

Rapport de Comparaison des Performances entre MySQL et MongoDB

Auteur: *Votre Nom*

Décembre 2024

Contents

Rapport de Comparaison des Performances entre MySQL et MongoDB
Auteur: *Brelot Julien*

Contents

Chapter 1

Introduction

Ce rapport présente une étude comparative des performances entre MySQL et MongoDB, deux systèmes de gestion de bases de données (SGBD) très utilisés, en mettant l'accent sur les opérations CRUD (Create, Read, Update, Delete).

L'objectif est de tester différentes configurations et scénarios d'utilisation pour analyser l'impact des index, de la réplication, et de la fragmentation sur les performances.

La structure des tests, ainsi que les résultats obtenus, sont décrits en détail.

Chapter 2

Contexte et Objectifs

2.1 Systèmes Comparés

2.1.1 MongoDB

MongoDB est un SGBD NoSQL orienté documents. Contrairement aux bases relationnelles, MongoDB stocke les données sous forme de documents JSON ou BSON, offrant une grande flexibilité structurelle. Ses caractéristiques principales incluent :

- **Stockage orienté documents** : Les données sont regroupées en collections, chaque document étant un objet JSON avec une structure flexible.
- **Requêtes riches** : MongoDB prend en charge des requêtes complexes incluant des opérations d'agrégation et des jointures.
- **Scalabilité horizontale** : Avec la fragmentation (sharding), les données peuvent être réparties sur plusieurs nœuds.
- **Réplication** : MongoDB utilise des Replica Sets pour garantir la haute disponibilité et la tolérance aux pannes.
- **Indexation avancée** : MongoDB offre un support pour les index secondaires et composites afin d'optimiser les performances des requêtes.

2.2 Étude des aspects de MongoDB

2.2.1 Comparaison des Performances des Requêtes

MongoDB se distingue des bases relationnelles comme MySQL par sa capacité à exécuter des requêtes sur des données semi-structurées et non structurées. La comparaison des performances des requêtes révèle :

- MongoDB excelle dans les scénarios où les données sont semi-structurées ou nécessitent une mise à jour fréquente de leur structure.
- MySQL, en revanche, est plus performant pour les requêtes transactionnelles complexes grâce à son moteur ACID robuste.

Des tests sur des collections contenant des millions de documents montrent que MongoDB est plus rapide pour des lectures non indexées, tandis que MySQL excelle lorsque des index adaptés sont en place.

2.2.2 Performances avec et sans Indexation

L'indexation est un facteur clé des performances :

- **MongoDB sans indexation** : Les requêtes nécessitent un balayage complet des documents, ce qui ralentit considérablement les performances.
- **MongoDB avec indexation** : Les performances des lectures augmentent de façon exponentielle grâce à la réduction du nombre de documents scannés.
- **MySQL avec indexation** : Étant donné sa nature relationnelle, MySQL exploite efficacement les index pour optimiser les lectures et écritures.

2.2.3 Impact de la Réplication et de la Fragmentation

La réplication et la fragmentation influencent directement les performances et la tolérance aux pannes.

Réplication dans MongoDB :

- **Réplication primaire-secondaire** : Permet une haute disponibilité des données. Les écritures se font uniquement sur le nœud primaire, tandis que les lectures peuvent être réparties sur les nœuds secondaires.
- **Élections automatiques** : En cas de panne du nœud primaire, un nouveau primaire est élu parmi les secondaires, garantissant une continuité de service.
- **Impact sur les performances** : La latence d'écriture augmente avec un `write concern` élevé, car l'écriture doit être confirmée par plusieurs nœuds.

Fragmentation dans MongoDB :

- **Distribution des données** : Les documents sont répartis entre plusieurs shards, permettant de gérer efficacement de grandes volumétries.
- **Impact sur les performances** : La fragmentation améliore les temps de réponse pour les lectures et écritures parallèles, mais peut augmenter la complexité des requêtes agrégées.

Comparaison avec MySQL :

- **Réplication** : MySQL utilise un modèle maître-esclave classique, où les lectures peuvent être réparties, mais les écritures sont limitées au maître.
- **Fragmentation** : Disponible via MySQL Cluster, mais nécessite une configuration plus complexe comparée à MongoDB.

Chapter 3

Configurations des Tests

3.1 Architecture MongoDB

3.1.1 Standalone

Un seul serveur MongoDB.

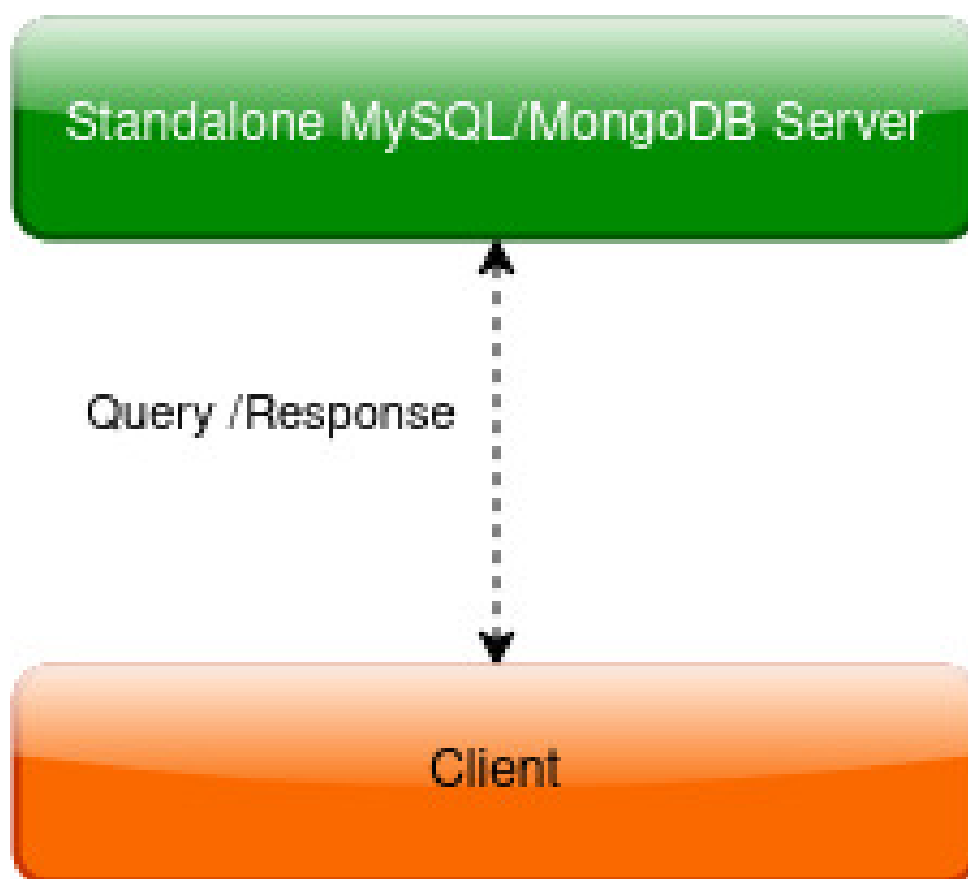


Figure 3.1: Architecture Standalone.

3.1.2 Replica Set

Un Replica Set avec un nœud primaire et deux nœuds secondaires.

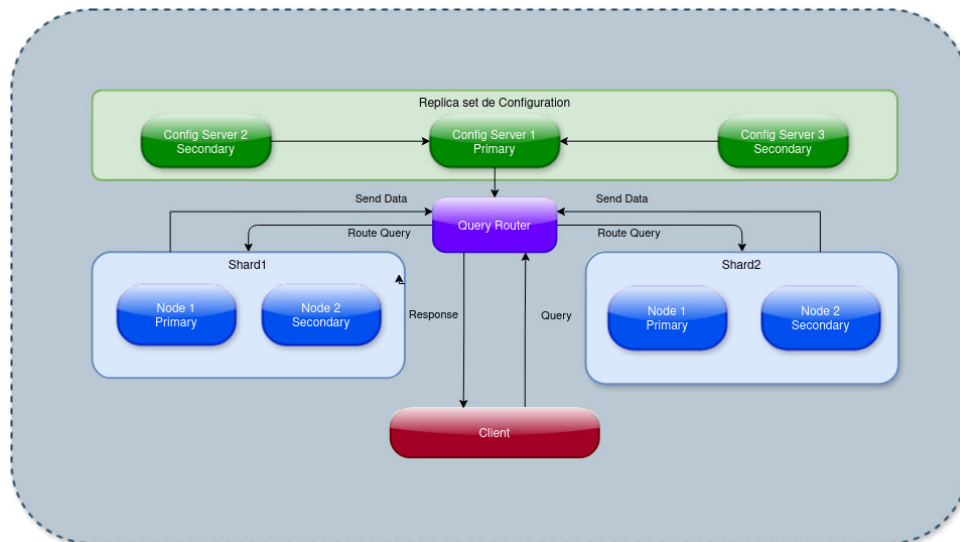


Figure 3.3: Architecture Sharded Cluster de MongoDB.

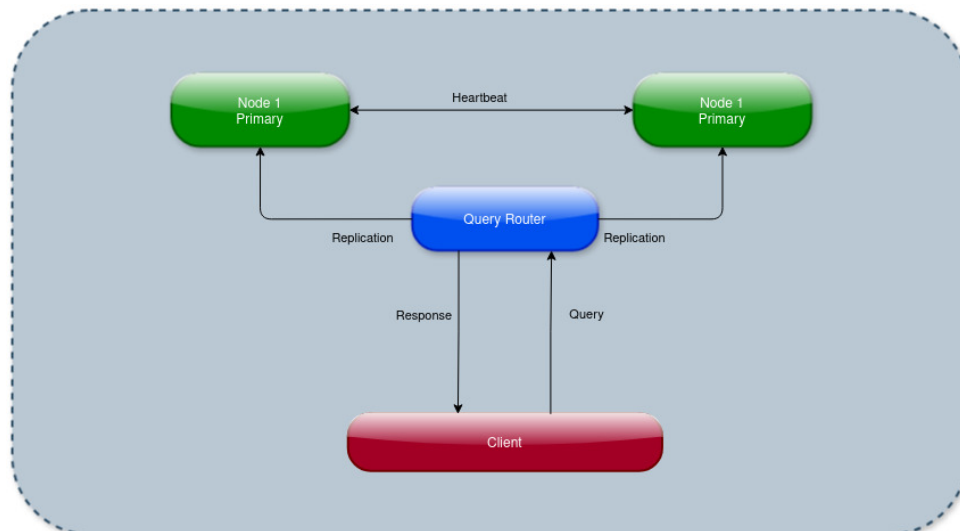


Figure 3.2: Architecture Replica Set.

3.1.3 Sharded Cluster

Un cluster avec trois Serveurs de configuration, un Query router, et deux shards constitués chacun de deux nœuds.

3.2 Architecture MySQL

3.2.1 Standalone

Un serveur MySQL simple. Voir la figure ??.

3.2.2 Sharded Architecture

Architecture MySQL-NDB avec :

- Un gestionnaire (manager).
- Deux nœuds de données.
- Un noeud SQL.

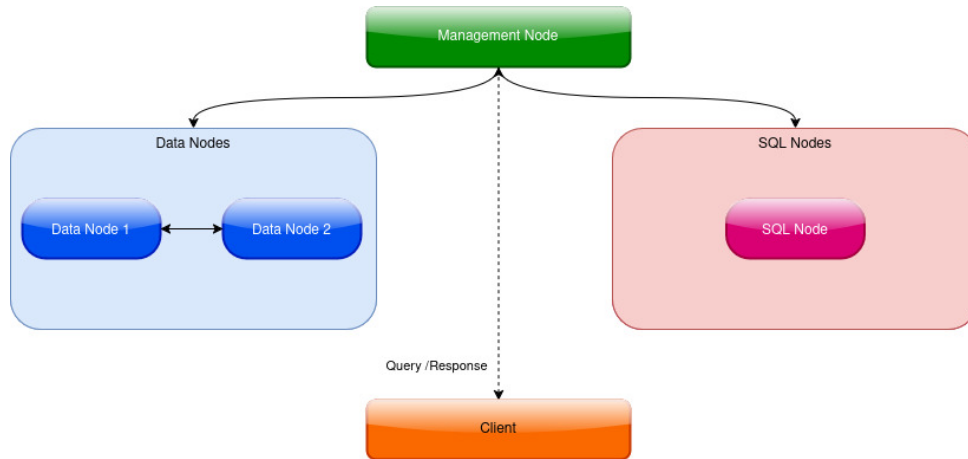


Figure 3.4: Architecture MySQL Cluster.

Chapter 4

Structure des Données

La table/collection "test" utilisée pour les tests a la structure suivante :

- **id** : Identifiant unique (UUID ou auto-incrément pour MySQL, ObjectId pour MongoDB).
- **title** : Titre du livre (String).
- **author** : Auteur du livre (String).
- **published_date** : Date de publication (Date).
- **genre** : Genre du livre (String).
- **price** : Prix du livre (Float).
- **copies_sold** : Nombre d'exemplaires vendus (Integer).
- **ran** : Champ aléatoire pour les tests entre 0 et `num_records_per_many - 1` (Integer).

Ce qui donne pour MySQL :

```
CREATE TABLE test (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  title VARCHAR(255),  
  author VARCHAR(255),  
  published_date DATE,  
  genre VARCHAR(255),  
  price FLOAT,  
  copies_sold INT,  
  ran INT  
);
```

Chapter 5

Batterie de Tests

5.1 Types de Tests

- Tests CRUD : insertion, lecture, mise à jour, suppression.
- Tests avec indexation et sans indexation.

5.2 Scénarios

- Opérations globales sur `num_records` données.
- Insertion ou mise à jour de `num_records_per_many` données avec variation du nombre de données initiales.

Chapter 6

Résultats et Analyses

6.1 Graphiques des Performances

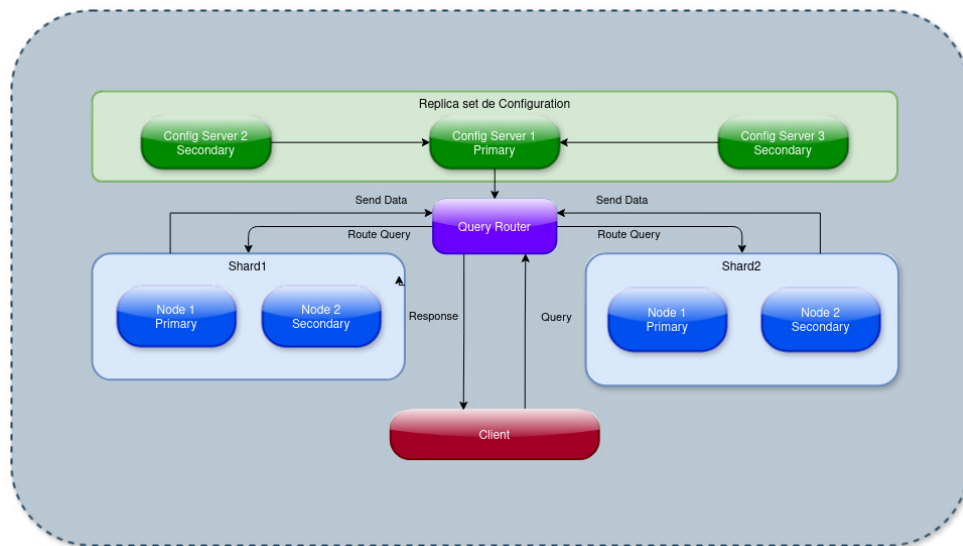


Figure 6.1: Comparaison des performances des opérations CRUD.

6.2 Impact de l'Indexation

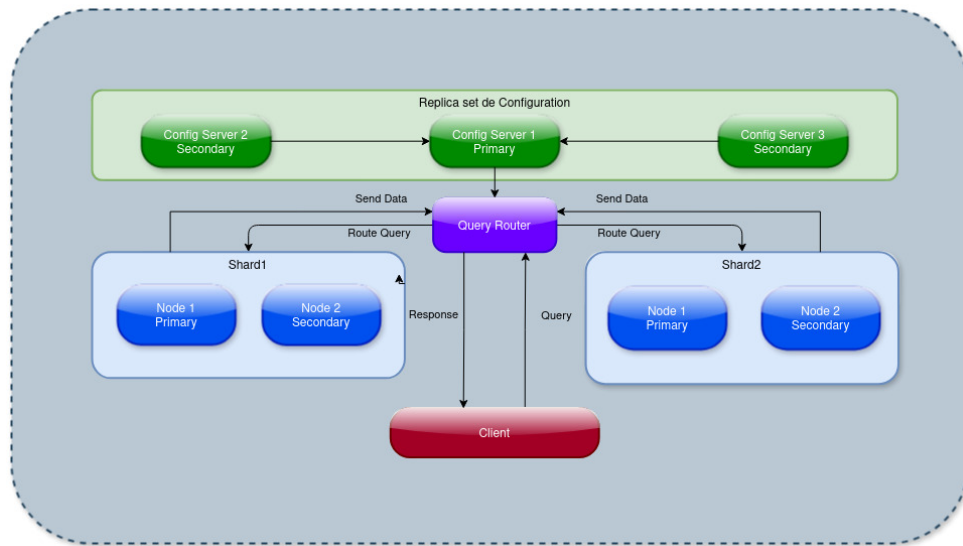


Figure 6.2: Impact de l'indexation sur les performances.

6.3 Réplication et Sharding

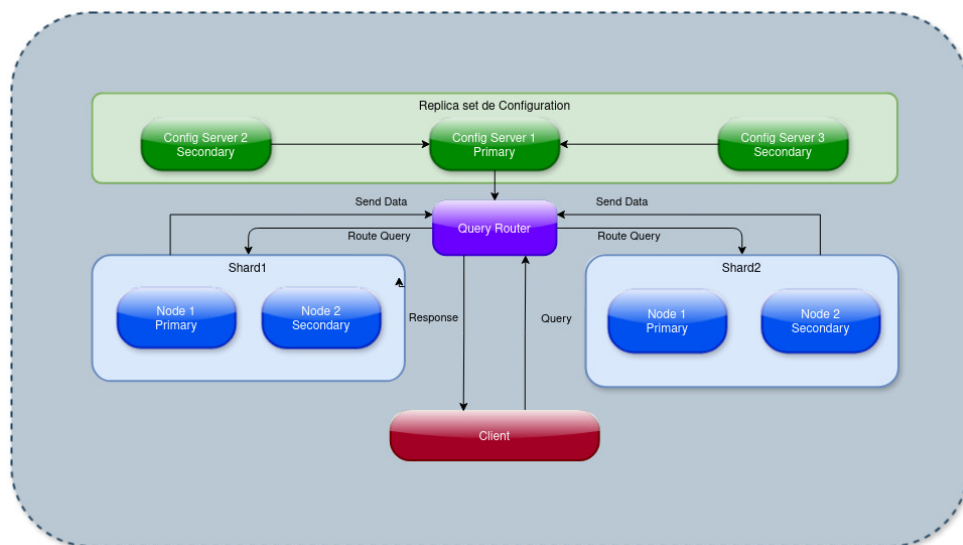


Figure 6.3: Impact de la réplication et du sharding sur les performances.

Chapter 7

Analyse des Résultats

7.1 Performances des Requêtes

- MongoDB excelle pour les lectures non indexées et les mises à jour fréquentes de la structure des données.
- MySQL est plus performant pour les requêtes transactionnelles complexes et les lectures indexées.
- L'indexation est un facteur clé pour améliorer les performances des requêtes.
- Les performances de MongoDB augmentent grandement avec l'indexation.
- MySQL est plus rapide pour les requêtes indexées.
- Les performances de MongoDB sont meilleures pour les lectures non indexées.
- L'indexation est un facteur clé pour améliorer les performances des requêtes.

7.2 Impact de la Réplication et du Sharding

- La réplication et le sharding améliorent la tolérance aux pannes et la scalabilité.
- La réplication augmente la latence d'écriture, mais garantit une haute disponibilité des données.
- Le sharding améliore les performances des lectures et écritures parallèles.
- MySQL Cluster offre des fonctionnalités similaires à MongoDB, mais avec une configuration plus complexe.
- MongoDB est plus adapté pour les applications nécessitant une scalabilité horizontale et une haute disponibilité.

7.3 Critique des résultats

- Les tests ont été réalisés sur des configurations de base, sans optimisations spécifiques.

- Les performances peuvent varier en fonction de la charge, de la volumétrie des données, et de la configuration matérielle.
- Les résultats sont indicatifs et doivent être confirmés par des tests plus approfondis.
- Les performances des SGBD dépendent fortement de la nature des requêtes et de la structure des données.
- Les tests ont été réalisés sur des scénarios simples, mais les performances peuvent varier dans des cas d'utilisation réels.

Chapter 8

Conclusion

Cette étude met en évidence les avantages et inconvénients des architectures NoSQL et relationnelles dans divers scénarios. MongoDB offre une grande flexibilité et scalabilité, tandis que MySQL reste performant pour des cas nécessitant des transactions complexes.