

Високопродуктивна система управління потоками даних на основі MySQL і MongoDB

Вінницький національний технічний університет

Анотація

За результатами проведеного аналізу було виявлено переваги та недоліки реляційних та документо-орієнтованих баз даних та розглянуті підходи до створення системи з їх врахуванням для підвищення швидкодії.

Ключові слова: реляційна база даних, документо-орієнтована база даних, SQL, NoSQL, MySQL, MongoDB, мікросервісна архітектура.

Abstract

According to the results of the analysis revealed the advantages and disadvantages of relational and document-oriented databases and considered approaches to creating a system based on them to improve performance.

Keywords: relational database, document-oriented database, SQL, NoSQL, MySQL, MongoDB, microservice architecture.

Актуальність обраної тематики зумовлена необхідністю в підвищенні продуктивності серверних систем управління даними. Часто конкретні бази даних застосовують для цілей в яких вони показують низьку продуктивність зумовлюючи це тим, що з одною базою даних простіше працювати. І це дійсно так, більшість мов програмування надають інструменти для роботи лише з одною базою даних в одному додатку, і конфігурація більше ніж одної бази викликає складнощі. Однак при використанні мікросервісної архітектури ця проблема зникає.

При такому підході можна розділити потоки даних для роботи з двома базами даних SQL та NoSQL. Кожна з них має свої переваги та недоліки які потрібно врахувати при проектуванні системи.

Всім відомі реляційні бази даних мають ряд переваг. Вони прості в розумінні, оскільки складаються з єдиної інформаційної конструкції — таблиці. При проектуванні таких баз даних використовуються правила які базуються на математичному апараті, наприклад різноманітні нормальні форми. Реляційна модель забезпечує незалежність даних, при зміні структури бази зміни, які потрібно провести в прикладній програмі мінімальні. Реляційні бази даних досить швидко обробляють запити на оновлення даних.

Однак реляційні бази даних мають також і недоліки. При запиті даних з двох зв'язаних таблиць приходиться операція з'єднання яка є досить повільною. Для організації зв'язків один-до-одного, багато-до-одного та одного-до-багатьох необхідні додаткові поля-ключі, а у випадку зв'язку багато-до-багатьох і ціла таблиця. Всі ці додаткові поля збільшують розмір бази даних на накопичувачі.

З приходом NoSQL баз даних, таких як MongoDB розпочав поширюватись документо-орієнтований підхід до проектування баз даних. Такі бази даних оперують поняттями колекцій і документів. Форма і поля документа не обмежується заданою структурою колекції, як це є в таблицях і їх полях. В MongoDB документи зберігаються в форматі BSON (Binary JSON), що є надзвичайно зручним при групуванні даних в складні структури. При такому підході в базі даних відсутні операції з'єднання, що пришвидшує запити на отримання та збереження складних структур даних. Відсутність додаткових полів та таблиць зменшує розміри такої бази даних, за умови, що в ній дані мінімально дублюються.

До недоліків документо-орієнтованих баз даних можна віднести низьку швидкодію при оновленні полів в документах, а також те, що кожен документ містить в собі назву поля, що дещо збільшує розміри даних. Також відсутня пряма підтримка складних транзакцій.

Тому в складних системах, де дані можуть приймати різні форми в залежності від переваг різних підходів до організації баз даних слід розділити їх потоки в різні бази. При використанні мікросервісної архітектури це буде досить просто зробити, для цього потрібно створити сервіс з реляційною базою

даних, сервіс з документо-орієнтованою базою даних та сервіс який буде регулювати запити до цих баз.

Основний принцип полягатиме в тому, щоб зберігати складні комплексні об'єкти та дані які часто записуються і не редагуються в MongoDB, а прості об'єкти які часто обновлюються в SQL базі. Також дані з полями які часто дублюються доцільніше буде зберігати в SQL базі для економії простору на накопичувачі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Garcia-Molina H., Ullman J. D., Widom J., Database Systems The Complete Book Second Edition. — Department Of Computer Science Stanford University, 2009. — 1240 с.
2. Banker K., MongoDB in Action. — Manning Publications 2011.— 312 с.
3. Nadareishvili I., Mitra R., McLarty, M., Amundsen M., Microservice Architecture: Aligning Principles, Practices, and Culture. — O'Reilly Media 2016. — 146 с.

Шевчук Максим Олегович — студент групи ІКІ-20м, кафедри комп'ютерної інженерії ВНТУ, Вінницький національний технічний університет, м. Вінниця, e-mail: m.shevchuk.o@gmail.com

Науковий керівник: **Ткаченко Олександр Миколайович** — професор кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця, e-mail: alextk1960@gmail.com

Shevchuk Maksym Olehovich — student of the Computer Techniques Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: m.shevchuk.o@gmail.com

Research advisor: **Tkachenko Oleksandr Mykolaiovych** — Professor of the Computer Techniques Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: alextk1960@gmail.com