TP 4 : NoSQL –MongoDB

Réplication

Nous voulons ici tester le système de tolérance aux pannes de MongoDB. Pour cela, nous allons lancer des serveurs avec des ReplicaSets.

1. Installation d'un replicaSet

Nous allons créer ici un replicaSet composé de 5 serveurs, dont 1 primaire, 3 secondaires et un arbitre.

1. À chaque serveur, un répertoire de données doit être créé :

c :/data/rs1
c :/data/rs2
c :/data/rs3
c :/data/arb

2. Choisir un nom de réplication ReplicaSet: **RS0**

3. Un port d'écoute pour chaque serveur : 27018 (à incrémenter)

4. Ouvrir 4 nouvelles consoles dos (cmd), chacune pour un serveur

5. Dans chaque console dos, lancer un replicat avec un nouveau port de connexion :

```
mongod --replSet rs0 --port 27018 --dbpath /data/RS1

mongod --replSet rs0 --port 27019 --dbpath c:/data/RS2

mongod --replSet rs0 --port 27020 –dbpath c:/data/RS3
```

6. Ouvrir une nouvelle console pour se connecter au serveur qui sera choisi comme serveur primaire:

```
mongo --port 27018
```

7. Lancer le mode de réplication (initialiser replicaset):

```
rs.initiate();
```

Dans la console "mongo" apparaîtra **RS1:PRIMARY >** désignant le serveur primaire. Votre serveur est le premier membre du ReplicaSet "rs0", il a récupéré 1 voix (la sienne) pour devenir *PRIMARY*, toutefois, sans arbitre. Vous pouvez d'ailleurs constater dans votre console que le serveur est bien primary pour rs0 .

8. Rajouter chaque réplicat (en associant le "host" de votre machine) :

```
rs.add(" <host>:27018");
rs.add("<host>:27019");
```

rs.add("<host>:27020");
host : localhost ou 127….(si vous travaillez en local)

9. Vérifier la configuration de l'ensemble des serveurs :
Le ReplicaSet a été initié. Vous pouvez d'ailleurs connaître le nom réseau de votre machine dans la clé "**me**" (votre cas : local). Regardons la configuration du *ReplicaSet* :
rs.conf();

10. **P**our départager les votes (nombre pair de serveur par exemple), un arbitre sera nécessaire pour désigner le *PRIMARY*. Lorsqu'il y a plusieurs réplicats, il est donc nécessaire de définir un **arbitre** dans le cas où le serveur primaire tombe en panne. Ouvrir une nouvelle console :
en spécifiant le port 30000 :

Pour ce faire : Utiliser le répertoire **/data/arb** (C:\data\arb sous Windows)

mongod -replSet RS1 -dbpath /data/arb -port 30000

11. Ajouter l'arbitre dans la console d'admibistration RS1:PRIMARY>,

rs.addArb(" <host>:30000");
Pour consulter le statut actuel du ReplicaSet, il suffit de faire l'instruction suivante :
Rs.statut()
Qu est ce que vous constatez ?

II- Test de réplication

Pour tester la tolérance aux pannes, importons un jeu de données ex votre base de donnée " MovieLens" (*mongoimport*) comme dans le premier tp, en utilisant le port du serveur *PRIMARY* (ici 27020). Vérifions l'existence des données avec une requête.

Simulons maintenant une panne du serveur *PRIMARY* en fermant le processus correspondant (dans le cas présent, le serveur ayant le port 27020). Vous pourrez alors constater qu'un nouveau serveur est élu *PRIMARY* (cette opération peut prendre un peu de temps).

Après connexion au nouveau *PRIMARY*, vous pourrez constater que votre requête retourne toujours le même résultat. Vous êtes tolérant aux pannes avec votre premier *ReplicaSet* !

1. Dans la console RS1:PRIMARY> vérifier le contenu de la BD " MovieLens " :
use MovieLens;
db.movies.count();

2. Ouvrir une nouvelle console sur un serveur réplicat : mongo -port 27018
Dans la console mongo apparaîtra RS1:SECONDARY> désignant le serveur secondaire (réplicat).

3. La commande :
use **MovieLens**;
db. **movies**count();
que se passe-t-il... ?
retourne une erreur car l'utilisation d'un réplicat n'a pas été autorisé. Pour ce faire, taper la commande :

rs.slaveOk();
Refaire le test

4. Dans la console RS1:PRIMARY>, ajouter un document ex :"{ test : 1}"
5. Dans la console RS1:SECONDARY>, vérifier son existance :
use **MovieLens**;
db.movies.find({test:1});
Cela peut prendre quelques secondes (souvent quelques milli-secondes)

III- Test de tolérance aux pannes

1. Dans la console RS1:PRIMARY>, ajouter un document ex "{ test : 2}" à la collection *movies*
2. Tuer le processus mongod "primaire" (console ouvrant mongod avec le port 27017) avec **"ctrl+C"**
Ou dans la console RS1:PRIMARY>, taper :
use admin; puis
db.shutdownServer();

3. Regarder le comportement des autres mongod, en particulier l'arbitre (port 30000). Sera alors désigné le serveur qui deviendra le primaire.
Il est aussi possible de connaître le serveur primaire dans la console RS1:SECONDARY> :
rs.status();
4. Dans la console RS1:PRIMARY>, taper 'exit'. Relancer "mongo" mais avec le port du nouveau serveur primaire
5. Vérifier l'existence de votre document dans la collection movies :
use **MovieLens**;
db.movies.find({test:2});

IV Fichier de configuration (facultatif)

Nous souhaitons automatiser le lancement d'un serveur réplicat. Pour cela, nous allons créer un fichier de configuration (/data/mongo1.conf) Il est nécessaire de créer un fichier de configuration par serveur (chaque serveur en théorie est sur une machine distincte).
Voici à quoi ressemble un fichier de configuration :

```
# mongo1.conf
#Fichier de log du serveur. Le répertoire doit être créé et mongo doit avoir les droits d'écriture.
#Un fichier de log par serveur
logpath=/var/log/mongodb/mongod1.log
logappend=true
# Permet de reprendre la main dans le terminal lorsque le serveur est lancé
fork=true
# Paramètres de connexion et de stockage à changer pour chaque serveur
bind_ip=127.0.0.1
port=27017
dbpath=/data/db
# Fichier créé pour vérifier si le serveur tourne
pidfilepath=/var/run/mongodb/mongod.pid
# Niveau de "oplog" pour la réplication
# 0=off (default), 1=W, 2=R, 3=both, 7=W+some reads
diaglog=1
# Replication : Mettre le nom du replicaSet
replSet=RS1
# Taille maximum de l"oplog" pour la réplication
oplogSize=1024
```

L'ensemble des options du fichier de configuration se trouve ici :
http://docs.mongodb.org/manual/reference/configuration-options/
1. Créer un fichier de configuration pour chaque serveur
2. Lancer chaque serveur en utilisant le fichier de configuration correspondant :

mongod -config /data/mongo1.conf
3. Dans le cas où le replicaSet ne serait pas initialisé (ce qui n’est pas le cas actuellement), il est
possible d’initialiser avec l’ensemble des serveurs :

```
rsconf = {
_id: "RS1",
members: [
{_id: 0,host: "<hostname>:27017"},
{_id: 1,host: "<hostname>:27018"},
{_id: 2,host: "<hostname>:27019"},
{_id: 3,host: "<hostname>:27020"},
{_id: 4,host: "<hostname>:30000",arbiterOnly:true},
]
};
rs.initiate(rsconf);
```

http://docs.mongodb.org/manual
http://camillepradel.fr/
http://b3d.bdpedia.fr/