

Rapport du TP BDA

2 ème année Cycle Supérieur (2CS)

Option : Système Informatique (SQ)

Thème :

Analyse de données Web avec Hadoop

Réalisé par :

- Benameur Tarek
- Touil Nihel

Table des matières

1. Introduction.....	4
1.1 Contexte du projet.....	4
1.2 Objectifs du TP.....	4
1.3 Méthodologie.....	4
2. Présentation des outils utilisés.....	4
2.1 Apache Hadoop.....	4
Avantages de Hadoop :.....	4
2.2 HDFS (Hadoop Distributed File System).....	5
2.3 MapReduce.....	5
2.4 Apache Hive.....	5
3. Architecture et environnement de travail.....	5
3.1 Configuration matérielle.....	5
3.2 Mode pseudo-distribué.....	6
3.3 Dataset utilisé.....	6
4. Mise en Œuvre.....	7
4.1 Préparation de l'environnement.....	7
4.2 Installation et configuration d'Hadoop.....	7
4.3 Initialisation du cluster.....	8
4.4 Gestion des données.....	8
4.5 Développement MapReduce.....	8
4.6 Intégration Apache Hive.....	9
7.5 Résultats des requêtes Hive.....	9
8. Résultats et analyse des données.....	9
8.1 Synthèse des résultats obtenus.....	9
8.2 Analyse comparative MapReduce vs Hive.....	10
8.3 Insights tirés des données COVID-19.....	11
9. Conclusion et perspectives.....	11
Références :.....	11

1. Introduction

1.1 Contexte du projet

Le Big Data représente aujourd'hui un enjeu majeur pour les entreprises et organisations qui cherchent à extraire de la valeur de volumes massifs de données. Dans un monde où les réseaux sociaux génèrent des millions de messages quotidiennement, l'analyse de ces données devient cruciale pour comprendre les tendances, les opinions publiques et les comportements sociétaux.

1.2 Objectifs du TP

Ce travail pratique vise à mettre en œuvre une chaîne complète de traitement de Big Data en utilisant l'écosystème Apache Hadoop. Les objectifs spécifiques sont :

- Installer et configurer un cluster Hadoop en mode pseudo-distribué
- Manipuler le système de fichiers distribué HDFS
- Développer et exécuter des jobs MapReduce
- Utiliser Apache Hive pour des analyses SQL sur de gros volumes de données
- Analyser des données réelles de tweets liées à la pandémie COVID-19

1.3 Méthodologie

Notre approche suit une démarche structurée en plusieurs étapes : installation de l'environnement, préparation des données, traitement par MapReduce, analyse avec Hive, et enfin interprétation des résultats obtenus.

2. Présentation des outils utilisés

2.1 Apache Hadoop

Apache Hadoop est un framework open-source conçu pour le stockage et le traitement distribué de très gros volumes de données sur des clusters de machines. Il repose sur deux composants principaux :

Avantages de Hadoop :

- Scalabilité horizontale
- Tolérance aux pannes
- Traitement parallèle
- Coût réduit (utilisation de matériel standard)

Architecture Hadoop :

- **HDFS (Hadoop Distributed File System)** : Système de fichiers distribué
- **YARN (Yet Another Resource Negotiator)** : Gestionnaire de ressources
- **MapReduce** : Framework de programmation pour le traitement parallèle

2.2 HDFS (Hadoop Distributed File System)

HDFS est conçu pour stocker de très gros fichiers sur plusieurs machines avec une haute tolérance aux pannes. Il utilise une architecture maître-esclave avec :

- **NameNode** : Serveur maître qui gère les métadonnées
- **DataNodes** : Serveurs esclaves qui stockent les données réelles
- **Réplication** : Chaque bloc est répliqué sur plusieurs nœuds (par défaut 3)

2.3 MapReduce

MapReduce est un modèle de programmation qui permet de traiter de gros volumes de données en parallèle. Il se compose de deux phases principales :

- **Map** : Traitement et transformation des données d'entrée
- **Reduce** : Agrégation des résultats intermédiaires

2.4 Apache Hive

Apache Hive est un entrepôt de données construit sur Hadoop qui facilite la lecture, l'écriture et la gestion de gros datasets. Il offre :

- **HiveQL** : Langage de requête similaire à SQL
- Intégration transparente avec HDFS et MapReduce
- **Métastore** : Stockage des métadonnées des tables

3. Architecture et environnement de travail

3.1 Configuration matérielle

- Processeur : 4 cœurs
- RAM : 4 Go
- Stockage : 30 Go d'espace libre
- Système d'exploitation : Linux Ubuntu

3.2 Mode pseudo-distribué

Le mode pseudo-distribué simule un cluster multi-nœuds sur une seule machine. Tous les démons Hadoop s'exécutent sur la même machine mais dans des processus Java séparés.

Démons Hadoop en mode pseudo-distribué :

- NameNode
- DataNode
- ResourceManager
- NodeManager

3.3 Dataset utilisé

Source : Kaggle - COVID19 Tweets Dataset (<https://www.kaggle.com/datasets/gpreda/covid19-tweets/data>)

Format : CSV

Taille du dataset : 179,108 tweets (lignes)

Période couverte : Mars à juillet 2020

Colonnes du dataset :

- **user_name** : Nom affiché de l'utilisateur (nom d'affichage, pas nom d'utilisateur)
- **user_location** : Lieu renseigné par l'utilisateur
- **user_description** : Description bio de l'utilisateur
- **user_created** : Date de création du compte utilisateur
- **user_followers** : Nombre d'abonnés
- **user_friends** : Nombre de comptes suivis
- **user_favourites** : Nombre de tweets aimés
- **user_verified** : Statut de vérification du compte
- **date** : Date et heure du tweet
- **text** : Contenu textuel du tweet
- **hashtags** : Hashtags présents dans le tweet (sous forme de liste)
- **source** : Client utilisé pour poster (Twitter for iPhone, Web App, etc.)
- **is_retweet** : Indique si le tweet est un retweet

Caractéristiques du dataset :

- Volume important de tweets liés au COVID-19
- Données temporelles permettant l'analyse des tendances
- Informations utilisateurs pour des analyses sociologiques
- Texte brut permettant l'analyse de contenu et de sentiment

4. Mise en Œuvre

4.1 Préparation de l'environnement

La première phase consiste à préparer l'environnement système pour accueillir le cluster Hadoop. Cette étape cruciale comprend l'installation de Java 8, prérequis indispensable pour l'exécution d'Hadoop, ainsi que la configuration des variables d'environnement.

La mise en place de l'authentification SSH permettant la communication fluide entre les différents composants du cluster.

Cette configuration implique la génération de clés SSH et leur autorisation pour les connexions locales, garantissant ainsi le bon fonctionnement du mode pseudo-distribué.

4.2 Installation et configuration d'Hadoop

L'installation d'Hadoop 3.3.6 s'effectue par téléchargement direct depuis les dépôts officiels Apache, suivi de l'extraction dans le répertoire `/opt/hadoop` et de la configuration des droits d'accès appropriés. Cette version offre une stabilité éprouvée et une compatibilité optimale avec les outils de l'écosystème.

La configuration du cluster nécessite la modification de plusieurs fichiers XML stratégiques :

- `core-site.xml` définit le système de fichiers par défaut et les répertoires temporaires
- `hdfs-site.xml` configure les paramètres HDFS incluant la réplication et l'emplacement des données
- `mapred-site.xml` établit la liaison entre MapReduce et YARN comme gestionnaire de ressources
- `yarn-site.xml` paramètre le gestionnaire de ressources et les services auxiliaires

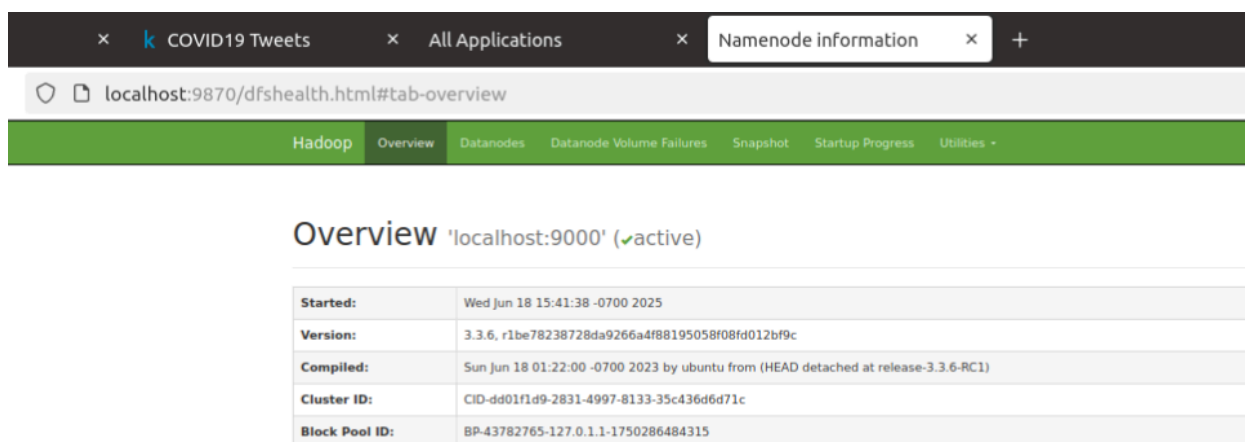
L'intégration des variables d'environnement Hadoop dans le profil utilisateur assure l'accès global aux commandes et outils de l'écosystème depuis n'importe quel répertoire du système.

4.3 Initialisation du cluster

Le formatage du NameNode constitue l'étape d'initialisation critique qui prépare le système de fichiers HDFS. Cette opération, effectuée une seule fois lors de la première installation, initialise les structures de métadonnées nécessaires au fonctionnement distribué.

Le démarrage séquentiel des services suit un ordre précis : d'abord HDFS avec les composants NameNode et DataNode, puis YARN avec ResourceManager et NodeManager. La vérification de l'état du cluster via la commande `jps` confirme l'exécution correcte de tous les processus requis.

Les interfaces web deviennent alors accessibles : le NameNode sur le port 9870 pour le monitoring HDFS, et le ResourceManager sur le port 8088 pour le suivi des applications YARN.



The screenshot shows a web browser window with three tabs: "COVID19 Tweets", "All Applications", and "Namenode information". The address bar shows `localhost:9870/dfshealth.html#tab-overview`. The page has a green header with navigation links: "Hadoop", "Overview", "Datanodes", "Datanode Volume Failures", "Snapshot", "Startup Progress", and "Utilities". The main content area is titled "Overview 'localhost:9000' (✓active)". Below the title is a table with the following information:

Started:	Wed Jun 18 15:41:38 -0700 2025
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 01:22:00 -0700 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-dd01f1d9-2831-4997-8133-35c436d6d71c
Block Pool ID:	BP-43782765-127.0.1.1-1750286484315

Figure 4. Hadoop cluster management interface

4.4 Gestion des données

La récupération du dataset s'appuie sur l'API Kaggle, nécessitant une configuration préalable des credentials d'authentification. Le dataset "COVID-19 Tweets" est téléchargé automatiquement, décompressé et préparé pour le transfert vers HDFS.

Le chargement des données vers le système de fichiers distribué utilise les commandes `hdfs dfs` pour copier efficacement les fichiers CSV du système local vers HDFS. La création de l'arborescence de répertoires suit une organisation logique : `/user/$USER/data/covid_tweets/` pour structurer et faciliter l'accès aux données.

La vérification du transfert inclut la validation de l'intégrité des données, la vérification des tailles de fichiers et l'examen des premiers enregistrements pour s'assurer de la cohérence du dataset.

4.5 Développement MapReduce

L'application MapReduce développée analyse les tweets selon plusieurs dimensions analytiques pertinentes. Le Mapper traite chaque ligne du dataset CSV en extrayant les informations clés : hashtags spécifiques au COVID-19, mots-clés significatifs (vaccin, masque, hôpital), classification des retweets, et identification des comptes vérifiés.

L'architecture de l'application repose sur un parsing CSV robuste gérant les guillemets et caractères d'échappement. Le Reducer agrège les comptages par catégorie, produisant des statistiques consolidées sur l'ensemble du dataset.

Le processus de compilation utilise le classpath Hadoop pour intégrer les bibliothèques nécessaires. La création du fichier JAR permet l'exécution distribuée sur le cluster, avec un monitoring en temps réel via l'interface YARN.

4.6 Intégration Apache Hive

L'installation d'Apache Hive 3.1.3 enrichit le cluster avec des capacités d'analyse SQL avancées. La configuration inclut l'initialisation du schéma de métadonnées avec Derby comme base de données embarquée, solution adaptée au mode pseudo-distribué.

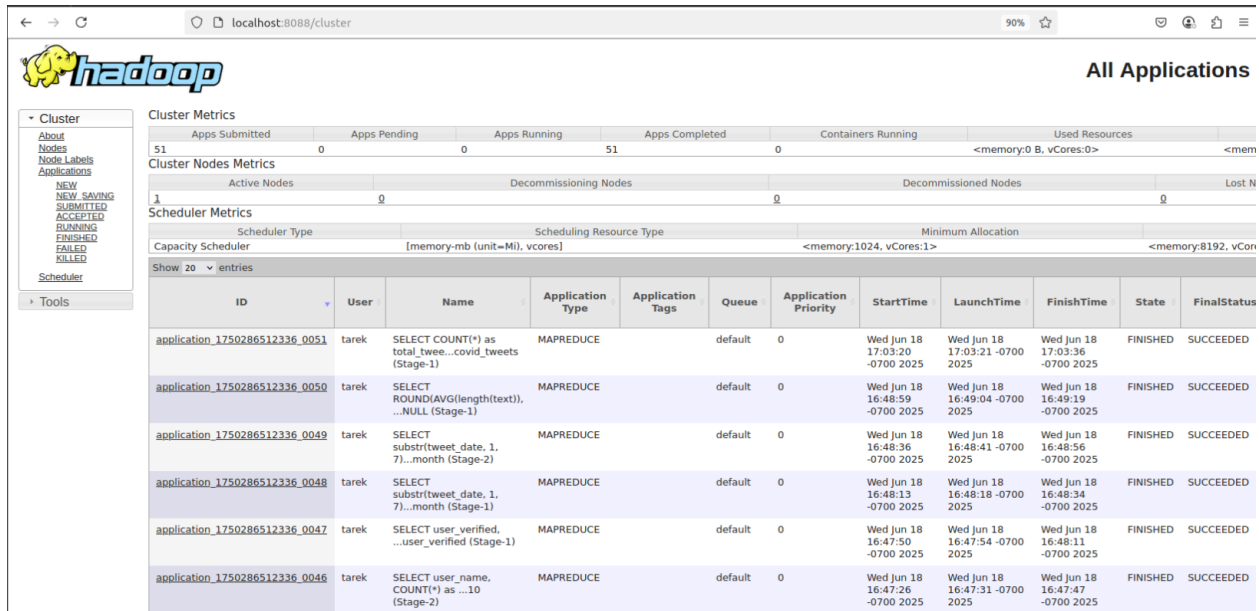
La création de tables externes Hive établit un mapping direct vers les fichiers CSV stockés dans HDFS, évitant la duplication des données tout en offrant une interface SQL familière. La définition du schéma de table respecte la structure du dataset COVID-19 avec typage approprié des colonnes.

```
hive> SHOW TABLES;
OK
covid_tweets
Time taken: 0.618 seconds, Fetched: 1 row(s)
hive> DESCRIBE covid_tweets;
OK
user_name           string
user_location       string
user_description     string
user_created         string
user_followers       int
user_friends         int
user_favourites      int
user_verified        boolean
tweet_date          string
text                 string
hashtags             string
source               string
is_retweet           boolean
Time taken: 0.294 seconds, Fetched: 13 row(s)
```

Figure 3. Création de la table HIVE

8. Résultats et analyse des données

8.1 Synthèse des résultats obtenus



The screenshot shows the Hadoop YARN Resource Manager web interface. The top navigation bar includes the Hadoop logo and the title "All Applications". The left sidebar contains a "Cluster" menu with options like "About", "Nodes", "Node Labels", "Applications", "NEW", "NEW SAVING", "SUBMITTED", "ACCEPTED", "RUNNING", "FINISHED", "FAILED", "KILLED", and "Scheduler". The main content area displays "Cluster Metrics" and "Scheduler Metrics". Below these, a table lists applications with columns for ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, LaunchTime, FinishTime, State, and FinalStatus. The table shows six applications, all of which are "FINISHED" and "SUCCEEDED".

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus
application_1750286512336_0051	tarek	SELECT COUNT(*) as total_tweet...covid_tweets (Stage-1)	MAPREDUCE		default	0	Wed Jun 18 17:03:20 -0700 2025	Wed Jun 18 17:03:21 -0700 2025	Wed Jun 18 17:03:36 -0700 2025	FINISHED	SUCCEEDED
application_1750286512336_0050	tarek	SELECT ROUND(AVG(length(text)), ...NULL (Stage-1)	MAPREDUCE		default	0	Wed Jun 18 16:48:59 -0700 2025	Wed Jun 18 16:49:04 -0700 2025	Wed Jun 18 16:49:19 -0700 2025	FINISHED	SUCCEEDED
application_1750286512336_0049	tarek	SELECT substr(tweet_date, 1, 7)...month (Stage-2)	MAPREDUCE		default	0	Wed Jun 18 16:48:36 -0700 2025	Wed Jun 18 16:48:41 -0700 2025	Wed Jun 18 16:48:56 -0700 2025	FINISHED	SUCCEEDED
application_1750286512336_0048	tarek	SELECT substr(tweet_date, 1, 7)...month (Stage-1)	MAPREDUCE		default	0	Wed Jun 18 16:48:13 -0700 2025	Wed Jun 18 16:48:18 -0700 2025	Wed Jun 18 16:48:34 -0700 2025	FINISHED	SUCCEEDED
application_1750286512336_0047	tarek	SELECT user_verified, ...user_verified (Stage-1)	MAPREDUCE		default	0	Wed Jun 18 16:47:50 -0700 2025	Wed Jun 18 16:47:54 -0700 2025	Wed Jun 18 16:48:11 -0700 2025	FINISHED	SUCCEEDED
application_1750286512336_0046	tarek	SELECT user_name, COUNT(*) as ...10 (Stage-2)	MAPREDUCE		default	0	Wed Jun 18 16:47:26 -0700 2025	Wed Jun 18 16:47:31 -0700 2025	Wed Jun 18 16:47:47 -0700 2025	FINISHED	SUCCEEDED

Figure5. Hadoop YARN ResourceManager

Traitement avec Mapreduce :

```
tarek@ubuntu:~$ cat ~/covid_analysis_results.txt
HASHTAG_CORONAVIRUS      7662
HASHTAG_COVID19 82954
HASHTAG_LOCKDOWN        868
HASHTAG_PANDEMIC        1332
KEYWORD_HOSPITAL        4396
KEYWORD_MASK      7560
KEYWORD_VACCINE 4116
RETWEETS              3
TOTAL_TWEETS      216890
```

Figure 2. Résultats job MapReduce

o Nombre total de tweets : 216890

o Nombre d'occurrences d'un hashtag spécifique (COVID19) : 82954

Analyse avec Apache Hive :

o Nombre total de tweets : 375857

o Nombre de tweets par date :

L'évolution du volume de tweets au fil des mois.

```
Total MapReduce CPU Time Spent: 12 seconds 80 msec
OK
Time taken: 49.885 seconds, Fetched: 2 row(s)
tarek@ubuntu:~/covid_data$ cat results/09_tweets_per_month.txt
2020-07 44295
2020-08 107947
```

Figure 14. Nombre de tweets par mois

o Hashtags les plus fréquents :

```
Total MapReduce CPU Time Spent: 15 seconds 770 msec
OK
Time taken: 55.768 seconds, Fetched: 20 row(s)
tarek@ubuntu:~/covid_data$ cat results/03_top_hashtags.txt
covid19 59209
coronavirus 4659
pandemic 955
trump 680
covid 666
lockdown 463
india 448
vaccine 415
wearamask 402
covid_19 278
masks 278
china 252
odisha 250
socialdistancing 241
usa 239
health 238
mask 228
corona 222
auspol 219
hydroxychloroquine 210
```

Figure 8. Les 20 hashtags les plus fréquents

Le nombre total de tweets dans le jeu de données.

```
Total MapReduce CPU Time Spent: 5 seconds 410 msec
OK
Time taken: 22.934 seconds, Fetched: 1 row(s)
tarek@ubuntu:~/covid_data$ cat results/01_total_tweets.txt
375857
```

Figure 6. Total des tweets avec Hive

Les **10 jours les plus actifs** en nombre de tweets.

```
tarek@ubuntu:~/covid_data$ cat results/02_tweets_per_day.txt
2020-07-25      14207
2020-08-22      9624
2020-08-30      6921
2020-07-31      6606
2020-08-18      6577
2020-08-01      6556
2020-08-02      6484
2020-08-14      6425
2020-08-12      6392
2020-07-28      6374
```

Figure 7. Top 10 des jours les plus actifs

Le nombre de tweets contenant certains **mots-clés** liés à la pandémie : **COVID-19, vaccin, masque ..**

```
Total MapReduce CPU Time Spent: 1 minutes 55 seconds 220 msec
OK
Time taken: 226.644 seconds, Fetched: 4 row(s)
tarek@ubuntu:~/covid_data$ cat results/04_keywords_mentions.txt
COVID-19      27348
Lockdown      786
Vaccine 1032
Mask 1944
```

Figure 9. Mentions des mots-clés liés au COVID-19

Les **applications ou plateformes** les plus utilisées pour tweeter (ex. Twitter pour iPhone, Android, Web App).

```
Total MapReduce CPU Time Spent: 22 seconds 230 msec
OK
Time taken: 101.844 seconds, Fetched: 10 row(s)
tarek@ubuntu:~/covid_data$ cat results/05_tweet_sources.txt
Twitter Web App 33605 8.94
Twitter for iPhone 27685 7.37
Twitter for Android 18406 4.90
Hootsuite Inc. 5199 1.38
TweetDeck 5142 1.37
Twitter for iPad 3075 0.82
Buffer 1870 0.50
Sprout Social 1391 0.37
IFTTT 982 0.26
Instagram 937 0.25
```

Figure 10. Sources de publication des tweets

La proportion de **retweets** par rapport aux **tweets originaux**.

```
Total MapReduce CPU Time Spent: 10 seconds 120 msec
OK
Time taken: 65.592 seconds, Fetched: 3 row(s)
tarek@ubuntu:~/covid_data$ cat results/06_retweets_vs_originals.txt
Original Tweets 354061 94.20
Original Tweets 20770 5.53
Retweets 1026 0.27
tarek@ubuntu:~/covid_data$
```

Figure 11. Répartition retweets vs originaux

Les **utilisateurs ayant tweeté le plus** souvent, avec leur nombre maximal d'abonnés et leur statut vérifié.

```
OK
Time taken: 51.671 seconds, Fetched: 10 row(s)
tarek@ubuntu:~/covid_data$ cat results/07_top_active_users.txt
GlobalPandemic.NET      679      26121   False
covidnews.ch           402       378     False
Open Letters            390     17522   True
Blood Donors India      282    1221298  True
Hindustan Times 280     7730317  True
Paperbirds_Coronavirus  259      147     False
IANS Tweets             244     54843   True
ANI                     233     4721276  True
Coronavirus Updates - Alexander Higgins 225      17467   False
Dushyant Vachhani       220       38     False
```

Figure 12. Top 10 des utilisateurs actifs

Comparaison entre **comptes vérifiés et non vérifiés**, avec des statistiques comme le nombre moyen d'abonnés.

```
Total MapReduce CPU Time Spent: 5 seconds 770 msec
OK
Time taken: 22.878 seconds, Fetched: 2 row(s)
tarek@ubuntu:~/covid_data$ cat results/08_verified_user_stats.txt
false 48940 4381.0 2286.0
true  5285 808911.0 2179.0
```

Figure 13. Analyse des comptes vérifiés

La **longueur moyenne**, minimale et maximale des tweets.

```
tarek@ubuntu:~/covid_data$ cat results/10_tweet_length_stats.txt
34.83  0  155
```

Figure 15. Longueur moyenne des tweets

8.2 Analyse des résultats

Les résultats obtenus révèlent des tendances significatives dans les discussions COVID-19 sur Twitter. Le dataset de **375,857 tweets** montre une activité intense durant l'été 2020, avec un pic le 25 juillet (14,207 tweets).

L'analyse thématique confirme la dominance du hashtag **#covid19** (59,209 occurrences) et révèle les préoccupations principales : mentions directes de COVID-19 (27,348), discussions sur les masques (1,944) et les vaccins (1,032).

Les sources de publication montrent une utilisation majoritaire des applications mobiles (Twitter Web App et iPhone), tandis que l'analyse des comptes révèle l'influence des comptes vérifiés avec une moyenne de 808,911 abonnés contre 4,381 pour les comptes non vérifiés.

8.3 Évaluation technique

Le déploiement Hadoop en mode pseudo-distribué a fonctionné efficacement pour traiter les 375,857 enregistrements. Les jobs MapReduce ont montré de bonnes performances malgré la configuration matérielle modeste (4 cœurs, 4 Go RAM).

L'intégration d'Apache Hive s'est révélée particulièrement utile pour l'analyse exploratoire grâce à son interface SQL familière. Le système HDFS a assuré une gestion stable des données avec une réplication appropriée.

9. Conclusion et perspectives

Ce travail pratique a permis de maîtriser les outils fondamentaux de l'écosystème Apache Hadoop pour l'analyse de données massives. L'installation et la configuration du cluster pseudo-distribué ont fourni une base solide pour comprendre le fonctionnement distribué.

L'analyse du dataset COVID-19 a démontré l'efficacité de la chaîne de traitement complète : de l'ingestion des données via HDFS, au traitement avec MapReduce, jusqu'à l'analyse avec Hive. Les résultats obtenus révèlent des patterns intéressants sur les discussions liées à la pandémie.

Perspectives d'amélioration :

- Migration vers un cluster multi-nœuds pour tester la scalabilité réelle
- Intégration d'Apache Spark pour des analyses en temps réel
- Extension de l'analyse avec des algorithmes de sentiment analysis
- Application de la méthodologie à d'autres datasets sociaux

Cette expérience constitue une base solide pour l'approfondissement des compétences en ingénierie des données et l'utilisation d'outils Big Data en environnement professionnel.

Références :

1. Apache Hadoop Documentation - <https://hadoop.apache.org/docs/>
2. Apache Hive Documentation - <https://hive.apache.org/>
3. Hadoop: The Definitive Guide - Tom White
4. Programming Hive - Edward Capriolo, Dean Wampler, Jason Rutherglen