Compte rendu du TP1 d'INFO 834

Maxime PAULIN Ryan RISS

February 5, 2023

Préface

Nous avons installé mongodb sur un vps que nous possèdons pour que cela soit plus pratique a travailler en groupe de 2. Le serveur est situé à Torronto, ce qui ajoute 100ms de ping par requête.

6 Curseurs

```
let q6 = async () => {
        let cursor = collections.communes.find({});
        let timeJson = {};
        let time = Date.now();
        while(await cursor.hasNext()){
                console.log(await cursor.next());
        }
}
```

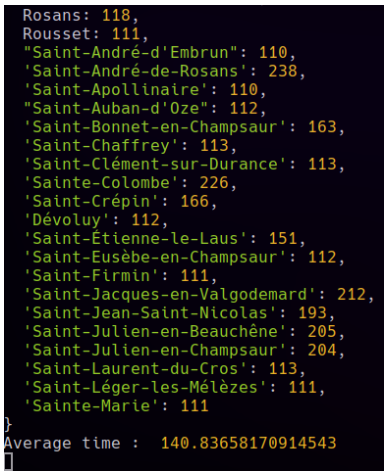
On récupère l'intégralité de la base de donnée. C'est un peu long, mais ca marche.

7 Benchmark

```
let q7 = async () => {
        //find all towns
        let townList = [];
        let cursor = collections.communes.find({});
        while(await cursor.hasNext()){
                townList.push((await cursor.next()).nom_commune);
        }

        let timeJson = {};//for usefull data holder
        let timeList = [];//for handy time calc
        for(let town of townList){
                let time = Date.now();

                if(timeList.length > 2000) break;
                //150ms * 39000 ~ 2hours, cant be bothered
                await collections.communes.findOne({nom_commune: town})
                timeJson[town] = Date.now() - time;
                timeList.push(timeJson[town]);
                console.log(town, ":", timeJson[town])
                }

        let total = 0;
        for(let k of timeList) {//calc of average
                        total += k;
        }
        console.log("Average time :", total / timeList.length);
        //140ms, but my server is hosted in Torronto (approx 110ms ping
           )
        //110 when adding an index to the db
        //after indexing : 112 ms
```

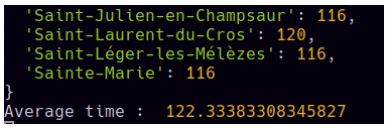
}

On trouve en moyenne 140ms par ville :



8 Index

En utilisant un index :



9 $inc et .update

```
db.freq.update({_id : ObjectId("63e008bacc12ed9b5d0c54f8")}, {$inc:
{nb: 1}
})
```

```
db.freq.find().pretty()
```

_id	acknowledged	insertedId	matchedCount	modifiedCount	upsertedCount
	true	null	1.0	1.0	0.0

Vérification du résultat

_id	url	nb
63e008bacc12ed...	https://www.g...	3

10 Array dans les documents

Insert:

```
list = {name : "mailing-list", emails: []}

db.lists.insert(list)
```

pull:

```
db.lists.update({_id: ObjectId("63e010a0de1fd3a8787ce5f3")},
{$pull: {emails: "mokhilss.bitcoin@univ-smb.fr"}}
)
```

result pull:

lists > name

_id	name	emails
63e010a0de1fd3...	mailing-list	[1 elements]

updates:

```
db.lists.update({_id: ObjectId("63e010a0de1fd3a8787ce5f3")},
{$push: {emails: "Marc-Philippe.Huget@univ-smb.fr"}}
)
db.lists.update({_id: ObjectId("63e010a0de1fd3a8787ce5f3")},
{$push: {emails: "mokhilss.bitcoin@univ-smb.fr"}}
)
```

results:

lists > emails > 0

{Document id}	0
63e010a0de1fd3...	Marc-Philippe....

lists > name

_id	name	emails
63e010a0de1fd3...	mailing-list	[2 elements]

11 Liaison de document

On insert, puis drop et remplace:

```
doc1 = {name: "user1", firstname: "a", email: "a@ok.fr"}
doc2 = {name: "user2", firstname: "b", email: "b@ok.fr"}
doc3 = {name: "user3", firstname: "c", email: "c@ok.fr"}

db.users.insert(doc1)
db.users.insert(doc2)
db.users.insert(doc3)
```

```
db.lists.drop()

list = {"name": "mailing-list", "users": []}

db.lists.insert(list)
```

On vérifie :

```
let cursor = db.users.find({});

while(cursor.hasNext())
{
    userID = (cursor.next())._id;
    db.lists.update({_id: ObjectId("63e0174fde1fd3a8787ce5f7")},
    {$push: {users: userID}}
    )
}
```

```
{
    "_id" : ObjectId("63e0174fde1fd3a8787ce5f7"),
    "name" : "mailing-list",
    "users" : [
        ObjectId("63e0138ade1fd3a8787ce5f4"),
        ObjectId("63e0138ade1fd3a8787ce5f5"),
        ObjectId("63e0138ade1fd3a8787ce5f6")
    ]
}
```

En python:

```python
from pymongo import MongoClient

# Connexion à la base de données
client = MongoClient("mongodb://ryan:XXX@170.75.165.66:27017/mailing")

# Sélection de la base de données et des collections
db = client["mailing"]
lists = db["lists"]
users = db["users"]

# Récupération des mailing lists
mailingLists = lists.find()

# Boucle
for document in mailingLists:
    usersID = document['users']
    for userID in usersID:
        user = users.find({"_id": userID})
        print(list(user))
```

résultat:

```
[{'_id': ObjectId('63e0138ade1fd3a8787ce5f4'), 'name': 'user1', 'firstname': 'a', 'email': 'a@ok.fr'}]
[{'_id': ObjectId('63e0138ade1fd3a8787ce5f5'), 'name': 'user2', 'firstname': 'b', 'email': 'b@ok.fr'}]
[{'_id': ObjectId('63e0138ade1fd3a8787ce5f6'), 'name': 'user3', 'firstname': 'c', 'email': 'c@ok.fr'}]
```

12 Authentification

Nous avons commencé le TP par protèger la base de donnée. Chaque utilisateur (sauf ceux de maxime, qui sont admin) n'as les droits que pour sa base. Voici ce qu'on obtient dès que quelqu'un se connecte anonymement :

```
test> use France
switched to db France
France> show collections
MongoServerError: command listCollections requires authentication
France> 
```

13 Map Reducer

```
France> db.mapReducerOut.find().sort({value: -1})
[
  { _id: 'Passy', value: 9 },
  { _id: 'Entrelacs', value: 7 },
  { _id: 'Bessenay', value: 6 },
  { _id: 'Léchère', value: 6 },
  { _id: 'Aime-la-Plagne', value: 6 },
  { _id: "Val d'Arcomie", value: 5 },
  { _id: 'Satolas-et-Bonce', value: 5 },
  { _id: 'Beaulieu', value: 5 },
  { _id: 'Saint-Étienne', value: 5 },
  { _id: 'Saint-Genès-Champanelle', value: 5 },
  { _id: 'Plagne Tarentaise', value: 5 },
  { _id: 'Thizy-les-Bourgs', value: 5 },
  { _id: 'Belleville', value: 5 },
  { _id: 'Saint-Chamond', value: 4 },
  { _id: 'Béligneux', value: 4 },
  { _id: 'Vans', value: 4 },
  { _id: 'Beaumont', value: 4 },
  { _id: 'Saint-Hilaire', value: 4 },
  { _id: 'Sainte-Catherine', value: 4 },
  { _id: 'Colombier-Saugnieu', value: 4 }
]
```

14 Backup and Dump

On fait un backup :

```
$mongodump --u USERNAMETOHIDE --p PASSWORDTHATISVERYSAFE ---out=/tmp
2023-02-05T21:32:32.923+0000    writing admin.system.users to /tmp/
    admin/system.users.bson
2023-02-05T21:32:32.924+0000    done dumping admin.system.users (7
    documents)
2023-02-05T21:32:32.925+0000    writing admin.system.version to /tmp/
    admin/system.version.bson
2023-02-05T21:32:32.926+0000    done dumping admin.system.version (2
    documents)
2023-02-05T21:32:32.929+0000    writing France.communes to /tmp/France/
    communes.bson
2023-02-05T21:32:32.931+0000    writing France.yes to /tmp/France/yes.
    bson
2023-02-05T21:32:32.934+0000    writing France.mapReducerOut to /tmp/
    France/mapReducerOut.bson
2023-02-05T21:32:32.937+0000    writing France.orders to /tmp/France/
    orders.bson
2023-02-05T21:32:32.950+0000    done dumping France.orders (10
    documents)
2023-02-05T21:32:32.959+0000    done dumping France.mapReducerOut (4020
     documents)
2023-02-05T21:32:32.959+0000    writing France.map\_reduce\_example to
    /tmp/France/map\_reduce\_example.bson
2023-02-05T21:32:32.968+0000    done dumping France.yes (4020 documents
    )
2023-02-05T21:32:32.968+0000    writing mailing.users to /tmp/mailing/
    users.bson
2023-02-05T21:32:32.971+0000    done dumping France.map\_reduce\
    _example (4 documents)
2023-02-05T21:32:32.971+0000    writing Analytics.freq to /tmp/
    Analytics/freq.bson
2023-02-05T21:32:32.971+0000    writing mailing.lists to /tmp/mailing/
    lists.bson
2023-02-05T21:32:32.972+0000    done dumping mailing.users (3 documents
    )
2023-02-05T21:32:32.973+0000    done dumping Analytics.freq (1 document
    )
2023-02-05T21:32:32.974+0000    writing France.map reduce example to /
    tmp/France/map+reduce+example.bson
2023-02-05T21:32:32.974+0000    done dumping mailing.lists (1 document)
2023-02-05T21:32:32.975+0000    done dumping France.map reduce example
    (0 documents)
2023-02-05T21:32:33.148+0000    done dumping France.communes (39201
    documents)
```

On le recharge:

```
$sudo mongorestore --u STILLAGOODUSERNAME --p EPICPASSWORD /tmp/Analytics
    /freq.bson
```

```
2023-02-05T21:38:41.589+0000    WARNING: On some systems, a password
    provided directly using --password may be visible to system status
    programs such as `ps` that may be invoked by other users. Consider
    omitting the password to provide it via stdin, orusing the --config
     option to specify a configuration file with the password.
2023-02-05T21:38:41.606+0000    checking for collection data in /tmp/
    Analytics/freq.bson
2023-02-05T21:38:41.606+0000    reading metadata for Analytics.freq
    from /tmp/Analytics/freq.metadata.json
2023-02-05T21:38:41.607+0000    restoring to existing collection
    Analytics.freq without dropping
2023-02-05T21:38:41.607+0000    restoring Analytics.freq from /tmp/
    Analytics/freq.bson
2023-02-05T21:38:41.613+0000    continuing through error: E11000
    duplicate keyerror collection: Analytics.freq index: _id_ dup key:
    { _id: ObjectId('63e008bacc12ed9b5d0c54f8') }
2023-02-05T21:38:41.649+0000    finished restoring Analytics.freq (0
    documents, 1 failure)
2023-02-05T21:38:41.649+0000    no indexes to restore for collection
    Analytics.freq
2023-02-05T21:38:41.649+0000    0 document(s) restored successfully.
1 document(s) failed to restore.
```

Ici, on ne recharge pas, car la base de donnée tourne déjà avec les élements charger, et n'authorise pas les duplicat d'id.