

Rapport Technique : Découverte et Utilisation de Redis

1. Introduction : Positionnement parmi les bases NoSQL

Les bases de données NoSQL représentent une alternative moderne aux bases relationnelles, adaptées aux besoins de performance et de scalabilité. On distingue quatre grandes familles :

1. **Clé-valeur** (Redis, DynamoDB) : stockage simple et rapide pour des paires clé-valeur.
2. **Orientées colonnes** (Cassandra, HBase) : adaptées aux traitements analytiques de grands volumes de données.
3. **Orientées documents** (MongoDB, CouchDB) : stockage semi-structuré flexible.
4. **Orientées graphes** (Neo4j, Amazon Neptune) : pour modéliser des relations complexes.

Redis se classe dans la famille clé-valeur, tout en supportant des structures de données avancées.

2. Présentation de Redis

Redis (Remote Dictionary Server) est une base NoSQL en mémoire vive, connue pour ses performances élevées et ses nombreuses fonctionnalités : - Stockage clé-valeur avec persistance optionnelle. - Structures de données avancées : listes, ensembles, hashes, etc. - Modèle client-serveur : serveur par défaut sur le port 6379, accessible via CLI ou interfaces graphiques.

3. Installation et Configuration

1. **Installation de Redis** : Voici les étapes pour installer Redis sur un système basé sur Debian (comme Ubuntu) :

```
sudo apt update  
sudo apt install redis-server
```

2. **Démarrage du serveur Redis** :

```
redis-server
```

1. **Connexion au client CLI** :

```
redis-cli
```

1. **Configuration** : Configurer des fichiers de sauvegarde ou ajuster les paramètres par défaut dans `redis.conf`.

4. Manipulations et Structures de Données

Redis supporte plusieurs structures de données clés :

- **Clés simples :**

```
SET ma_cle "Bonjour"  
GET ma_cle  
DEL ma_cle
```

- **Listes :**

```
LPUSH ma_liste "Element1"  
RPUSH ma_liste "Element2"  
LRANGE ma_liste 0 -1  
LPOP ma_liste
```

- **Ensembles :**

```
SADD mon_ensemble "A" "B"  
SMEMBERS mon_ensemble  
SREM mon_ensemble "A"
```

- **Ensembles ordonnés :**

```
ZADD classement 10 "Alice"  
ZRANGE classement 0 -1
```

- **Hashes :**

```
HSET utilisateur:1 nom "Alice" age 30  
HGET utilisateur:1 nom  
HGETALL utilisateur:1
```

5. Fonctionnalités Avancées

- **Pub/Sub :** Redis facilite la communication temps réel entre clients via des canaux.

```
SUBSCRIBE canal1  
PUBLISH canal1 "Message"
```

- **Bases multiples :** Redis supporte jusqu'à 16 bases isolées par défaut.

```
SELECT 1  
KEYS *  
SELECT 0
```

6. Gestion de la Persistance

Redis propose plusieurs options de persistance : 1. **Snapshots périodiques (RDB)** : fichiers stockés à intervalles réguliers. 2. **Journalisation (AOF)** : enregistrement des commandes pour une récupération précise.

7. Considérations de Performance

- Accès en RAM : très rapide ($\sim 10^{-8}$ s) mais limité par la capacité mémoire.
- Comparaison RAM/disque : la RAM est environ 1 million de fois plus rapide.

8. Conclusion

Redis est un outil puissant pour des cas nécessitant rapidité et structures flexibles. Cependant, ses limites résident dans la gestion mémoire et la configuration manuelle de la persistance.

Comment Redis gère-t-il les bases multiples ?

Redis utilise une fonctionnalité appelée “databases” pour gérer plusieurs bases de données. Par défaut, Redis démarre avec une seule base de données (base 0), mais il est possible de configurer jusqu’à 16 bases de données (numérotées de 0 à 15). Les bases multiples permettent de compartimenter les données, mais elles ne sont pas complètement isolées car elles partagent la même instance de Redis.

Points importants :

- **Changement de base de données** : Vous pouvez changer de base à l’aide de la commande SELECT.

```
SELECT 1 # Passe à la base de données 1
```

- **Limitations** : Les bases multiples ne sont pas adaptées pour des cas d’utilisation nécessitant une isolation stricte ou des autorisations spécifiques.

Quelles sont les limites de la persistance ?

Redis offre deux principales options pour la persistance des données : **RDB (Redis Database)** et **AOF (Append-Only File)**. Chacune a ses avantages et ses inconvénients.

RDB (Redis Database)

- **Avantages :**
 - Sauvegarde instantanée et compacte des données à des intervalles définis.
 - Consommation CPU minimale pendant les sauvegardes.
- **Inconvénients :**
 - Risque de perte de données si le serveur plante entre deux sauvegardes.
 - Sauvegardes gourmandes en ressources pour les grandes bases de données.

AOF (Append-Only File)

- **Avantages :**
 - Offre une plus grande durabilité grâce à l'enregistrement de chaque opération en temps réel.
 - Moins de risque de perte de données.
- **Inconvénients :**
 - Taille des fichiers AOF peut être importante avec le temps.
 - Ralentissement potentiel des performances dû à l'écriture fréquente sur le disque.

Limites générales :

- La persistance peut augmenter la latence si les opérations de disque sont trop fréquentes.
- Risque de corruption des fichiers RDB ou AOF en cas de panne inattendue.