# Rapport de TP - Cassandra DB

loïc divad

Mai 2015

## 1 Introduction

...

## 2 Installation

Pour réaliser l'installation du logiciel avec la dernière version en date: *2.1.7*,
on procède d'une manière différente en indiquant toute les étapes.

```
lmdadm $ wget ftp://mirrors.ircam.fr/.../apache-cassandra-2.1.7-bin.tar.gz
lmdadm $ tar -zxvf apache-cassandra-2.1.7-bin.tar.gz
```

On redirige en suite les logs en éditant le fichier *logback.xml*.

```
...
<file> /home/lmdadm/log/cassandra/system.log </file>
...
```

Puis on édite le fichier *cassandra.yalm*. La liste suivante présente tous les paramètres personnalisés dans le cadre du TP:

- cluster_name: Cassandra Cluster
- data_file_directories: /home/lmdadm/data/cassandra
- commitlog_directory: /home/lmdadm/log/cassandra/commitlog
- saved_caches_directory: /home/lmdadm/tmp/cassandra/saved_caches

On lance ensuite la base de donnée. On note que l'option indiquée dans le TP -f signifie: *force foreground*. Conclusion, nous n'utiliserons surtout pas cette option. Après l'avoir lancé en mode démon, on ouvre la console:

```
lmdadm $ ./bin/cassandra
lmdadm $ ./bin/cqlsh
```

### 2.1 Logiciel d'administration

Il existe un bon nombre de logiciel d'administration pour cassandra, la pluspart propulsés par des communautés opensources.

Le logiciel retenu pour réaliser ce TP est la dernière version (*1.3.1*) de DevCenter édité par DataStax. Contrairement à d'autres logiciels essayés (OpsCenter, helenos) DevCenter est un client lourd, on n'y accède pas par un navigateur web. Il ne sera donc pas installé sur le serveur mais sur notre ordinateur personnel. On ouvre en suite une connection vers le Cleusteur.

*téléchargement:* http://www.datastax.com/download-ops-dev

Une fois le logiciel installé et la connection enregistrée (figure 1.) on demande à Cassandra d'accepter les connections à distance, les 'remote connections '. Pour cela on édite de nouveu le fichier *casssandra.yaml*.

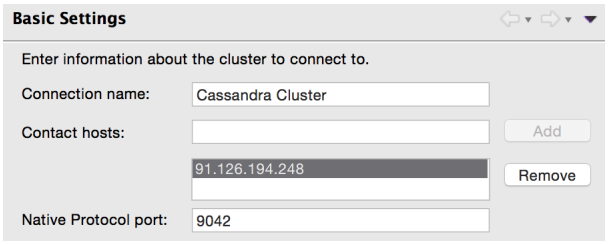


Figure 1: Ajout du serveur sur le quel tourne Cassandra.

- rpc_address: 0.0.0.0 (Anciennement localhost)
- broadcast_rpc_address: 1.2.3.4

Une fois le tout configuré il est possible d'accéder à l'ensemble des keyspaces. Et de réaliser des requêtes sur la base. Nour reviendrons sur cette interface pour illustrer des différents points du TP.

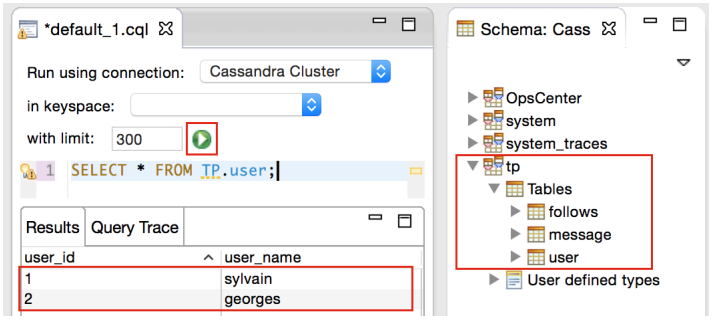


Figure 2: Exemple d'interface DevCenter par DataStax.

# 3 Base de donnée

## 3.1 Description de l'espace de clef

Pour mieux comprendre la structure de donnée dans cassandra cette partie propose un petit point sur ce qu'est un espace de clef.

```
cqlsh> CREATE KEYSPACE TP WITH REPLICATION = {
'class' : 'SimpleStrategy',
'replication_factor' : '1'
};
```

**Definition**(Keyspace): Il s'aggit de la plus grande structure de données de cassandra. Elle est comparable aux bases (ou schéma) dans les systemes classiques ou dans les systèmes document (mongodb, couchdb). Elle contient des familles de colonnes, qui elles, représentent la séparation entre les tables (mysql) ou les collections (mongodb). Chaque ligne insérée présente des informations pour certaines colonnes d'un famille de colonnes.

*Propriétées(Keyspace):*

- **Replication factor**: définit le nombre de noeuds possédant un copie de la donnée.
- **Replica placement strategy**: il s'agit du mode de répartition des réplicats. Il en existe trois. simple strategy, old network topology strategy, network topology strategy

- **Column families**: Liste des familles des colonnes associées. Dans notre cas sur le keyspace TP il y a user, follows et message.

La commande de création du Keyspace dans TP signifie donc: créer l'espace de clefs TP dont les lignes ne sont présentes que sur un noeud. Si il doit y avoir de la réplication le *partitioner* rédistrubura automatiquement sur le noeud suivant.
La commande DESCRIBE nous permet d'avoir la liste de keyspaces.

```
cqlsh> DESCRIBE KEYSPACES;
system_traces OpsCenter system tp
```

On peut ensuite accéder aux familles de colonnes sur un espace de clefs particulier.

```
cqlsh> use TP;
cqlsh:tp> DESCRIBE TABLES;
follows message user
```

On peut ensuite retrouver les colonnes d'une famille en particulier.

```
cqlsh:tp> DESCRIBE TABLE user;
CREATE TABLE tp.user (
   user_id bigint PRIMARY KEY,
   user_name text
)
```

```
~/products/cassandra/bin • lmdadm $>./cqlsh
Connected to Cassandra Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 2.1.7 | CQL spec 3.2.0 | Native protocol v3]
Use HELP for help.
cqlsh> DESCRIBE KEYSPACES;

system_traces  "OpsCenter"  system  tp

cqlsh> use TP;
cqlsh:tp> DESCRIBE TABLES;

follows  message  user

cqlsh:tp> SELECT * FROM TP.user;

 user_id | user_name
---------+-----------
       2 |   georges
       1 |   sylvain

(2 rows)
cqlsh:tp>
```

Figure 3: Quelques commandes cql.

## 3.2 le CQL, langage de requête Cassandra

Dans cette partie on exécutes les commandes présentes dans le TP pour crée la base de données qui sera utilisée par la suite.

*remarque:* On note que l'utilisation de double quotes (") dans la ligne de commande provoque l'erreur:

```
SyntaxException:  <ErrorMessage code=2000 [Syntax error in CQL query] message="line
1:58 extraneous input hello world expecting ) (...)  values (1,1 hello world[])]...)>
```

## 3.3 Clefs composites

Via Devcenter on insère les lignes demandées comme sur les figures suivantes.

Puis on vérifie la table des relations d'abonnement 'follows'. La premier relation d'abonnement (georges suit sylvain) à été supprimée. En effet, lors de la définition de la

```
1 INSERT INTO user (user_id, user_name) VALUES (3, 'patrick');
2
3 INSERT INTO follows (follower_id, followed_id) VALUES (2, 3);
```

Figure 4: Ajout de l'utilisateur patrick suivit par georges.

famille de clef follows, la colonne follower_id est déclarée comme une clef primaire, elle est donc unique.
`CREATE TABLE follows(follower_id bigint PRIMARY KEY, followed_id bigint);`
Autrement dit, on ne peut suivre qu'une seule personne à la fois.

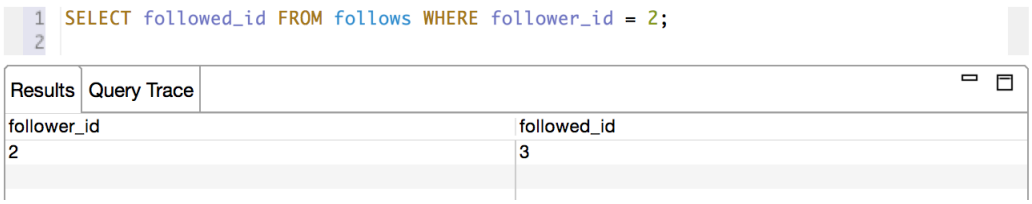


Figure 5: Unique relation de la table follows après l'ajout de patrick.

A l'aide de la commande indiquée dans l'énoncé du tp on défini un couple de clef. Désormais, c'est la relation suivit-abonné qui est unique. On ne peut suivre quelqu'un qu'une seul fois. On note que seul 3 ligne ont été inséré car les deux dernier INSERT sont des duplicats.

```
INSERT INTO follows (follower_id, followed_id) VALUES (2,3);
INSERT INTO follows (follower_id, followed_id) VALUES (2,1);
INSERT INTO follows (follower_id, followed_id) VALUES (3,1);
INSERT INTO follows (follower_id, followed_id) VALUES (3,1);
```

Figure 6: Série d'insertion d'abonnement.

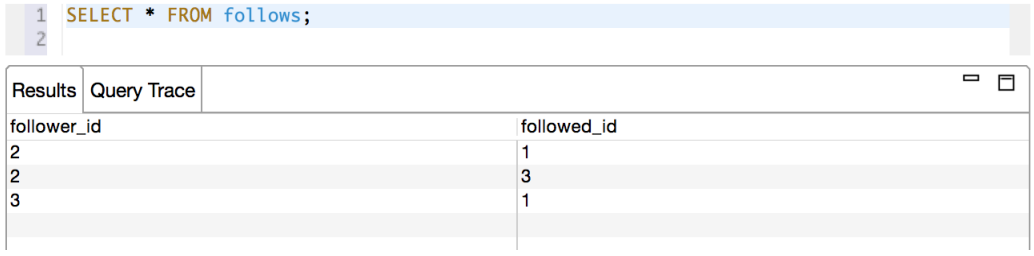


Figure 7: Relevé des différents abonnements.

# 4 Cas pratiques

## 4.1 Requêtes cql

Dans un premier temps on insersert quelques messages pour essayer la requête. Voici une liste des messages présents:
`SELECT * FROM tp.message;`

En cherchant la reqête qui affiche les messages on découvre que Cassandra restreint la clause WHERE aux champs indexés (C'est à dire aux clefs, primaire et autres...)

Ni les rêquetes imbriquées ni les jointures ne sont pérmises en CQL. On propose donc l'implémentation suivante en 2 étapes:

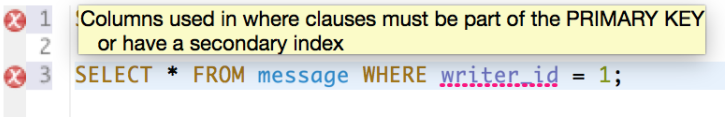


Figure 8: Message d'ereur suite à une clause WHERE.

- Recréer une clef composite avec msg_id et writer_id comme à la question suivante.
- Récupérer dans une reqête ultérieur les identifiants des utilisateurs suivits par george.

On obtient le résultat suivant:

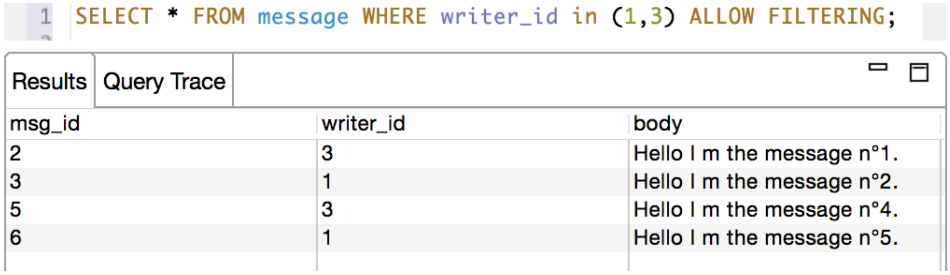


Figure 9: Time line de George.

*Conclusion:* Pour plusieur raison l'oppération proposée n'est pas satiffaisante. Dans un premier temps le changement de contrainte autorise des messages de même identifiant si l'émetteur du message est différent. La manque de requête imbriquée nous force à écrire les attributs de la clause WHERE à la main (1,3). Et enfin même avec cette astuce Cassandra indique que cette requête ne peut pas être effectuée pour des raisons de performance. On doit alors autoriser le filtrage à l'aide de l'instruction ALLOW FILTERING.

En somme la conclusion est que ce problème, comme beacoup de problème en NoSQL, doit être géré applicativement. C'est à dire dans le code de l'applicatiion qui fait appel à Cassandra et non dans la partie persistence du système.

## 4.2 Mini Twitter