TP02 – Big Data

Chapitre III : Systèmes distribués

TP02 – Hadoop

Ing. Sidi Mahmoud RHIL

Objectif

- Mettre en route la plateforme Big Data Cloudera à l'aide d'une machine virtuelle.
- Ecrire et exécuter un programme MapReduce
- Manipuler et gérer des fichiers au sein du système de fichier distribué HDFS.
- Explorer et analyser des données en utilisant Hive.

Environnement

- **Mise en route de la plateforme BigData Cloudera**

- Lancer la plateforme Big Data Cloudera via la VM Cloudera QuickStart et vérifier l'état des services.

- Faire le nécessaire pour que tous les services soient démarrés.

Exercice 1 : MapReduce – Analyse des logs Web

Vous travaillez comme Data Engineer junior dans l'entreprise WebMarket, une société qui exploite une plateforme de vente en ligne.

Chaque jour, des millions de requêtes HTTP sont envoyées au serveur web, et toutes ces requêtes sont enregistrées dans des fichiers de logs Apache conservés sur les serveurs Web.

Le service de supervision souhaite obtenir un tableau de bord quotidien permettant de suivre :

- Le volume global de trafic,
- Les erreurs rencontrées par les utilisateurs (ex : pages introuvables),
- Les éventuels problèmes internes (ex : erreurs serveur).

Pour commencer ce travail, on vous demande un premier job MapReduce simple permettant de résumer la qualité du trafic de la journée.

Votre tâche consiste à récupérer le fichier de log Apache access.log et le déposer dans HDFS et puis écrire un programme MapReduce en Python2 qui analyse le fichier de log et produit, pour chaque **code de réponse HTTP**, le **nombre total de requêtes** renvoyant ce code.

Exemples de codes HTTP fréquents :

- **200** : requête réussie
- **302** : redirection
- **404** : page non trouvée
- **500** : erreur interne du serveur

On vous fournit un extrait du fichier de log access.log :

```
192.168.1.2 - - [10/Jan/2025:13:55:36] "GET /index.html HTTP/1.1" 200 1024
192.168.1.3 - - [10/Jan/2025:13:55:42] "GET /contact.html HTTP/1.1" 404 512
192.168.1.2 - - [10/Jan/2025:13:55:46] "POST /login HTTP/1.1" 200 2048
192.168.1.4 - - [10/Jan/2025:13:55:52] "GET /secret HTTP/1.1" 403 256
192.168.1.3 - - [10/Jan/2025:13:55:56] "GET /index.html HTTP/1.1" 200 1024
```

Chaque ligne contient :

- L'adresse IP du client,
- La date et l'heure de la requête HTTP,
- La requête HTTP (méthode + ressource + protocole),
- Le code HTTP,
- La taille de la réponse en octet.

Le code HTTP, qui vous intéresse ici, est le sixième élément si on sépare la ligne par des espaces.

Travail demandé

Vous devez développer un **mapper** qui lit les lignes de logs une par une et extrait, pour chacune d'elles, le code HTTP associé à la requête.

Le mapper doit produire une sortie permettant au reducer de compter le nombre de requêtes par code.

Un **reducer** qui lit la sortie du mapper (regroupée par codes) et calcule, pour chaque code HTTP, le nombre total d'occurrences dans le fichier.

Le reducer doit produire un rapport final du type :

```
200    3
403    1
404    1
```

Exercice 2 : Intégration des logs web dans Hive pour analyses avancées

Dans l'exercice précédent, vous avez développé un programme MapReduce en Python permettant de compter les occurrences des codes HTTP dans les logs Apache de l'entreprise.

L'équipe souhaite maintenant aller plus loin.

Afin de faciliter les analyses, de réduire les temps de traitement et d'ouvrir la voie à la datavisualisation, la société décide de centraliser tous les logs web dans Hive, sous forme de d'une table optimisée pour les requêtes.

Votre mission consiste à mettre en place toute la chaîne d'ingestion et d'analyse.

Pour améliorer la visibilité sur l'activité du site (volumes, erreurs, navigation, performance), les équipes Data et BI souhaitent :

- Centraliser les fichiers de logs dans HDFS ;
- Créer une couche d'accès via Hive pour faciliter les requêtes SQL ;
- Créer une table optimisée (en ORC) pour accélérer les analyses ;
- Préparer les données pour un outil de visualisation externe (Power BI, Tableau, Qlik, etc.).

Vous êtes chargé de mettre en place cette première brique.

Travail demandé :

1. Création d'un répertoire HDFS et dépôt du fichier

Créer un répertoire HDFS qui contiendra tous les fichiers log et déposer le fichier access.log dans ce répertoire

2. Création d'une table externe

Créer une table externe pointant sur ce répertoire, cette table doit :

- Lire directement les logs sans les déplacer ;
- Utiliser un SerDe adapté au format Apache (RegexSerDe) ;
- Exposer les colonnes utiles : IP, ident, utilisateur, timestamp, requête, statut HTTP, taille de réponse.

Cette table permettra d'avoir une vue brute des logs.

3. Création d'une table interne ORC

Créer une table INTERNE en format ORC, optimisée pour, la compression, la vitesse de lecture et les requêtes analytiques.

Cette table doit contenir des champs restructurés, propres et bien typés, par exemple :

- host : adresse IP du client
- logdate : de type TIMESTAMP
- methode : méthode HTTP,
- path : chemin demandé,
- protocole
- status : code de retour,
- bytes de type INT

4. Transformation et insertion des données

À partir de la table externe :

- extraire proprement les informations de la requête (GET /page HTTP/1.1) ;
- convertir la date de STRING à TIMESTAMP
- convertir le bytes de STRING à INT
- insérer les données nettoyées dans la table ORC.

5. Requêtes analytiques (HiveQL)

Réaliser plusieurs requêtes d'analyse de données pour le service Data/BI :

1. Nombre total de requêtes par code HTTP
2. Top 10 des pages les plus visitées
3. Top 5 des IP les plus actives en nombre de requêtes HTTP
4. Volume total de données renvoyées par le serveur (en MO)
5. Proportion des erreurs HTTP (4xx, 5xx), en sortie il faut avoir 3 colonnes :

   client_errors : le nombre d'erreur avec code entre 400 et 499

   server_errors : le nombre d'erreur avec code entre 500 et 599

   total : le nombre total de requêtes
6. Nombre de requêtes par jour et par heure
7. Répartition du trafic par méthode HTTP (GET, POST, etc.), en sortie il faut avoir 3 colonnes :

   methode : méthode HTTP

   nb_requetes : nombre de requêtes de la méthode

   pourcentage : pourcentage par rapport au nombre total de requêtes

Livrables attendus : Un fichier PDF avec les éléments ci-dessous :

- Une copie d'écran de Cloudera manger avec les différents services Hadoop démarrés
- Le code source du mapper (mapper.py)
- Le code source du reducer (reducer.py)
- La commande permettant de tester localement votre programme avec une copie d'écran de son exécution
- La commande Hadoop Streaming utilisée pour exécuter votre job sur le cluster Hadoop avec une copie d'écran de son exécution
- Copies d'écran de toutes les étapes de l'exercice 2, ainsi que les requêtes HQL accompagnées de captures d'écran de leur exécution