Yohann Meyer, Yann Lederrey et Joel Schär

DB Caching avec Redis

Pour illustrer l'utilisation de redis pour faire du db caching, nous allons mettre un place une structre avec MongoDB et Express.js.

Le schéma réaliser va permettre de comprendre le fonctionnement et la mise en place des ressources nécessaires. Pour cela nous allons utiliser une structure fictive de librairie virtuelle.

Installation

Il est possible de faire l'installation en compilant l'application directement :

```
wget http://download.redis.io/redis-stable.tar.gz
tar xvzf redis-stable.tar.gz
cd redis-stable
make
```

Il est également possible selon la distribution de faire l'installation depuis un repo :

```
apt install redis-server
```

Une fois installé le serveur peut être démarré avec la commande :

```
redis-server
```

Configuration

Par défaut redis stock dans la RAM tant qu'il y trouve de la place. Il va donc surcharger la RAM au bout d'un moment. Pour éviter cela il est important de donner une règle

Memory

Definir l'espace maximum que le cache est autorisé à utiliser.

`--maxmemory 10mb`

Caching Policy

`--maxmemory-policy <policy>`

- `noeviction` : rien n'est retiré
- `allkeys-lru` : LRU (Least Recently Used), retire les clés utilisées les moins récemment.
- `volatile-lru` : éviction selon LRU pour les clé avec **expire set**
- `allkeys-random` : aléatoirement

- `volatile-random` : aléatoirement, mais seulement si flag volatile est set.
- `volatile-ttl` : clé avec **expire set** est un petit ttl seront évincée.

`volatile` : si aucune clé n'est définir (**expire set**) : on procède comme `noeviction`

Autre praramètres

`--port <port>`

`--slaveof <ip> <port>`

Utilisation

Démarrer : `redis-server <options>`

```
redis-server --maxmemory 10mb --maxmemory-policy allkeys-lru
```

Stopper : `/etc/init.d/redis-server stop`

Commandes CLI

ouvrir la ligne de commande CLI : `redis-cli`

- `keys <patern>` : permet de voir toutes les clés enregistrée dans le cache selon le paterne de la clé.
  - `keys *` : permet de voir toutes les clés
- `set <key> <value>` : permet d'enregistrer une valeur
- `get <key>` : retourne la valeur enregistrée.
- `monitor` : permet de voir toutes les actions du serveur
- `FLUSHALL` : vider le caches redis
- `DEL <key>` : delete value

Intégration

L'intégration de redis dans un application du fait qu'il va simplement permettre de consulter ou de stocker des clés-valeurs.

https://redis.io/clients

Garder le cache valide

Il est important de garder la cache valide si des données sont modifiée dans la base de donnée. Autrement les informations resterons erronées tant que la valeur est présente dans le cache.

source : https://www.sitepoint.com/caching-a-mongodb-database-with-redis/