

MIF04 GDW – TP

MAP/REDUCE sur MONGODB – partie 1

Résumé

Cette première partie du TP a le double objectif de découvrir MONGODB et de mettre en œuvre des traitements Map/Reduce en javascript. La seconde partie du TP, dans un autre sujet, sera à rendre.

1 Environnement de travail

1.1 Documentation de référence

- <http://docs.mongodb.org/manual/core/map-reduce/>
- <http://docs.mongodb.org/manual/reference/mongo-shell/>
- <https://docs.mongodb.com/manual/reference/method/db.collection.mapReduce/#db.collection.mapReduce>

1.2 Serveur

Le serveur MONGODB déployé sur la plate-forme de cloud¹ est accessible à l'adresse IP 192.168.74.37:27017. Vous utiliserez le client en ligne de commande mongo depuis Linux². Vous vous authentifierez avec l'utilisateur mif04 et le mot de passe m2i0f1084. Vous travaillerez sur les collections de la base de données mif04. Pour travailler sur votre propre serveur, vous pouvez télécharger les jeux de données et les importer avec les commandes suivantes :

```
wget http://media.mongodb.org/zips.json
wget https://raw.githubusercontent.com/ozlerhakan/mongodb-json-files/master/datasets/grades
wget https://raw.githubusercontent.com/mongodb/docs-assets/geospatial/restaurants.json
wget https://raw.githubusercontent.com/mongodb/docs-assets/geospatial/neighborhoods.json

mongoimport --host localhost --db mif04 --collection zips --drop --file zips.json
mongoimport --host localhost --db mif04 --collection grades --drop --file grades.json
mongoimport --host localhost --db mif04 --collection restaurants --drop --file restaurants
mongoimport --host localhost --db mif04 --collection neighborhoods --drop --file neighborh
```

2 Prise en main

Identifiez vous avec la commande suivante :

```
> mongo -u "mif04" -p "m2i0f1084" --authenticationDatabase "mif04" 192.168.74.37/mif04
```

1. accès depuis le campus uniquement, utiliser impérativement eduroam en wifi ou un tunnel ssh pour un accès depuis l'extérieur.

2. Pour l'installation <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/>

Exercice 1 : premiers pas en MongoDB

1. Exécutez les commande `show collections` et `show dbs`. Que font elles ?
2. Trouvez la commande MONGODB pour afficher *un document* de la collection `zips`. Expliquer la structure du document trouvé.
3. Trouvez la commande MONGODB pour calculer *le nombre de documents* de la collection `zips`.
4. Reprendre les questions précédentes avec les collections `grades` et `restaurants`.
5. Trouvez la commande MONGODB qui permet de donner la liste des *zips* de l'état de Californie.
6. Même question que précédemment, mais cette fois vous ne voulez garder dans le résultat que le nom de la ville et *aucun autre champ*.
7. Trouvez la commande MONGODB qui permet de lister les *zips* dont la population est supérieure à 100.000 habitants. Même question avec *le nombre de zips* dont la population est supérieure à 100.000 habitants.

3 Découverte de Map/Reduce

Dans toute la suite de ce TP, vous devrez répondre aux exercices avec des requêtes MAP/REDUCE. Vous n'avez pas les droits d'écriture sur le serveur, il faut donc afficher le résultat de vos jobs dans la console et vous devrez résoudre chaque exercice *avec un seul job*.

Télécharger le fichier `MIF04-GDW-TPMongoDB-1-Init.js`³ et consultez le. Ensuite, chargez le fichier avec la commande suivante :

```
> load("MIF04-GDW-TPMongoDB-1-Init.js");
```

Exercice 2 : premier comptage en Map/Reduce

Exécutez la commande `db.zips.mapReduce(init_map, init_red, {out : {inline:1}});`.

1. Expliquez ce que fait la ligne `emit(0, 1);`. Le résultat est-il le même si on la remplace par `emit("obiwan kenobi", 1);` ? Justifier. Que conseillerez vous au final comme constante à choisir ?
2. Qu'est-ce que ce job MAP/REDUCE calcule ? Exécutez ce job sur d'autres collections et vérifiez le résultat.
3. Que contient le champs `counts` de l'objet retourné ? Expliquez.
4. Exécutez maintenant `db.zips.mapReduce(init_map, bogus_red, {out : {inline:1}});`. Quel est le problème ? Justifiez⁴.
5. Écrire un job MAP/REDUCE qui, pour chaque état des USA, calcule sa population. Vérifiez la cohérence du résultat, sur par exemple la population de la Californie.
6. Écrire un job MAP/REDUCE comme précédemment qui, pour chaque état des USA, donne le nombre de *zips* dont la population est supérieur à 100.000.
7. Reprendre la question précédente en utilisant le champ `query` de `mapReduce`. Que constatez vous ? Expliquez.

3. <https://perso.liris.cnrs.fr/romuald.thion/files/Enseignement/MIF04/MIF04-GDW-TPMongoDB-1-Init.js>

4. <https://docs.mongodb.com/manual/reference/method/db.collection.mapReduce/#requirements-for-the-reduce-function>