

## Cassandra

Cassandra est une base de données NoSQL orientée colonnes. Elle permet de stocker une large quantité de données. Elle permet la scalabilité horizontale : possibilité d'ajouter continuellement de nouvelles machines pour répondre aux besoins.

Cassandra a été fondée initialement par Facebook. En 2008, Facebook transmet ce projet à la fondation Apache. Aujourd'hui, de nombreux acteurs du Web l'utilisent. On site Netflix, Spotify, eBay et Twitter.

Quand on parle de base de données NoSQL, on parle du théorème de CAP :

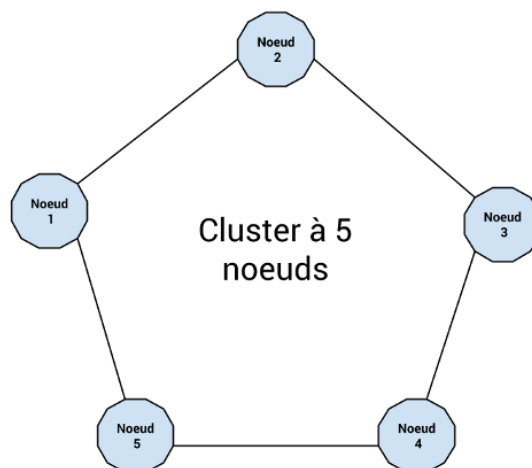
1. La cohérence (**C**onsistency)
2. La disponibilité (**A**vailability)
3. La tolérance aux pannes (**P**artition-tolerance)

On connaît aussi, très bien, qu'on ne peut assurer que deux parmi les trois paramètres.

Cassandra assure la disponibilité et la tolérance aux pannes en sacrifiant la cohérence. C'est-à-dire que les données seront éventuellement, avec le temps, cohérentes.

### Architecture de Cassandra

Cassandra est une base de données distribuée sur plusieurs serveurs appelés nœuds. Les données sont distribuées entre ces nœuds. Ces nœuds sont tous égaux : pas de notions maître/esclave. Ceux-ci communiquent ensemble pour la gestion des données. Le regroupement de nœuds est appelé un cluster.



Comment les données sont-elles distribuées entre les nœuds?

En pratique, à chaque opération de modification ou de lecture est associée une clé appelée clé de partition. Il y a deux manières de distribuer les données :

- 1) De manière ordonnée, chaque nœud prend en charge une plage de clé de partition triée par ordre croissant. Par exemple, si on est en train de stocker des données relatives à des personnes, et si, on a choisi le nom des personnes comme clé de partition, le premier nœud stockera les noms qui commencent par les lettres A jusqu'à D. le second nœud stockera les noms qui commencent par les lettres E à H etc... Cependant, cette méthode ne fournit pas une distribution égale de données entre les nœuds d'un cluster. Donc, pour une meilleure distribution des données, la deuxième méthode est plus conseillée.
- 2) De manière aléatoire, chaque nœud prend en charge une plage de la clé de partition distribuée uniformément. En effet, une opération de hachage est effectuée sur cette clé. Le résultat de cette opération est appelée token. Selon l'opération de hachage utilisée, le token tombera dans un intervalle déterminé fixe (par exemple, le résultat du hachage md5 tombe toujours dans l'intervalle  $[0 - (2^{127} - 1)]$ ). Cet intervalle sera divisé en plages et chaque nœud prendra une plage de cet intervalle. Ainsi, à l'arrivée d'une information, on effectue l'opération de hachage sur sa clé de partition ; et d'après le résultat obtenu, on détermine dans quelle plage le résultat se trouve, donc dans quel nœud l'information doit être placée.

Indépendamment de la manière choisie, un nœud au hasard reçoit une requête (lecture ou écriture). Ce nœud n'exécute pas nécessairement la requête en local car le hash de la clé de partition ne tombe pas forcément dans la plage de hash dont il est responsable. Il joue seulement le rôle de nœud coordinateur. Il reçoit la requête, fait le calcul nécessaire (le hachage) et envoie la donnée au nœud correspondant. A noter que chaque nœud du cluster peut à chaque moment être un nœud coordinateur. Puisqu'il n'y a pas de nœud maître et tous les nœuds sont égaux, les données arrivent au hasard chez l'un des nœuds. Celui-ci devient coordinateur juste pour rediriger les données au nœud correspondant.

Les nœuds dans un cluster communiquent périodiquement ensemble pour savoir, en tout moment, l'état et la disponibilité de chacun. Ils connaissent aussi d'autres informations comme la plage que s'occupe chacun des nœuds pour savoir à qui envoyer les données s'ils étaient des nœuds coordinateurs. Ce processus s'exécute toutes les secondes. Chaque nœud possède un fichier de configuration où ils stockent les informations concernant les autres.

Pour une bonne tolérance aux pannes, Cassandra réplique ses données sur les nœuds de cluster. On définit, initialement, le nombre de réplication (= nombre de nœuds) appelé facteur de réplication (RF). Si RF est égal à 1 c.à.d. on fait une seule réplication, si RF est égal à 2, alors 2 réplications... Lors de l'arrivée d'une requête, elle est envoyée au nœud correspondant puis répliquée selon le facteur de réplication. Le cluster, pendant ce temps-là, attend la réponse des nœuds pour savoir si la lecture ou l'écriture avec les réplications ont été faites avec succès et répondre au client. On distingue, ici, trois types de cohérence en écriture et en lecture (et on a à choisir un):

1. ALL : le cluster attend la réponse de tous les nœuds avant d'écrire ou lire la donnée avant de répondre au client,
2. ONE : le cluster attend seulement la réponse d'un des nœuds avant de répondre au client,
3. QUORUM : le quorum est défini comme la majorité absolue des réplicas. Si la donnée est répliquée 3 fois, alors le cluster attendra la réponse de 2 nœuds avant de répondre au client, c.à.d.  $\text{quorum} = (\text{nombre de réplication} / 2) + 1$ .

NB :

Si on choisit le type 'ALL', dans ce cas, on peut avoir la cohérence absolue. Cassandra, ici, perd la disponibilité puisqu'il faut s'assurer à chaque fois que l'opération a été faite partout, ce qui rend le système beaucoup moins disponible.

Comment les données sont-elles stockées dans un nœud ?

Pour travailler sur Cassandra il faut savoir les termes suivants : colonnes, lignes, famille de colonnes et keyspace.

Une **colonne** est la plus petite unité de modèle dans Cassandra. Elle est constituée d'un nom, d'une valeur et d'un timestamp. Le timestamp détermine quand cette colonne a été créée et par suite savoir la valeur la plus récente. Le nom peut atteindre 64Koets, la valeur 2Goets.

Nom
Valeur
<i>Timestamp</i>

Une **ligne** est un ensemble de colonnes. Elle peut contenir jusqu'à deux milliards de colonnes. Elle possède une clé qui peut atteindre aussi 64 Koctets.

baronm	familyName	firstName	age	address
	BARON	Mickael	36	Poitiers
duponto	familyName	firstName	phone	
	DUPONT	Olivier	+335432312	

On remarque que les colonnes ne sont pas identiques entre les lignes.

La **famille de colonnes** est un ensemble de lignes. Elle correspond à peu près à une table dans une base de données ordinaire.

Persons				
baronm	familyName	firstName	age	address
	BARON	Mickael	36	Poitiers
duponto	familyName	firstName	phone	
	DUPONT	Olivier	+335432312	

Le **Keyspace** est un ensemble de famille de colonnes.

CassandraDEMO				
Persons				
baronm	familyName	firstName	age	address
	BARON	Mickael	36	Poitiers
duponto	familyName	firstName	phone	
	DUPONT	Olivier	+335432312	
...				

Cassandra supporte les types suivants :

- 1) bytes
- 2) boolean
- 3) composite
- 4) counter
- 5) date (bientôt déprécié)
- 6) decimal
- 7) double
- 8) float
- 9) inetAddress
- 10) int32 (entier de taille fixe)
- 11) integer (entier de taille variable)
- 12) long
- 13) uuid
- 14) timestamp
- 15) timeuuid

Le langage de Cassandra s'appelle CQL.

## Installer Cassandra sur Ubuntu

Pour installer Cassandra, Il faut d'abord qu'un JRE (Java Runtime Environment) soit installé sur Ubuntu. Pour installer JRE, il faut rendre le JRE package disponible. Pour cela, il faut ouvrir le Terminal et exécuter la commande suivante :

```
$ sudo add-apt-repository ppa:webupd8team/java
```

Puis, une mise à jour par la commande suivante:

```
$ sudo apt-get update
```

Ensuite, on installe JRE avec cette commande:

```
$ sudo apt-get install oracle-java8-set-default
```

Après installation, une vérification pour savoir que tout est correcte:

```
$ java -version
```

Vous devez recevoir un message pareil :

```
java version "1.8.0_60"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_60-b27)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)
```

Maintenant, on installe Cassandra:

D'abord, on doit rendre disponible le package Cassandra et préciser la source à l'aide de ces 2 commandes :

```
$ echo "deb http://www.apache.org/dist/cassandra/debian 39x main" | sudo tee -a  
/etc/apt/sources.list.d/cassandra.sources.list
```

Puis,

```
$ echo "deb-src http://www.apache.org/dist/cassandra/debian 39x main" | sudo  
tee -a /etc/apt/sources.list.d/cassandra.sources.list
```

Il faut, ensuite, ajouter 3 clés publiques :

1. La première, à l'aide de ces 2 commandes :

```
$ gpg --keyserver pgp.mit.edu --recv-keys F758CE318D77295D
```

Puis,

```
$ gpg --export --armor F758CE318D77295D | sudo apt-key add -
```

2. La seconde, à l'aide de ces 2 commandes :

```
$ gpg --keyserver pgp.mit.edu --recv-keys 2B5C1B00
```

Puis,

```
$ gpg --export --armor 2B5C1B00 | sudo apt-key add -
```

3. La troisième, à l'aide de ces 2 commandes:

```
$ gpg --keyserver pgp.mit.edu --recv-keys 0353B12C
```

Puis,

```
$ gpg --export --armor 0353B12C | sudo apt-key add -
```

Maintenant, on met à jour package avant installation:

```
$ sudo apt-get update
```

Et, on installe Cassandra

```
$ sudo apt-get install Cassandra
```

Pour savoir le statut de Cassandra, si tout va bien et Cassandra est disponible on utilise cette commande:

```
$ sudo service cassandra status
```

Si Cassandra est disponible on verra le message suivant :

```
● cassandra.service - LSB: distributed storage system for structured data
   Loaded: loaded (/etc/init.d/cassandra)
   Active: active (running) since Mon 2017-03-06 10:05:25 PST; 4min 54s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 1111 ExecStart=/etc/init.d/cassandra start (code=exited, status=0/SUCCESS)
    CGroup: /system.slice/cassandra.service
            └─1728 java -Xloggc:/var/log/cassandra/gc.log -ea -XX:+UseThreadPr...

Mar 06 10:05:25 ubuntu systemd[1]: Starting LSB: distributed storage system.....
Mar 06 10:05:25 ubuntu systemd[1]: Started LSB: distributed storage system ...a.
```

Pour démarrer Cassandra on utilise cette commande :

```
$ sudo service Cassandra start
```

Pour l'arrêter :

```
$ sudo service Cassandra stop
```

Pour manipuler les données et utiliser le langage CQL on exécute d'abord la commande suivante pour pouvoir manipuler le cluster:

***\$ cqlsh***

On peut maintenant utiliser le langage CQL

On va créer un keyspace 'bibliotheque' avec une famille de colonne 'livre' et la remplir.

### Keyspace

Pour savoir les Keyspaces existants on utilise la commande :

***SELECT \* FROM system\_schema.keyspaces;***

Pour créer la Keyspace 'bibliotheque':

***CREATE KEYSPACE bibliotheque WITH REPLICATION = { 'class' :  
'SimpleStrategy', 'replication\_factor' : 2};***

Si je veux modifier dans ce Keyspace :

***ALTER KEYSPACE "bibliotheque" WITH REPLICATION = { 'class' :  
'SimpleStrategy', 'replication\_factor' : 3 };***

Si je veux supprimer un keyspace :

***DROP KEYSPACE 'bibliotheque';***

### Famille de colonne

Pour créer une famille de colonne dans 'bibliotheque':

***create table bibliotheque.livre(code int, nom varchar, auteur varchar, primary  
key(code));***

Pour savoir tous les détails sur cette famille de colonne :

***Desc bibliotheque ;***

Pour modifier une famille de colonne :

***ALTER TABLE bibliotheque.livre ALTER nom TYPE text; (changer le type)  
ALTER TABLE bibliotheque.livre ADD datePublication timestamp;(ajouter)***

Pour supprimer une famille de colonne :

***DROP COLUMNFAMILY bibliotheque.livre;***

Pour supprimer une colonne dans une famille de colonne :

***ALTER TABLE bibliotheque.livre DROP datePublication;***



Ligne :

Pour ajouter une ligne :

```
INSERT INTO bibliotheque.livre (code, datePublication, nom, auteur) VALUES  
(123," 'Mickael', 'Fontaine');
```

Pour modifier une ligne :

```
UPDATE bibliotheque.livre SET datePublication = '1993-03-12' WHERE code =123;
```

Pour supprimer une ligne :

```
DELETE FROM bibliotheque.livre WHERE code=123;
```

### Suppervision

L'utilitaire nodetool est un outil en ligne de commande qui permet de superviser et d'administrer Cassandra.

L'option status donne des informations sur les nœuds d'un cluster (état, la capacité des données...) :

```
$ sudo nodetool status
```

```
$ sudo nodetool -h localhost -p 7199 info
```

## **References**

<https://www.infoq.com/fr/articles/modele-stockage-physique-cassandra>

[http://blog.xebia.fr/wp-content/uploads/2015/07/Data\\_Xebia\\_Cassandra\\_Programmez.pdf](http://blog.xebia.fr/wp-content/uploads/2015/07/Data_Xebia_Cassandra_Programmez.pdf)

<http://soat.developpez.com/articles/cassandra/#LI>

<http://mbaron.developpez.com/tutoriels/nosql/cassandra/installation-outils-administration/>