



Rapport SAE NoSQL

BUT SD 3 VCOD

INTRODUCTION

OBJECTIF DU RAPPORT

Ce rapport a pour objectif de présenter la migration d'une base de données relationnelle vers une base de données NoSQL, en s'appuyant sur un projet de migration confié par Paula DUPONT, directrice d'une entreprise de voitures. La migration vise à résoudre des problèmes liés à la latence des requêtes et à la perte de données dus à des défaillances serveur fréquentes. Nous allons détailler le contexte de la migration, les difficultés rencontrées, la méthodologie utilisée, ainsi que l'architecture cible choisie pour cette transition.

CONTEXTE DE LA MIGRATION

L'entreprise gère une base de données relationnelle qui commence à rencontrer plusieurs limitations. En effet, les requêtes deviennent de plus en plus lentes à mesure que le volume des données augmente, et des défaillances serveur causent des pertes de données, ce qui perturbe les opérations de l'entreprise. Cette situation a conduit Paula DUPONT à envisager une migration vers une base de données NoSQL pour pallier ces problèmes.

Après analyse des besoins de l'entreprise et des spécificités du jeu de données (nommé ClassicModel), le choix s'est porté sur MongoDB, une base de données NoSQL flexible, capable de gérer efficacement de grandes quantités de données tout en offrant une meilleure résilience face aux défaillances serveur.

DIFFICULTÉS RENCONTRÉES

PROBLÈMES AVEC LA BASE DE DONNÉES RELATIONNELLE

Avant la migration, la base de données relationnelle était soumise à plusieurs limitations notables. Les principales difficultés rencontrées étaient :

- **Latence des requêtes** : À mesure que les données s'accumulaient, les performances des requêtes se dégradaient, rendant certaines opérations critiques lentes, ce qui affectait l'expérience utilisateur
- **Perte de données** : En raison des défaillances fréquentes des serveurs, des pertes de données surviennent souvent, ce qui représente un risque important pour l'intégrité du système.
- **Scalabilité limitée** : La base de données relationnelle ne répondait pas bien à la nécessité d'évoluer avec la croissance continue des données.

DÉFIS DE LA MIGRATION

La migration vers MongoDB a présenté plusieurs défis techniques :

- **Adaptation du modèle relationnel vers MongoDB** : La structure de données relationnelle, avec ses tables et ses relations entre elles, a dû être transformée en une structure NoSQL orientée document. Cela a demandé une réflexion approfondie pour déterminer comment grouper les données et les structurer efficacement.
- **Maintien de la cohérence des données** : Contrairement aux bases relationnelles qui garantissent une forte cohérence, MongoDB fonctionne sur un modèle de cohérence éventuelle. Cette différence a nécessité une gestion minutieuse des données durant la migration pour éviter toute incohérence.
- **Performance pendant la migration** : Il a fallu s'assurer que la migration n'affecte pas la disponibilité du service, ce qui a impliqué la mise en place de tests et de validations tout au long du processus.

MÉTHODOLOGIE DE MIGRATION

ÉTAPES DE LA MIGRATION

La migration a été réalisée en plusieurs étapes clés :

- **Analyse des besoins et choix de MongoDB :** En raison de ses capacités de scalabilité horizontale et de sa flexibilité pour gérer des données non structurées, MongoDB a été sélectionné comme SGBD NoSQL cible.
- **Planification de la migration :** Nous avons divisé la migration en phases pour assurer une transition en douceur. La première phase a consisté à préparer les données pour être transférées dans MongoDB, en définissant un nouveau modèle de données basé sur la structure du jeu de données ClassicModel.
- **Migration des données :** Nous avons utilisé des outils de transformation et de chargement de données pour convertir les données des tables relationnelles en documents MongoDB. Les tables ont été regroupées en ensembles logiques afin de faciliter leur gestion. Par exemple, les entités Orders, Products et OrderDetails ont été regroupées dans un même document, tout comme les entités Customers et Payments.
- **Validation et tests :** Après chaque étape, des tests ont été effectués pour s'assurer que la migration ne provoquait pas de perte de données et que les performances étaient conformes aux attentes.

OUTILS UTILISÉS POUR LA MIGRATION

Pour effectuer la migration, nous avons utilisé des outils d'ETL (Extract, Transform, Load) permettant d'extraire les données de la base relationnelle, de les transformer selon le modèle MongoDB, et de les charger dans la nouvelle base. En particulier, des scripts personnalisés ont été développés pour automatiser la conversion des données et assurer la cohérence des informations migrées.

L'ARCHITECTURE CIBLE

ARCHITECTURE NOSQL CHOISIE

La base de données NoSQL choisie est MongoDB, une base orientée document. Ce logiciel a été retenue pour plusieurs raisons :

- **Scalabilité horizontale** : MongoDB permet d'ajouter facilement de nouveaux serveurs pour étendre la capacité de la base de données, ce qui répond aux besoins croissants de l'entreprise.
- **Flexibilité** : Contrairement aux bases de données relationnelles, MongoDB permet de gérer des données semi-structurées et non structurées de manière plus souple, ce qui est particulièrement adapté aux exigences du jeu de données ClassicModel.
- **Haute disponibilité** : MongoDB offre une gestion de la réplication et des mécanismes de tolérance aux pannes, ce qui réduit le risque de perte de données en cas de défaillance serveur.

AVANTAGES DE L'ARCHITECTURE NOSQL

L'architecture MongoDB offre plusieurs avantages par rapport à la base de données relationnelle initiale :

- **Performance améliorée** : La gestion de grandes quantités de données non structurées devient plus efficace, avec des temps de réponse plus rapides.
- **Facilité de gestion des relations entre les données** : En regroupant certaines tables comme **Orders**, **Products**, et **OrderDetails**, ou **Customers** et **Payments**, nous avons simplifié l'architecture tout en conservant la cohérence des données.
- **Tolérance aux pannes et résilience** : MongoDB garantit une meilleure disponibilité des données en répliquant automatiquement les informations sur plusieurs serveurs.

COMPARAISON AVEC L'ARCHITECTURE RELATIONNELLE

Contrairement à la base relationnelle, MongoDB ne repose pas sur un modèle de données rigide avec des relations fixes entre les tables. Cette flexibilité permet une meilleure gestion des données en fonction des besoins spécifiques de l'entreprise, notamment pour les applications nécessitant une haute disponibilité et une gestion de données volumineuses.

CONCLUSION ET PERSPECTIVES

RÉSUMÉ DES BÉNÉFICES DE LA MIGRATION

La migration vers MongoDB a permis de résoudre plusieurs problèmes rencontrés avec la base de données relationnelle :

- Les performances des requêtes ont été nettement améliorées grâce à la scalabilité de MongoDB.
- La perte de données due aux défaillances serveur a été réduite grâce à la réplication des données.
- L'architecture NoSQL a apporté plus de flexibilité pour gérer les données non structurées et semi-structurées.

DIFFICULTÉS RESTANTES ET PERSPECTIVES FUTURES

Bien que la migration ait été un succès, certaines difficultés demeurent :

- La gestion de la cohérence des données dans un environnement distribué nécessite une attention continue.
- L'équipe devra s'adapter à l'architecture NoSQL et à la gestion des nouvelles fonctionnalités de MongoDB.

À l'avenir, il sera nécessaire de surveiller les performances de la base de données et d'optimiser les requêtes MongoDB pour s'assurer que l'entreprise continue de bénéficier d'une architecture scalable et performante.