TP 03
Bases des données orientées clé-valeur

Ce TP est noté. Vous devez soumettre un compte rendu (questions, requêtes et résultats) sur votre dépôt GitLab dans un dossier intitulé TP03 au plus tard à minuit, avant le 02 octobre.

Exercice 1 :

Redis (l'acronyme de REmote DIctionary Server qui peut être traduit par « serveur de dictionnaire distant ») est un magasin de structure de données clef/valeur en mémoire open source rapide. Redis propose un ensemble de structures de données en mémoire polyvalentes qui permet de créer facilement un large éventail d'applications personnalisées. Les principaux cas d'utilisation de Redis comprennent la mise en cache, la gestion des sessions. Il est écrit en code C optimisé et prend en charge plusieurs langages de développement.

Sa vitesse et sa facilité d'utilisation en font une solution de choix pour toutes les applications (web, mobiles, jeux, technologies publicitaires et Internet des objets) qui nécessitent les meilleures performances. AWS assure la prise en charge de Redis via un service de base de données entièrement géré et optimisé nommé Amazon ElastiCache for Redis, et permet également aux clients d'exécuter un système Redis autogéré sur AWS EC2.

Trois caractéristiques principales distinguent Redis :

- ➔ Redis maintient l'ensemble de sa base de données en mémoire, utilisant le disque uniquement pour la persistance des données.
- ➔ Redis propose un éventail de types de données relativement varié par rapport à d'autres systèmes de stockage clé-valeur.
- ➔ Redis emploie une architecture maître-esclave et prend en charge la réplication asynchrone qui permet de répliquer les données sur plusieurs serveurs esclaves

Installation de Redis sur Ubuntu

- Pour installer Redis sur Ubuntu, ouvrez le terminal et saisissez les commandes suivantes :pliquer des données vers un nombre illimité d'instances esclaves :

  $sudo apt-get update

  $sudo apt-get install redis-server

- Pour démarrer Redis sur Ubuntu, vous pouvez utiliser la commande suivante dans le terminal :

  redis-server

- Vérifier que Redis fonctionne :

  redis-cli ping

  Vous devriez obtenir une réponse PONG si Redis fonctionne correctement.

- Ouvrir une invite redis

  redis-cli

  Dans l'invite, 127.0.0.1 est l'adresse IP de votre machine et 6379 est le port sur lequel le serveur Redis s'exécute. Re-tapez maintenant la commande PING suivante.

Configuration

Dans Redis, il existe un fichier de configuration (redis.conf) disponible dans le répertoire racine de Redis. Cependant, vous pouvez obtenir toutes les configurations Redis à l'aide de la commande Redis CONFIG .

redis 127.0.0.1:6379> CONFIG GET CONFIG_SETTING_NAME.

exemple : La commande CONFIG GET loglevel dans Redis est utilisée pour obtenir le niveau de journalisation (log level) actuel configuré pour le serveur Redis. Pour obtenir tous les paramètres de configuration, utilisez * à la place de CONFIG_SETTING_NAME.

Pour mettre à jour la configuration, vous pouvez modifier directement le fichier **redis.conf** ou mettre à jour les configurations via la commande **CONFIG set**

CONFIG SET CONFIG_SETTING_NAME NEW_CONFIG_VALUE

1) Donner une une interprétation de la commande : CONFIG SET loglevel "notice"

Type des données

Redis est un magasin de données en mémoire qui utilise une structure **clé-valeur**. Cela signifie que chaque donnée est stockée sous forme de **clé** (unique) et de **valeur** (donnée associée). Redis prend en charge plusieurs types de données, chacun ayant ses propres caractéristiques et utilisations.

https://redis.io/docs/latest/develop/data-types/

- Strings : Les chaînes dans Redis sont binaires sécurisées, ce qui signifie qu'elles ont une longueur connue qui n'est pas déterminée par des caractères de terminaison spéciaux. Ainsi, vous pouvez stocker jusqu'à 512 mégaoctets dans une chaîne.
- Hachages : Un hachage Redis est une collection de paires clé-valeur. Les hachages Redis sont des mappages entre les champs de chaîne et les valeurs de chaîne. Par conséquent, ils sont utilisés pour représenter des objets.
- Sets : Une collection non ordonnée d'éléments uniques. Idéal pour stocker des valeurs distinctes.
- SortedSets : Similaire aux ensembles, mais chaque élément a un score qui détermine son ordre
- Lists : Une collection ordonnée d'éléments. Les éléments peuvent être ajoutés ou supprimés aux deux extrémités

Commandes Utiles :

- Pour définir une clé : SET key value
- Pour obtenir la valeur d'une clé : GET key

- Pour supprimer une clé : DEL key
- Pour vérifier si une clé existe : EXISTS key

Tous les commands Redis : https://redis.io/docs/latest/commands/

1) Comment créer une clé (key) username avec la valeur (value) Alice dans Redis ?
2) Quelle commande utiliseriez-vous pour récupérer la valeur de la clé username ?
3) Comment modifier la valeur de la clé username pour Bob ?
4) Comment vérifier la nouvelle valeur de la clé username ?
5) Comment créer une liste (list) fruits et ajouter les éléments apple, banana, orange ?
6) Quelle commande permet d'afficher tous les éléments de la liste fruits ?
7) Comment ajouter grape à la fin de la liste fruits ?
8) Comment afficher à nouveau tous les éléments de la liste fruits ?
9) Comment créer un ensemble (set) colors et ajouter les éléments red, blue, green ?
10) Que se passe-t-il si vous essayez d'ajouter red à l'ensemble colors ?
11) Quelle commande permet d'afficher tous les membres de l'ensemble colors ?
12) Comment créer un ensemble trié (sorted set) scores avec les éléments Alice (100) et Bob (200) ?
13) Quelle commande utiliseriez-vous pour afficher les membres de l'ensemble trié scores avec leurs scores ?
14) Comment ajouter Charlie avec un score de 150 à l'ensemble trié scores ?
15) Comment afficher les membres de l'ensemble trié scores à nouveau ?
16) Comment créer un hachage (hash) user:1001 avec les champs name (Alice) et age (30) ?
17) Quelle commande utiliseriez-vous pour récupérer le nom de l'utilisateur dans le hachage user:1001 ?
18) Comment ajouter un champ email à l'utilisateur dans le hachage user:1001 ?
19) Quelle commande permet d'afficher tous les champs du hachage user:1001 ?

Bases de données du SGBD Redis :

Par défaut, lorsque vous vous connectez à Redis, vous êtes automatiquement connecté à la base de données db0, qui est la première base de données (numéro

0). Cela signifie que toutes les opérations que vous effectuez sans spécifier de base de données se feront dans db0.

Pour faciliter la gestion de Redis, vous pouvez utiliser une application web nommée **RedisInsight**.

Le nombre de bases de données est fixe et défini dans le fichier de configuration, avec 16 bases de données par défaut. Chaque base est identifiée par un numéro correspondant à son nom. Pour connaître le nombre de bases de données, utilisez la commande **:**

CONFIG GET databases

La commande INFO keyspace affiche des informations sur les espaces de clés (keyspaces) dans Redis.

20) Tester et interpréter le résultat de la commande INFO keyspace
21) Exécuter et interpréter les commandes suivantes :

a)

```
SET visites 10
GET visites
INCR visites
GET visites
DECR visites
GET visites
```

b)

```
INCRBY visites 10
GET visites
DECRBY visites 5
GET visites
```

c)

```
SET hello "Hello World"
SETRANGE hello 6 "Redis"
GET hello
GETRANGE hello 0 5
```

d)

```
SET hello "Hello World"
STRLEN hello
```

e)

```
SET hello "Hello"
APPEND hello " World !"
GET hello
```

22) Créer **les clés suivantes**

- user:1 avec pour valeur "Damien Faure"
- user:1:city avec pour valeur "La Rochelle"
- user:1:age avec pour valeur "35"
- user:1:activity avec pour valeur "Running" et une expiration dans 10 minutes
- user:1:hobby avec pour valeur "Course à pieds"
- user:1:weight avec pour valeur "74"

23) Écrire et exécuter les commandes pour lire séparément les clés :

user:1, user:1:city, user:1:age

24) Écrire et exécuter les commandes pour lire en une fois les clés de user:1:activity, user:1:hobby et user:1:weight

25) Définir maintenant la clé user:1 avec pour valeur "Thomas Davaut" à condition que cette clé n'existe pas.

26) Définir la clé user:1:city à “Bordeaux" à condition que cette clé existe.

27) Notre user a pris un an, et à perdu trois kilos grâce à la course à pied. À l'aide des commandes Redis adéquates, modifiez ces valeurs. De plus, il fait le trajet régulièrement entre La Rochelle et Bordeaux. Mettez à jour la valeur de la clé city à "La Rochelle/Bordeaux" sans l'écraser complètement.
28) Notre user s'est mis à la course hippique. Sans écraser toute la valeur de la clé hobby, mettez à jour ce dernier. Récupérez la longueur de la clé hobby.
29) Pour finir, récupérez juste le fragment de valeur "Bordeaux" de la clé city.
30) Antoine et Thomas organisent une soirée. Antoine veut inviter : Sophie, Etienne, Hélène, Paul, Charles, Ethan et Pauline Thomas lui souhaite inviter : Etienne, Paul, Catherine, Ethan, Pauline, Sophia et Mickaël Ayant un espace limité, ils souhaitent inviter uniquement leurs amis communs. Qui sont-ils ? La semaine suivante, ils souhaitent organiser une soirée avec les personnes qui n'ont pas été invité à la première. À l'aide de la documentation, établissez une liste des invités d'Antoine et une liste des invités de Thomas dans de nouveaux SET. Quelle-est la liste finale des invités pour cette seconde soirée ? Une fois établie, exportez-là dans un nouveau set avec l'aide de la documentation.

Exercice 2 :

1. Créer la clé User avec comme valeur Dario
2. Créer, avec une seule commande, les 2 clés suivantes
   a. User:3 avec comme valeur Maud
   b. User:4 avec comme valeur Xavier
3. Lister toutes les clés définies dans la base
4. Lister toutes les clés commençant par User User:3
5. Lister les valeurs des différentes clés
6. Ajouter un ‘e’ à la fin de valeur de la clé
7. Tester et interpréter la commande : SETNX User:3 "Aude"
8. Créer une clé temporaire User:5 avec comme valeur Toto de durée de vie 30 secondes
9. Vérifier la durée de vie de la clé User:5
10. Vérifier la durée de vie de la clé User:1
11. Créer les clés suivantes :

a. `User:6` avec pour valeur `Toto`
b. `User:6:City` pour valeur `Paris`
c. `User:6:Age` avec pour valeur 25
d. `User:6:Activity` avec pour valeur `Tutorial` et une expiration dans 10 minutes (avec la commande `SET`)
e. Vérifier plusieurs fois la durée de vie de la clé `User:6:Activity`

12. Ajouter, à la valeur de User:6:Activity, la valeur " Redis" (commande APPEND) et vérifier la valeur de User:6:Activity
13. Tester les 2 instructions suivantes et analyser leur résultat :
    MSETNX User:6:Note 20 User:4:Time 5
    MSETNX User:6:Note 20 User:4:Validation Redis
    MSETNX User:5:Note 20 User:4:Validation Redis
14. Créer un utilisateur avec des champs similaires en utilisant les commandes HSET et HMSET.
15. Tester les commandes HKEYS et HVALS sur ce nouvel utilisateur.
16. Tester la commande suivante pour insérer une valeur en JSON:
    SET User:8 {"nom":"Titi","age":23,"Activity":["Redis","MongoDB"]}
17. Créer une liste Cours en y ajouter la valeur NoSQL (cf. commande LPUSH) 2
18. Ajouter la valeur MicroServices dans la liste Cours (cf. commande LPUSH)
19. Lister les éléments de la liste
20. Exécuter la commande MULTI
21. Créer une liste Cours2 en y ajouter la valeur NoSQL
22. Ajouter la valeur MicroServices dans la liste Cours

Bon travail