Gloire BAYOUNDOULA
INFOA3

Matière : **Bases de données NoSQL**
Partie : **CouchDB**
Enseignant : **Monsieur Samir Youcef**

Rendu TP3

Disclaimer : J'ai réalisé ce TP sur Windows en utilisant Docker desktop. Une fois, Docker desktop installé, il comporte un terminal linux dans lequel on peut utiliser toutes les commandes linux requises pour le TP.

- Documentation officielle de CouchDB : couchdb - Official Image

Introduction à MapReduce et CouchDB

MapReduce est un modèle de programmation utilisé pour le traitement parallèle de grandes quantités de données. Il est particulièrement utilisé dans le domaine du Big Data. CouchDB, un système de gestion de bases de données NoSQL, implémente MapReduce pour effectuer des calculs distribués de manière efficace.

CouchDB est un SGBD NoSQL orienté document avec une API REST qui peut être installé localement ou via Docker. Les documents sont stockés en JSON, et l'identifiant (_id) est unique. Il supporte le versionnement des documents.

Mais on va utiliser le client curl à la place. Mais il y a d'autres clients ou des plugins que l'on peut ajouter à son navigateur également.
Pour afficher la représentation d'une ressource

Installation avec Docker

Pour installer CouchDB avec Docker et mapper les volumes :

```
docker run -d --name couchdb-demo -e COUCHDB_USER=youssef -e COUCHDB_PASSWORD=samir -p 5984:5984 couchdb
```

couchdbdemo : nom pour le conteneur
-e : pour définir une variable d'environnement
Il y'a deux variables d'environnement:

- ➢ COUCHDB_USER : nom d'utilisateur
- ➢ COUCHDB_PASSWORD : mote de passe

Mapping des ports, port d'écoute par défaut de CouchDB : 5984
couchDB : Nom de l'image

Vérifiez que le conteneur est en cours d'exécution : docker ps

1. Commandes MapReduce

MapReduce repose sur deux fonctions principales : **Map** et **Reduce**.

1.1 Fonction Map

La fonction Map est appliquée individuellement à chaque document d'une base de données JSON. Elle extrait des informations et émet des paires (clé, valeur) qui seront ensuite regroupées.

```
// Fonction Map
function (doc) {
  if (doc.type === "user") {
    emit(doc._id, doc.name);
  }
}
```

Explication :

- Cette fonction parcourt chaque document.
- Si le type du document est "user", elle émet l'identifiant du document (doc._id) comme clé et son nom (doc.name) comme valeur.

1.2 Fonction Reduce

La fonction Reduce agrège les résultats produits par la fonction Map.

```
// Fonction Reduce
function (keys, values, rereduce) {
  return values.length;
}
```

Explication :

- Cette fonction prend en entrée une liste de clés et leurs valeurs associées.
- Elle retourne le nombre total de valeurs, permettant par exemple de compter le nombre d'utilisateurs.

2. Commandes CouchDB

CouchDB expose une API REST facilitant la manipulation des bases de données et documents.

Tester la connexion à CouchDB

```
curl -X GET http://youssef:samir@localhost:5984/
```

L'interface graphique est accessible via http://localhost:5984/_utils.

Importation et Manipulation de Données

2.1 Afficher toutes les bases de données

```
curl -X GET http://localhost:5984/_all_dbs
```

Explication :
Cette commande liste toutes les bases de données disponibles sur CouchDB.

2.2 Créer une base de données

```
curl -X PUT http://localhost:5984/ma_base
```
Il renvoie ok si le document est bien créé.

Explication :
Elle crée une nouvelle base de données nommée ma_base.

2.3 Ajouter un document

```
curl -X POST http://localhost:5984/ma_base -H "Content-Type: application/json" -d '{
  "_id": "user_001",
  "type": "user",
  "name": "Jean Dupont",
  "age": 30
}'
```

Explication :

- Ajoute un document avec un identifiant (_id).
- Il contient des champs : type, name, et age.

2.4 Récupérer un document

```
curl -X GET http://localhost:5984/ma_base/user_001
```

Explication :
Cette commande récupère le document ayant _id = user_001.

2.5 Mettre à jour un document

```
curl -X PUT http://localhost:5984/ma_base/user_001 -H "Content-Type: application/json" -d '{
  "_id": "user_001",
  "_rev": "1-xxxxxxxxxxx",
  "type": "user",
  "name": "Jean Dupont",
  "age": 31}'
```

Explication :

- Pour mettre à jour un document, il faut spécifier _rev (révision actuelle).
- Ici, l'âge de l'utilisateur est modifié à **31**.

2.6 Supprimer un document

curl -X DELETE http://localhost:5984/ma_base/user_001?rev=<REV_ID>

Remarque : <REV_ID> est la version du document, obtenue avec la commande GET.

Explication :
Supprime le document user_001 en précisant sa révision (_rev).

2.7 Supprimer une base de données

curl -X DELETE http://localhost:5984/ma_base

Explication :
Supprime complètement la base de données ma_base.

2.8 Lister les documents de la base

curl -X GET http://youssef:samir@localhost:5984/films/_all_docs

Lister les bases existantes
curl -X GET http://youssef:samir@localhost:5984/_all_dbs

2.9 Insérer un document sans préciser l'ID (génération automatique)

curl -X POST http://youssef:samir@localhost:5984/films -d '{"title": "Matrix", "year": 1999, "genre": "Action"}' -H "Content-Type: application/json"

Insertion d'une collection de documents (bulk insert)

En supposant que nous ayons un fichier films_couchdb.json :

curl -X POST http://youssef:samir@localhost:5984/films/_bulk_docs -d @films_couchdb.json -H "Content-Type: application/json"

3. Exemple d'application : Nombre de films par année

Prenons un exemple pratique où l'on veut calculer le nombre de films par année.

3.1 Importation des données

```
curl -X POST http://localhost:5984/films/_bulk_docs -H "Content-Type: application/json" -d @films.json
```

Explication :

- Cette commande importe une collection de films dans la base films.
- films.json contient un ensemble de documents JSON représentant des films.

3.2 Fonction Map pour extraire les films par année

```
function (doc) {
  if (doc.year && doc.title) {
    emit(doc.year, doc.title);
  }
}
```

Explication :

- Cette fonction extrait l'année (doc.year) et le titre (doc.title) de chaque film.

3.3 Fonction Reduce pour compter le nombre de films par année

```
function (keys, values, rereduce) {
  return values.length;
}
```

Explication :
Cette fonction compte combien de films sont associés à chaque année.

Conclusion

Grâce à CouchDB et MapReduce, il est possible d'effectuer des traitements massivement parallèles sur des documents JSON. L'utilisation des fonctions Map et Reduce permet d'extraire, transformer et agréger des données de manière efficace, particulièrement utile pour l'analyse de données volumineuses.