UNIVERSITÉ PARIS 8

LIASD
Laboratoire informatique avancée de Saint-Denis
Université Paris 8

Rapport de projet

Framework Big Data

Jean-Christophe LAFLEUR
MASTER 1 OPTION BIG DATA
UNIVERSITE PARIS 8
Prof : Mme JAZIRI

Année Scolaire : 2021-2022

Sommaire

I. Introduction

De nos jours, les plateformes de streaming sont de plus en plus nombreuses et nous permettent d'accéder à du contenu cinématographique international, plus facilement, en nous évitant de chercher un lien menant à un site de streaming souvent infester par des virus en tout genre.

Mon projet a pour but, d'analyser les données de différent service de streaming afin de déterminer celle la mieux adapté à nos besoins.

La problématique dont je me suis posé a été « Quelle plateforme de streaming choisir ? ».

II. Base de Donnée

L'ensemble des données sont issues de fichiers CSV provenant des plateformes de Netflix, Amazon Prime, Disney+ et HULU, ayant été récupérer sur KAGGLE. Elles ont été mises à jour, il y a 4 mois (Octobre 2021).

Les fichiers CSV seront par la suite converti en fichier SQL, à l'aide d'un outil en ligne, afin de faciliter l'importation des données sur mysql, lors de l'utilisation de Sqoop.

III. Déroulement

a) Première approche

Afin de mener à bien ce projet, j'ai sélection un ensemble d'outils qui me seront utile, et pour cela j'ai choisi Hadoop, Hive et Sqoop. J'ai instancié 4 machines Ubuntu avec Hadoop sous AWS en EC2, afin de créer un cluster de plusieurs noeuds. Chaque machine a été configurer l'une après l'autre, afin d'obtenir un cluster suivant l'architecture maitre-esclaves. Nous avons une machine « master » avec le NameNode et 3 « slaves » avec des DataNodes.

Pour accéder aux machines depuis Windows et ainsi échanger avec AWS, j'ai utilisé PuTTY et winSCP.

Le HDFS a demandé une recherche continue pour le rendre fonctionnel, car les nœuds communiqué mal entre elles. Il permet au NameNode d'accéder correctement aux DataNodes. Sur la plateforme WebUI du DataNode (localhost :8088), on observe bien la présence des DataNodes.

De plus, Hive et Sqoop ont été installé sur l'instance NameNode, mais n'ont pas été tester à la suite de bugs d'instance fortement présent. Pour l'utilisation de Sqoop, il nécessite l'installation de mysql-connector-java pour permettre la connection avec mysql.

De plus, lors de bug récurrent, d'une ou plusieurs instances, sous EC2, nous devons constamment changer les DNS publics constamment. En effet, il se réinitialise à chaque démarrage d'instance, ce qui est pénible, et fait perdre énormément de temps lors de re-paramétrage. De plus, la faible disponibilité de RAM (t2.micro 1 Go) rend le cluster lent, et donc l'impossibilité de transférer des données vers les datanodes.

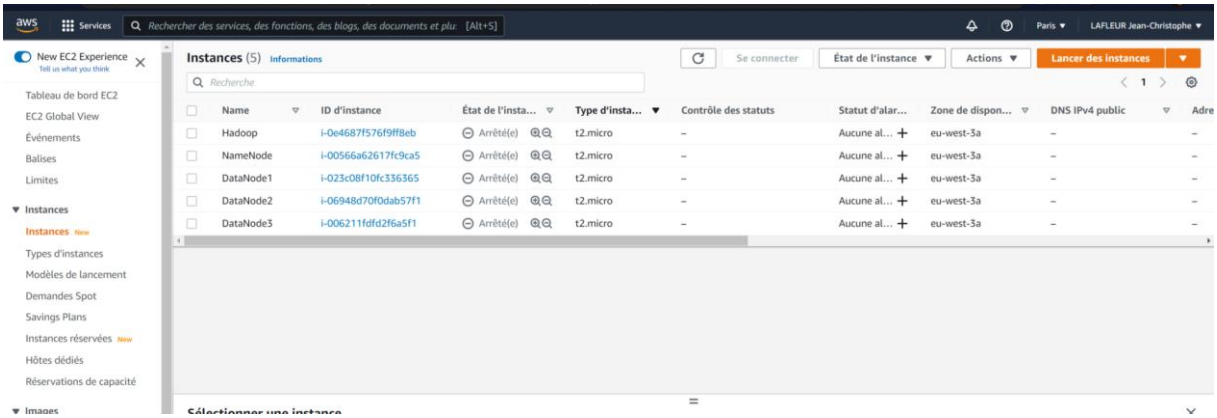

Figure 1 : Instances sous AWS

Comme nous pouvons le voir sur la Figure 1, les instances sont toutes à l'arrêt à la suite d'une erreur signalé par Amazon pour cause d'abus de ressources, rendant l'utilisation contraignant (Voir Figure 2).

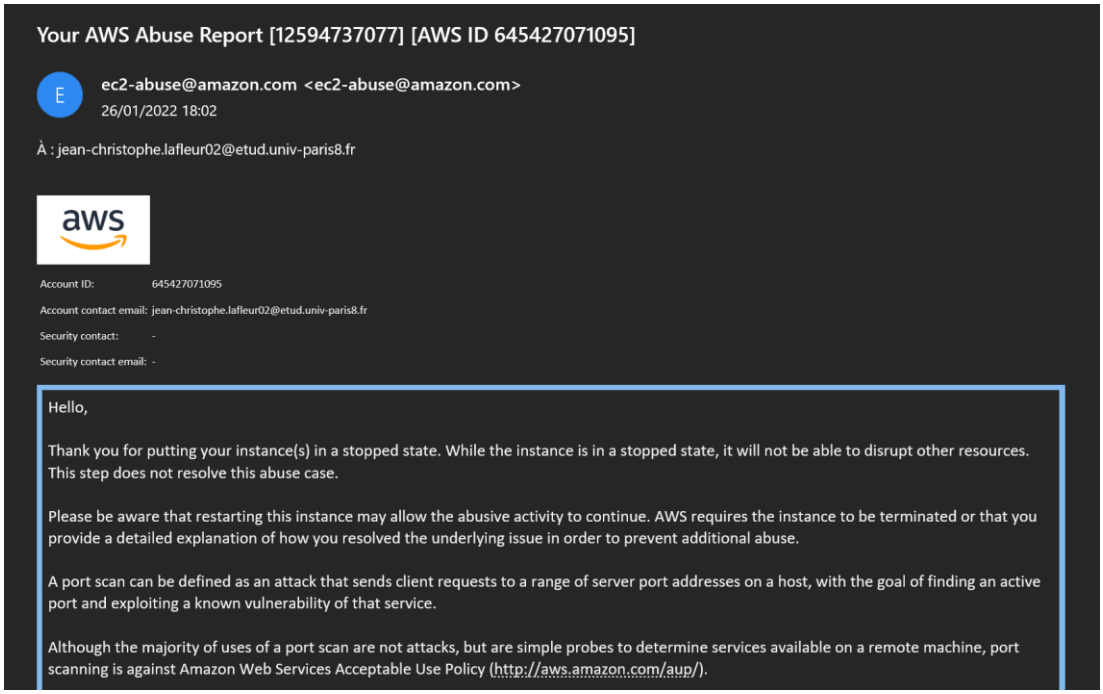

Figure 2: Mail d'abus AWS

b) Deuxième approche

Souhaitant encore travailler sur un cluster multi-nœuds, je me suis rapprocher des cloud de google (Google Cloud Platform) et de Microsoft (Azure). Mais j'ai vite changé d'avis car les fonds pouvant être obtenu sous Google Cloud Platform se dépense très vite et demande une vérification d'identité pour l'utilisation des machines. De plus, il n'est pas possible d'obtenir un compte Azure à l'aide de l'adresse mail étudiante de l'université.

c) Troisième approche

Pour pallier les difficultés rencontrées, lors de la première approche, j'ai décidé de réaliser mon projet en local sur un cluster à un seul nœud, en utilisant les outils vus en cours Hadoop, Hive et Sqoop directement installer sur le terminal Linux (WSL2 Ubuntu). Cela m'a permis de comprendre d'où certaines erreurs d'installation pourrait provenir, lors de la mise en route de Hadoop, tout en facilitant la mise en place d'un cluster.

En premier lieu, j'ai essayé d'exporter les tables créer sous Hive sans passer par la conversion de fichier SQL, en récupérant les données des fichiers CSV, mais après plusieurs essaie d'appel de requête « sqoop export » permettant de les transférer vers mysql, je n'ai pas réussi à réaliser ce dont, je souhaiter effectuer, et ai donc abandonner cette solution.

```
2022-01-28 19:57:39,263 INFO mapreduce.ExportJobBase: Exported 0 records.
2022-01-28 19:57:39,264 ERROR mapreduce.ExportJobBase: Export job failed!
2022-01-28 19:57:39,264 ERROR tool.ExportTool: Error during export:
Export job failed!
        at org.apache.sqoop.mapreduce.ExportJobBase.runExport(ExportJobBase.java
        at org.apache.sqoop.manager.SqlManager.exportTable(SqlManager.java:931)
        at org.apache.sqoop.tool.ExportTool.exportTable(ExportTool.java:80)
        at org.apache.sqoop.tool.ExportTool.run(ExportTool.java:99)
        at org.apache.sqoop.Sqoop.run(Sqoop.java:147)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
        at org.apache.sqoop.Sqoop.runSqoop(Sqoop.java:183)
        at org.apache.sqoop.Sqoop.runTool(Sqoop.java:234)
        at org.apache.sqoop.Sqoop.runTool(Sqoop.java:243)
```

Figure 3: Erreur obtenu lors de l'appel de sqoop export

L'une des solutions, dont j'ai utilisé finalement a été l'importation des données sur mysql et et faire appel à une requête « Sqoop import » permettant de les données vers HDFS et seront traiter à l'aide de Hive.

IV. Traitement de l'information

L'importation des données, sous mysql a été réalisé par les requêtes suivantes :

```
sudo mysql -u root -p // accès à mysql

source /tmp/netflix_data.sql  //importation données sql

source /tmp/amazon_data.sql  //importation données sql

source /tmp/disney_data.sql //importation données sql

source /tmp/hulu_data.sql //importation données sql
```

Une fois les tables ont été créer, je les ai importés sous HDFS avec les requêtes Sqoop suivantes :

```
sqoop import --connect jdbc:mysql://localhost:3306/project --username root --password root --table disney_data --m 1 --target-dir /user/hive/warehouse/stream_db.db/disney_data

sqoop import --connect jdbc:mysql://localhost:3306/project --username root --password root --table hulu_data --m 1 --target-dir /user/hive/warehouse/stream_db.db/hulu_data

sqoop import --connect jdbc:mysql://localhost:3306/project --username root --password root --table netflix_data --m 1 --target-dir /user/hive/warehouse/stream_db.db/netflix_data

sqoop import --connect jdbc:mysql://localhost:3306/project --username root --password root --table amazon_data --m 1 --target-dir /user/hive/warehouse/stream_db.db/amazon_data
```

```
drwxr-xr-x   - jc supergroup          0 2022-01-30 22:38 /user/hive/warehouse/stream_db.db/amazon_data
-rw-r--r--   1 jc supergroup          0 2022-01-30 22:38 /user/hive/warehouse/stream_db.db/amazon_data/_SUCCESS
-rw-r--r--   1 jc supergroup    4014721 2022-01-30 22:38 /user/hive/warehouse/stream_db.db/amazon_data/part-m-00000
drwxr-xr-x   - jc supergroup          0 2022-01-30 22:41 /user/hive/warehouse/stream_db.db/disney_data
-rw-r--r--   1 jc supergroup          0 2022-01-30 22:41 /user/hive/warehouse/stream_db.db/disney_data/_SUCCESS
-rw-r--r--   1 jc supergroup     377905 2022-01-30 22:41 /user/hive/warehouse/stream_db.db/disney_data/part-m-00000
drwxr-xr-x   - jc supergroup          0 2022-01-30 22:39 /user/hive/warehouse/stream_db.db/hulu_data
-rw-r--r--   1 jc supergroup          0 2022-01-30 22:39 /user/hive/warehouse/stream_db.db/hulu_data/_SUCCESS
-rw-r--r--   1 jc supergroup    1128690 2022-01-30 22:39 /user/hive/warehouse/stream_db.db/hulu_data/part-m-00000
```

Figure 4: Données présent dans le HDFS

J'ai par la suite créé des tables externes sur Hive afin de garder les données même après la suppression de ces tables, avec les requêtes suivantes :

```
create external table hulu_data(show_id string,type string,title string,director string,cast_ string,country string,date_added string,release_year int,rating string,duration string,listed_in string,description string) ROW FORMAT DELIMITED FIELDS TERMINATED BY "," LINES TERMINATED BY '\n';

create external table netflix_data(show_id string,type string,title string,director string,cast_ string,country string,date_added string,release_year int,rating string,duration string,listed_in string,description string) ROW FORMAT DELIMITED FIELDS TERMINATED BY "," LINES TERMINATED BY '\n';

create external table amazon_data(show_id string,type string,title string,director string,cast_ string,country string,date_added string,release_year int,rating string,duration string,listed_in string,description string) ROW FORMAT DELIMITED FIELDS TERMINATED BY "," LINES TERMINATED BY '\n';

create external table disney_data(show_id string,type string,title string,director string,cast_ string,country string,date_added string,release_year int,rating string,duration string,listed_in string,description string) ROW FORMAT DELIMITED FIELDS TERMINATED BY "," LINES TERMINATED BY '\n';
```

J'ai ensuite réalisé quelques requêtes Hive basiques, pour vérifier que les données sont correctes :

```
Select*from netflix_data limit 10 ;

Select*from amazon_data limit 10 ;

Select*from disney_data limit 10 ;

Select*from hulu_data limit 10 ;
```

Par la suite, j'ai réalisé des buckets afin de récupérer les données dont le type et « Movie » ou « TV show »:

```
create table netflix_bucket(show_id string,type string,title string,director string,cast_ string,country string,date_added string,release_year int,rating string,duration string,listed_in string,description string) clustered by (type) into 4 buckets ROW FORMAT DELIMITED FIELDS TERMINATED BY '/t' ;
```

```
create table amazon_bucket(show_id string,type string,title string,director string,cast_ string,country string,date_added string,release_year int,rating string,duration string,listed_in string,description string) clustered by (type) into 12 buckets ROW FORMAT DELIMITED FIELDS TERMINATED BY '/t' ;

create table disney_bucket(show_id string,type string,title string,director string,cast_ string,country string,date_added string,release_year int,rating string,duration string,listed_in string,description string) clustered by (type) into 2 buckets ROW FORMAT DELIMITED FIELDS TERMINATED BY '/t' ;

create table hulu_bucket(show_id string,type string,title string,director string,cast_ string,country string,date_added string,release_year int,rating string,duration string,listed_in string,description string) clustered by (type) into 14 buckets ROW FORMAT DELIMITED FIELDS TERMINATED BY '/t' ;
```

J'ai ensuite inséré les valeurs dans ces tables comme suit, pour clusterer les données :

```
Insert overwrite table netflix_bucket select*from netflix_data;

Insert overwrite table amazon_bucket select*from amazon_data;

Insert overwrite table disney_bucket select*from disney_data;

Insert overwrite table hulu_bucket select*from hulu_data;
```

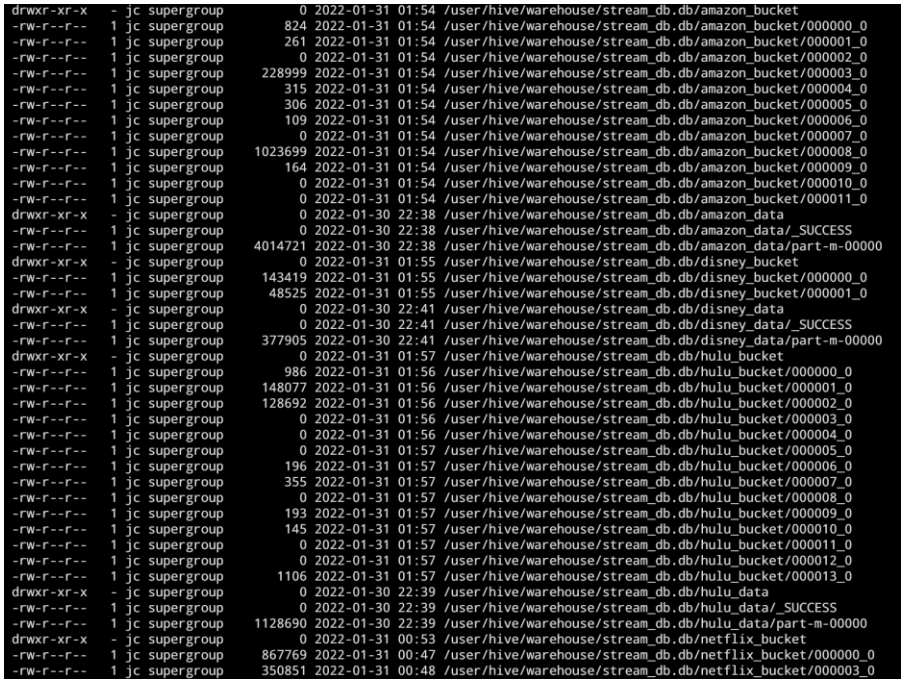

Figure 5: HDFS buckets

Par la suite, j'ai supprimé les fichiers dont les données ne correspondaient à ce dont je souhaitais dans le HDFS, puisqu'ils n'appartiennent à aucun des types souhaité (« Movie » et « TV show »)

```
hdfs dfs -rm -r /user/hive/warehouse/stream_db.db/….
```

J'ai par la suite, chercher à connaître le nombre de « Movie » et « TV show » présent afin de les comparer.

```
select type, count(type) from netflix_bucket group by type;

select type, count(type) from disney_bucket group by type;

select type, count(type) from amazon_bucket group by type;

select type, count(type) from hulu_bucket group by type;
```

A la suite de cette comparaison, j'ai tenté de réaliser un pourcentage permettant d'indiquer la plateforme la plus apte à nos besoins en termes de contenu, que ce soit en Film ou série télévisée.

J'ai d'abord exécuté plusieurs requêtes sous Hive permettant d'obtenir cette information mais dont le résultat restait fixé sur 0.0. Les voici ci-dessous :

```
select type, sum(type)/count(type) from netflix_bucket group by type;

select type, sum(type)/count(type)*100 from disney_bucket where type <> "NULL" group by type;

select type,sum(type)/count(type) from disney_bucket tablesample(bucket 1 out of 2 on type) group by type;
```

Etant passer outre, j'ai récupéré les données sous Hive, et j'ai réalisé un graphique sous Excel permettant de mieux visualiser ces différences.

	Netflix	Amazon	Disney	Hulu
Movie	6131	7814	1052	1484
TV show	2676	1854	398	1589

Valeur obtenue par les requêtes précédentes

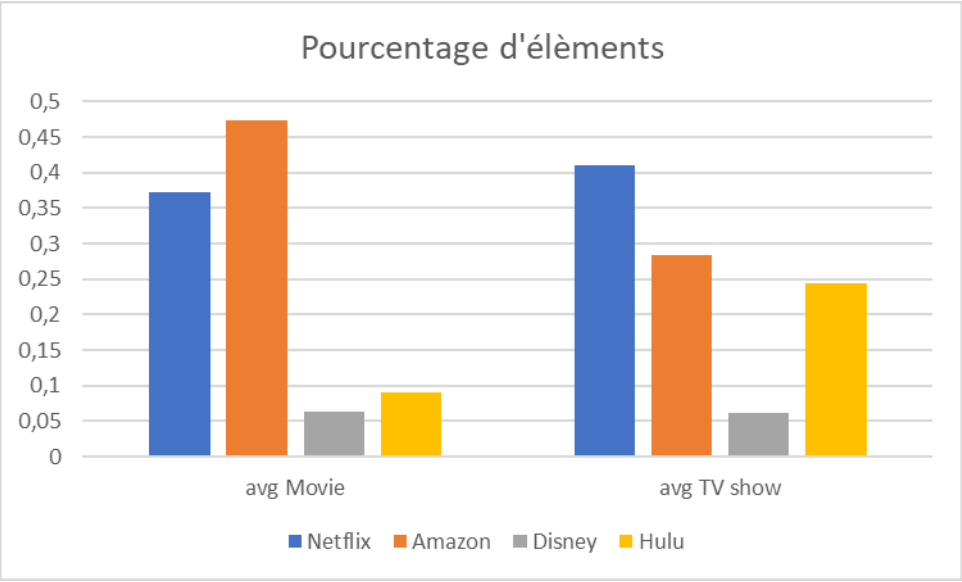

Figure 6 : Graphique

D’après ces données, on observe qu’en terme de « film », la plateforme Amazon Prime semble posséder un catalogue plus important, par rapport à ces concurrents, tandis que Netflix est plus attractif sur sa richesse en « série TV ».

V. Conclusion

Les données présentes dans les fichiers CSV sont très mal organisée dans la plupart des colonnes et peuvent porter à confusion sur la qualité du traitement des données. Sqoop me permet de gérer la partie transfert de donnée entre RMBD et HDFS avec l’import/export dont ce dernier semble rencontrer des difficultés à s’exécuter sur mon PC. Hive me permet de lancer des requêtes à l’ensemble des données présent dans la database. En adaptant cette méthode de travail avec ces outils, je sais exactement qu’elle tâche leur son attribué.

A l’issus de ce projet, j’ai pu mettre en place qu’une partie des idées dont je souhaitais réaliser, puisque j’ai consacré beaucoup de temps à l’instanciation des machines sous EC2, dont les problèmes s’enchainer les uns après les autres. De plus, lors du traitement des données, on aurait pu utiliser une table unique qui aurait nettement simplifier certaines opération, telle que le calcul des moyennes et bien d’autres. Mais cela demanderait de vérifier que chaque donnée soit bien agencé, ce qui n’est pas le cas. Chaque table est traitée de façon indépendante, afin d’avoir la certitude qu’il n’y a pas d’erreur.

De plus, le choix d’une plateforme de streaming dépend du type de contenu que l’on souhaite regarder mais aussi du choix qui peut être présenter. Nous pouvons prendre le cas de Disney+ dont le catalogue en série TV est faible mais dont la qualité est riche. Le choix d’une plateforme peut également être influencé par les tendances actuelle ou passé. Il faut également souligner que les notations du contenu de chacune des plateformes n’est pas présente dans les bases de données, et aurait pu permettre de mien cerné certains contrastes, tel que la qualité.