

## Rapport de Comparaison de CouchDB et Redis

### 1. Introduction

Ce rapport présente une étude comparative de deux bases de données NoSQL populaires : CouchDB et Redis. Nous avons examiné leurs architectures, avantages, limites, et leur relation avec le théorème CAP. Nous avons également effectué des tests pratiques en mesurant les performances des opérations CRUD (Create, Read, Update, Delete).

### 2. Description de CouchDB

**Architecture et Fonctionnement :** CouchDB est une base de données NoSQL qui utilise des documents JSON pour stocker des données. Elle repose sur une architecture orientée document avec un schéma libre, permettant une grande flexibilité dans la structure des données. CouchDB supporte la réplication et la synchronisation entre bases de données, ce qui en fait un choix robuste pour les applications distribuées.

#### Avantages :

- Haute disponibilité et tolérance aux pannes grâce à la réplication.
- Flexibilité dans la structure des données.
- Interface de gestion conviviale via Fauxton.

id	key	value
author1	author1	{ "rev": "1-6e93e8aeb4144ba6c51885..." }
author2	author2	{ "rev": "1-0f66d8d89f54c700f9662da2..." }
book1	book1	{ "rev": "10-0a3f163aced4f2350a42be..." }
book2	book2	{ "rev": "1-87c78f96a73dd9da2e75108..." }
book3	book3	{ "rev": "1-3a018c980b68de66534b15..." }
borrow1	borrow1	{ "rev": "1-bc5ffa7469fe31969bbeec1b..." }
borrow2	borrow2	{ "rev": "1-c7260090b7e2131e0c3a227..." }

#### Limites :

- Performance inférieure pour des applications nécessitant des transactions complexes.

**Relation avec le Théorème CAP :** CouchDB favorise la Consistance et la Tolérance au partitionnement, avec un compromis sur la Disponibilité en cas de partitions réseau.

### 3. Description de Redis

**Architecture et Fonctionnement :** Redis est une base de données NoSQL en mémoire clé-valeur qui supporte divers types de données tels que les chaînes, listes, ensembles, et hachages. Il est conçu pour des performances élevées avec une faible latence grâce à son fonctionnement entièrement en mémoire.

#### Avantages :

- Performance élevée et faible latence.
- Simplicité d'utilisation et de configuration.
- Supporte différents types de données.

#### Limites :

- Limité par la taille de la mémoire disponible.
- Persistance des données secondaire, nécessitant des sauvegardes régulières pour éviter la perte de données.

**Relation avec le Théorème CAP :** Redis favorise la Disponibilité et la Tolérance au partitionnement, avec un compromis sur la Consistance en mode distribué.

### 4. Étude Théorique

#### CouchDB :

- Modèle de données flexible.
- Replication Set et Sharding.
- Idéal pour les applications nécessitant une réplication et une synchronisation faciles.

#### Redis :

- Modèle de données simple et rapide.
- Supporte différents types de données.
- Idéal pour les applications nécessitant des performances élevées et une faible latence.

### 5. Tests Pratiques et Résultats des Performances

Nous allons maintenant développer des scripts pour charger un grand volume de données dans CouchDB et Redis, puis mesurer les performances des opérations CRUD.

**Données simulées :** Après les tests, nous avons trouvé les résultats suivants

#### CouchDB :

- Time taken: 313.5557017326355 seconds Pour l'insertion de 1000 documents
- Time taken: 0.02392101287841797 seconds
- Time taken: 0.018618345260620117 seconds
- Time taken: 0.01402139663696289 seconds
- Time taken: 0.039955854415893555 seconds
- Time taken: 0.02508378028869629 seconds

Time taken: 0.07315921783447266 seconds

### Redis :

- Time taken: 6.314136981964111 seconds
- Time taken: 0.0019996166229248047 seconds
- Time taken: 0.01465749740600586 seconds
- Time taken: 0.0010218620300292969 seconds
- Time taken: 0.0 seconds

## 6. Conclusion

Les résultats des tests pratiques montrent que Redis est significativement plus rapide que CouchDB pour toutes les opérations CRUD. Redis, grâce à son fonctionnement en mémoire, offre une latence extrêmement faible, ce qui le rend idéal pour des applications nécessitant des performances élevées. Cependant, CouchDB offre des avantages en termes de flexibilité des données et de réplication, ce qui peut être crucial pour des applications distribuées et tolérantes aux pannes.

En résumé, le choix entre CouchDB et Redis dépend des exigences spécifiques de l'application :

- **Redis** est recommandé pour des applications nécessitant des performances élevées et une faible latence.
- **CouchDB** est recommandé pour des applications nécessitant une réplication facile et une tolérance aux pannes avec des données flexibles.

## Annexes

Les scripts Python utilisés pour importer les données et mesurer les performances sont inclus en annexe pour référence.