UNIVERSITE GASTON BERGER

L'excellence au service du développement

INSTITUT POLYTECHNIQUE DE SAINT-LOUIS

Big data

Ingestion de données dans Big data

Présenté par :

Abdoul Majid Ba

ING3-GeIT

Professeur :

Dr. Djibril MBOUP

INTRODUCTION

Le terme « big data » désigne principalement des ensembles de données trop volumineux ou trop complexes pour être traités par les logiciels d'application traditionnels de traitement des données. Ainsi les bases de données relationnellles qui jadis servaient de stockage de données peuvent maintenant être regroupées dans ce qu'on appelle un entrepôt de données (data warehouse). Ce dernier est un type de système de gestion de données conçu pour permettre et faciliter les activités de Business Intelligence (BI), en particulier l'analytique.

En effet pour faire la migration des données stockées au niveau des bases de données relationnelles dans un data warehouse, nous pouvons utiliser certains outils comme apache scoope pour cela. Dans ce tutoriel, nous allons essayer de montrer comment utiliser apache scoope pour faire la migration des données contenues dans une base de données mysql vers le big data. Dans un premier temps, nous allons nous concentrer sur l'ingestion des données avec apache scoope et ensuite nous allons nous pencher sur le traitement des données avec apache hive.

I. Ingestion de données avec apache scoope

Apache Sqoop (SQL-to-Hadoop) est un outil utilisé pour transférer des données volumineuses entre les systèmes de stockage de données relationnelles (comme les bases de données SQL) et le système de fichiers distribués Hadoop (HDFS) ou les entrepôts de données Hadoop tels que Hive. Cela dit pour faire l'ingestion de données dans le big data nous aurons besoin d'une base de données (mysql dans notre cas), d'un data warehouse (nous utilserons Hive) et d'un outil pour la migration : apache scoope.

Cette partie sera diviser en deux étapes : d'abord nous allons nous intéresser sur la base de données relationnelle et ensuite nous allons lancer notre machine virtuelle et importer les données dans apache Hive.

1. Structure et chargement de la base de données

Dans ce tutoriel nous allons utiliser une base de données contenant déjà des données. Retail DB est une base de données qui contient des données de ventes d'une entreprise Ecommerce.

Voici le schema de Retail DB :

Avant de faire le chargement de la base de données, nous allons d'abord créer un utilisateur, et puis une base de données où les données seront chargées.

- Création d'un compte utilisateur admin

```
MySQL  localhost:3306 ssl  SQL > \connect root@localhost
Creating a session to 'root@localhost'
Please provide the password for 'root@localhost': ****
Save password for 'root@localhost'? [Y]es/[N]o/Ne[v]er (default No): y
Fetching global names for auto-completion... Press ^C to stop.
Closing old connection...
Your MySQL connection id is 15
Server version: 9.0.1 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL  localhost:3306 ssl  SQL > CREATE user retail_user identified by 'hadoop';
Query OK, 0 rows affected (0.0119 sec)
MySQL  localhost:3306 ssl  SQL >
```

- Création de la base de données et affectation des privilèges

Après création de la base de données avec la commande

```
CREATE database retail_db;
```

Nous ajoutons les droits d'utilisateurs sur la base de donnée. Voir image ci-dessous.

```
MySQL  localhost:3306 ssl  SQL > GRANT ALL ON retail_db.* to retail_user;
Query OK, 0 rows affected (0.0046 sec)
MySQL  localhost:3306 ssl  SQL >
```

- Connection avec retail_user

```
MySQL  localhost:3306 ssl  SQL > \connect retail_user@localhost
Creating a session to 'retail_user@localhost'
Fetching global names for auto-completion... Press ^C to stop.
Closing old connection...
Your MySQL connection id is 27
Server version: 9.0.1 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL  localhost:3306 ssl  SQL >
```

- Chargement des données

Pour faire le chargement des données, nous allons d'abord télécharger le script SQL de Retail DB puis le charger avec la commande suivante : source « chemin du script »;

```
MySQL  localhost:3306 ssl  retail_db  SQL > source C:\Users\Majid\Downloads\retail_db.sql;
Query OK, 0 rows affected (0.0004 sec)
Query OK, 0 rows affected (0.0002 sec)
Query OK, 0 rows affected (0.0004 sec)
Query OK, 0 rows affected (0.0003 sec)
Query OK, 0 rows affected (0.0003 sec)
Query OK, 0 rows affected (0.0003 sec)
Query OK, 0 rows affected (0.0003 sec)
Query OK, 0 rows affected (0.0003 sec)
```

Pour vérifier que le chargement est bien effectué, nous allons consulter les tables de la base :

```
MySQL  localhost:3306 ssl  retail_db  SQL > show tables;
+----------------------+
| Tables_in_retail_db  |
+----------------------+
| categories           |
| customers            |
| departments          |
| order_items          |
| orders               |
| products             |
+----------------------+
```

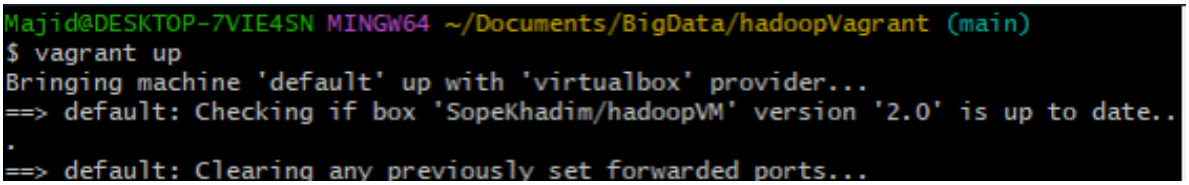
Nous constatons que toute les tables ont été bien importées et même les données de la base :

```
MySQL  localhost:3306 ssl  retail_db  SQL > select customer_id, customer_fname, customer_lname from customers where cus
tomer_id=1;
+-------------+---------------+---------------+
| customer_id | customer_fname | customer_lname |
+-------------+---------------+---------------+
|           1 | Richard        | Hernandez      |
+-------------+---------------+---------------+
1 row in set (0.0009 sec)
```

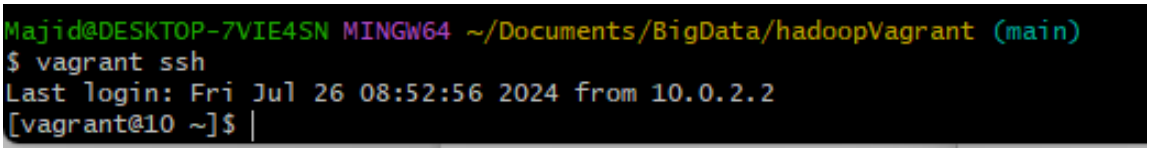
2. Ingestion des données dans Hive

Tout d'abord nous allons démarrer vagrant VM et nous connecter.

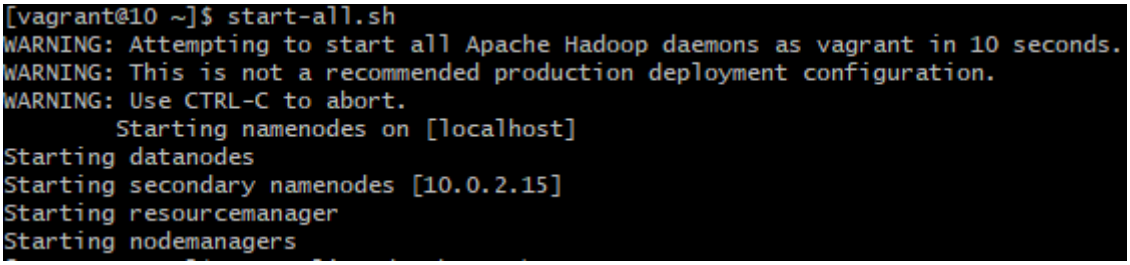
Démarrage de vagrant :



Connection avec vagrant ssh :



Etant déjà connecté sur vagrant, pour pouvoir accéder sur apache scoope, nous allons démarrer hadoop avec la commande : start-all.sh



Une fois hadoop démarrer nous allons essayer de nous connecter à la base de données se trouvant dans ma machine locale. Donc nous devons nous assurer que la machine virtuelle se trouve dans le même réseau que celui local pour permettre la connexion.

Pour que la machine virtuelle soit dans le même réseau, il faut modifier le fichier vagrantfile et mettre le non du réseau sur lequel la machine local est connecté.

Voici la ligne correspondante à modifier dans le fichier :

config.vm.network :public_network, :bridge => "wlo1 Wi-Fi (nom de votre réseau)"

Une fois faite, nous pouvons vérifier si la connexion est établie :

sqoop list-databases --connect "jdbc:mysql://Addresse IP machine local:3306/retail_db" --username retail_user --password hadoop

```
[vagrant@10 ~]$ sqoop list-databases --connect "jdbc:mysql://192.168.1.11:3306/r
etail_db" --username retail_user --password hadoop
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/lib/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2024-07-27 10:50:20,140 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2024-07-27 10:50:20,256 WARN tool.BaseSqoopTool: Setting your password on the co
mmand-line is insecure. Consider using -P instead.
2024-07-27 10:50:20,391 INFO manager.MySQLManager: Preparing to use a MySQL stre
aming resultset.
Sat Jul 27 10:50:25 UTC 2024 WARN: Establishing SSL connection without server's
identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ an
d 5.7.6+ requirements SSL connection must be established by default if explicit
option isn't set. For compliance with existing applications not using SSL the ve
rifyServerCertificate property is set to 'false'. You need either to explicitly
disable SSL by setting useSSL=false, or set useSSL=true and provide truststore f
or server certificate verification.
mysql
information_schema
performance_schema
sys
world
retail_db
[vagrant@10 ~]$
```

Nous constatons que c'est bon et que nous voyons bien la base de données retail_db que nous avions en local dans mysql.

Cela veut dire que la machine virtuelle peut accéder maintenant à ma base de données mysql et pourra voir les tables de la base et éventuellement importer des données car l'utilisateur connecté retail_user a tous les privilèges sur retail_db. Voici des images illustratives :

```
[vagrant@10 ~]$ sqoop list-tables --connect "jdbc:mysql://192.168.1.11:3306/reta
il_db" --username retail_user --password hadoop
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/lib/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2024-07-27 10:53:25,933 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2024-07-27 10:53:26,046 WARN tool.BaseSqoopTool: Setting your password on the co
mmand-line is insecure. Consider using -P instead.
2024-07-27 10:53:26,161 INFO manager.MySQLManager: Preparing to use a MySQL stre
aming resultset.
Sat Jul 27 10:53:30 UTC 2024 WARN: Establishing SSL connection without server's
identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ an
d 5.7.6+ requirements SSL connection must be established by default if explicit
option isn't set. For compliance with existing applications not using SSL the ve
rifyServerCertificate property is set to 'false'. You need either to explicitly
disable SSL by setting useSSL=false, or set useSSL=true and provide truststore f
or server certificate verification.
categories
customers
departments
order_items
orders
products
[vagrant@10 ~]$
```

L’image ci-dessus motre la liste des tables en utilisant la commande :

sqoop list-tables --connect "jdbc:mysql://192.168.1.11:3306/retail_db" --username retail_user --password hadoop
Dès lors nous pouvons maintenant importer les tables dans hive avec la commande :

sqoop import --connect "jdbc:mysql://192.168.1.11:3306/retail_db" --username retail_user --password hadoop --table nom table --as-parquetfile --target-dir=/user/hive/warehouse/retailDb/”nom table” --delete-target-dir

Voici le résultats après import:

```
[vagrant@10 ~]$ hdfs dfs -ls /user/hive/
Found 1 items
drwxrwxr-x   - vagrant supergroup          0 2024-07-27 11:33 /user/hive/warehouse
[vagrant@10 ~]$ hdfs dfs -ls /user/hive/warehouse/retailDb
Found 6 items
drwxr-xr-x   - vagrant supergroup          0 2024-07-27 12:02 /user/hive/warehouse/retailDb/categories
drwxr-xr-x   - vagrant supergroup          0 2024-07-27 12:42 /user/hive/warehouse/retailDb/customers
drwxr-xr-x   - vagrant supergroup          0 2024-07-27 12:49 /user/hive/warehouse/retailDb/departments
drwxr-xr-x   - vagrant supergroup          0 2024-07-27 12:51 /user/hive/warehouse/retailDb/order_items
drwxr-xr-x   - vagrant supergroup          0 2024-07-27 12:56 /user/hive/warehouse/retailDb/orders
drwxr-xr-x   - vagrant supergroup          0 2024-07-27 13:00 /user/hive/warehouse/retailDb/products
[vagrant@10 ~]$
```

Pour finir cette partie d’ingestion, nous allons voir si les tables sont créées dans hive.

```
hive> show tables;
OK
Time taken: 1.008 seconds
hive>
```

On voit que après l’importation les tables n’ont pas été créée.

Donc avant de faire le traitement nous allons d’abord créer les tables dans Hive.

II. Data processing avec apache Hive

Comme mentionné précédemment, nous allons d’abord créer le schéma des tables dans Hive.

Voici un script pour la création de la table customers :

```
CREATE EXTERNAL TABLE IF NOT EXISTS customers (
        customer_id int,
        customer_fname STRING,
        customer_lname STRING,
        customer_email STRING,
        customer_password STRING,
        customer_street STRING,
        customer_city STRING,
        customer_state STRING,
        customer_zipcode STRING )
```

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS PARQUET

LOCATION 'hdfs:///user/hive/warehouse/retailDb/customers';

Après exécution des différents scripts, nous obtenons :

```
hive> show tables;
OK
categories
customers
departments
order_items
orders
products
Time taken: 0.038 seconds, Fetched: 6 row(s)
hive>
```

Cela montre que les tables ont bien été ajoutées dans Hive.

Nous allons maintenant essayer de répondre aux questions en écrivant les requête sql correspondant aux différentes questions :

1) Trouver le nombre total de commandes passées par chaque client au cours de l'année 2014. Le statut de la commande doit être COMPLET, le format order_date est au format unix_timestamp
   Requéte : select order_id, count(*) as total_commande
   from orders where order_status="COMPLET"
   and DATE_FORMAT(FROM_UNIXTIME(order_date),'%Y')='2014'
   group by order_customer_id;

2) Afficher le nom et le prénom des clients qui n'ont passé aucune commande, triés par customer_lname puis customer_fname
   Requête : select customer_fname, customer_lname
   from customers where customer_id
   not in (select order_customer_id from orders)
   order by customer_lname, customer_fname;

3) Afficher les détails des top 5 clients par revenue pour chaque mois. Vous devez obtenir tous les détails du client ainsi que le mois et les revenus par mois. Les données doivent être triées par mois dans l'ordre croissant et les revenus par mois dans l'ordre décroissant.
   Requête :

7) Afficher tous les clients qui ont passé une commande d'un montant supérieur à 200 $

   Requête :
   select customer_fname, customer_lname
   from customers c, orders o, order_items ot
   where c.customer_id=o.order_customer_id and ot.order_item_order_id=o.order_id
   and ot.order_item_subtotal>200;

8) Afficher les clients de la "customers" dont les noms customer_fname commence par "Rich"
Requête :

```
select customer_fname
from customers
where customer_fname like "Rich%" ;
```

9) Fournir le nombre total de clients dans chaque état (state) dont le prénom commence par « M »
Requête :

```
select customer_state, count(*) as total_client
from customers
where customer_fname like "M%"
group by customer_state ;
```

10. Trouver le produit le plus cher dans chaque catégorie

```
select product_id, product_name, product_category_id
from products
where product_price= select max(product_price) from product
group by product_category_id ;
```

11) Trouvez les 10 meilleurs produits qui ont généré les revenus les plus élevés

```
select p.product_id, p.product_name,
SUM(oi.order_item_subtotal) AS total_revenue
From products p
JOIN
order_items oi ON p.product_id = oi.order_item_product_id
GROUP BY
p.product_id,
p.product_name
 ORDER BY
total_revenue DESC
LIMIT 10;
```