veECt6AML"
        accessToken = "1270269131506548736-EDwf471vKNQ7U1sEU1PhJJYffNJ0qU"
        accessTokenSecret = "WAoIYtbTjX1U01GksxmcIdTAtqAglhH5MAWkMixXgfCwf"
        #etablir la connection avec our API
        auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
        auth.set_access_token(accessToken, accessTokenSecret)
        api = tweepy.API(auth)
```

- Les résultats d'analyse : :

```
Run:  spark_stream
C:\Users\ToshiBa\PycharmProjects\project6\venv\Scripts\python.exe C:/Users/ToshiBa/PycharmProjects
Saisissez un tag pour rechercher SVP :Islam
Entrez le nombre de tweets à rechercher SVP: 1000
Comment les gens réagissent à Islam en analysant  1000 tweets.
Positive

47.90% des gens pensaient que c'était positif
11.40% des gens pensaient que c'était negative
40.70% des gens pensaient que c'était neutral

6: TODO  4: Run  Terminal  Python Console
```



Conclusion

le script qu'on a réalisé est un script simple contient quleques fonctions (gettweets() pour obtenir les tweets ,percentage() pour calculer le pourcentge et la fonction plot() pour visualizer les resultats

Conclusion générale

Tout en travaillant sur ce projet, J'ai appris beaucoup de technologies en développant mes connaissances dans le domaine d'Apache Spark, même si j'ai rencontré beaucoup de problèmes pendant le travail sur ce projet, comme la configuration de Spark, la compatibilité des versions, problèmes de Path et variables d'environnements.

Enfin, mais non le moindre, j'ai appris à surmonter les problèmes. L'expérience m'a fait techniquement qualifiée et développé mon pouvoir de réflexion autour des problèmes.L'expérience m'a fait techniquement qualifié et développé mon pouvoir de réflexion autour des problèmes