І.С. Зінов'єва

МОЖЛИВОСТІ РОБОТИ З ВЕЛИКИМИ ДАНИМИ В СЕРЕДОВИЩІ MONGODB

Віртуалізація та діджіталізація усіх сфер діяльності, розширення можливостей доступу до глобальної мережі Інтернет і напрямів його використання, зростання операційних можливостей обчислювальної техніки сприяє пришвидшенню темпів накопичення інформації. Наприклад, сучасна сонячна електростанція обладнання сотнею сонячних панелей, складним високотехнологічним устаткуванням, датчиками, інверторами здатна продукувати терабайти та петабайти даних (так звані великі дані). При цьому, формується неоднорідна структура даних, у рази збільшуються обсяги неструктурованої інформації.

У роботі з великими даними висувається ряд обов'язкових вимог, яким мають задовольняти інструменти їх обробки: швидкість та точність обробки, гнучкість систем управління даними, продуктивність, якість результатів тощо. Організація роботи та управління великими даними відбувається на кількох рівнях: обладнання, аналітики, баз даних, алгоритмів тощо. Саме бази даних вважаються найбільш «вузьким» місцем в роботі з великими даними [1].

Відмінність документо-орієнтованої нереляційної бази даних MongoDB від традиційних SQL баз даних полягає в особливостях зберігання даних. У MongoDB та інших нереляційних базах даних (званих NoSQL) відсутня жорстка схема зберігання даних. Фактично, у MongoDB усі дані зберігаються у вигляді документів, що наповнюють колекції (рис. 1). З точки зору розробника ПЗ документо-орієнтовані дані є більш простими і значно гнучкими в управлінні, ніж реляційні дані. На відміну від реляційних даних, що зберігаються за жорсткою фіксованою структурою у таблицях (рядках, стовпчиках) із наперед визначеними зв'язками, документи створюються індивідуально і можуть зберігати будь-які дані, в т.ч. і неструктуровані. При цьому, дотримання структури документу не вимагається, поля документів у межах колекції можуть мати довільні формати, бути пустими, мати значення NULL, або взагалі бути відсутніми в окремих документах. Єдиним обов'язковим полем документа є поле ключ (_id), яке містить унікальний набір символів. При чому, MongoDB створює поле ключ автоматично, якщо розробник цього не зробив. Таким чином досягається гнучкість структури даних, зберігаючи, при цьом,у відповідність єдиної базової моделі.

Ідеологія роботи із даними в MongoDB підпорядковується так званій концепції CRUD (абревіатура від скорочення назв основних операцій роботи із БД: створення (create), читання (read), оновлення (update), видалення (delete)) [2]. Для виконання запитів використовується Mongodb query language (мова запитів).

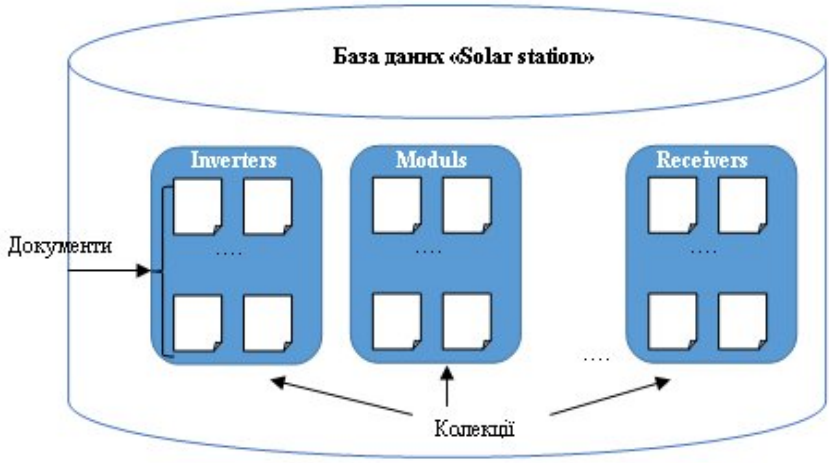


Рис. 1. Графічна схема структури нереляційної бази даних MongoDB

Оперувати великими даними в MongoDB можна кількома способами: 1) безпосередньо через консоль (командну строку); 2) через графічний клієнт. Існує значна кількість різноманітних графічних клієнтів, які працюють із MongoDB: Compass, Robo 3T, 3T MongoChef, RockMongo, UMongo, NoSQL Viewer та ін. Використання першого способу орієнтоване на розробників і потребує вмінь роботи із об'єктами JSON і функціями JavaScript. Другий спосіб розрахований на користувачів баз даних, оскільки графічний клієнт має інтуїтивний та зручний інтерфейс, який забезпечує доступ до об'єктів даних. Наприклад, використання графічного клієнта Compass значно спрощує процедуру введення і видалення даних у середовищі MongoDB, має потужний інструментарій візуалізації даних та, що є досить важливим, забезпечує простоту та зручність обробки даних.

MongoDB підтримує розподілене управління великими даними. Окрім взаємодії через архітектуру «клієнт-сервер», нереляційна база даних підтримує роботу в хмарі (за допомогою підключення до MongoDB Atlas Cluster). Проведені дослідження показали, що ключовими перевагами використання MongoDB в роботі з великими даними: адаптивність, горизонтальна масштабованість, висока швидкість обробки даних, гнучка структура даних. Використання MongoDB в роботі з великими даними сприяє оптимізації операційних процесів та підвищенню ефективності управління.

[1] Ahmad Abdullah, Qingfeng Zhuge, «From Relational Databases to NoSQL Databases: Performance Evaluation», Research Journal of Applied Sciences, Engineering and Technology, no. 11(4), p. 434-439, 2015

[2] MongoDB: Open Source Document Database, 2019. [Online]. Available: https://www.mongodb.com. Accessed on: April 21, 2019