

Порівняльна характеристика реляційних та NoSQL баз даних

Вінницький національний технічний університет

Анотація

Досліджено переваги та недоліки реляційних та нереляційних NoSQL баз даних. Розглянуто відмінності властивостей ACID і BASE.

Ключові слова: ACID, SQL, NoSQL, BASE, транзакція.

Abstract

The advantages and disadvantages of relational and non-relational NoSQL databases are explored. The differences of ACID and BASE properties are considered.

Keywords: ACID, SQL, NoSQL, BASE, transaction.

Вступ

З розвитком цифрових технологій актуальною стала проблема в зберіганні та обробці великих об'ємів даних, які невпинно зростають. Так, за прогнозами експертів міжнародної дослідницької компанії IDC вже за період з 2018 по 2023 рік обсяг усієї світової інформації, що зберігається на електронних носіях, повинна подвоїтись і становитиме близько 11,7 зеттабайт (10^{21} байт) [1]. Об'єм дискової пам'яті, необхідний для рішення типових задач у корпоративному сегменті, як правило, не перевищує 72 Тбайт (10^{12} байт), однак для вирішення задач сфери великих даних необхідно до 300 Пбайт (10^{15} байт) [2]. Згідно з іншим дослідження компанії IDC [3] до 2025 року близько 80% усієї світової інформації становитиме корпоративна інформація, а тому зростатиме і відповідальність компаній за захист даних. Крім того, компанії висувають підвищені вимоги й до швидкості обробки даних. Таким чином, для задоволення зростаючих потреб користувачів необхідно обирати продуктивну модель організації баз даних. За даними ресурсу [4] реляційна модель даних все ще займає провідне місце, однак тренди свідчать про поступове зростання ролі нереляційних NoSQL баз даних, що зумовлено швидким розвитком сегменту великих даних, інтернету речей тощо, де дані є слабо структурованими, а головною вимогою є їх швидка обробка. Тому дослідження переваг та недоліків реляційних і нереляційних баз даних є актуальною задачею.

Результати дослідження

Реляційна модель даних базується на математичній теорії (теорія множин, теорія відношень, теорія логіки), тоді як нереляційні бази даних (БД) не мають єдиної математичної теорії. Реляційна модель даних дозволяє забезпечити базі даних надійність, гнучкість, стабільність.

Надійність моделі БД визначається за допомогою властивостей ACID (Atomicity, Consistency, Isolation, Durability) [5]:

- **Нерозривність (Atomicity)** найкраще характеризується висловом «все або нічого», тобто будь-яка транзакція виконується повністю, або не виконується взагалі.
- **Узгодженість (Consistency)** передбачає, що будь-яка транзакція повинна зберігати узгодженість БД.
- **Ізольованість (Isolation)** гарантує незалежне виконання транзакцій та недоступність проміжних результатів виконання незавершеної транзакції.
- **Стійкість (Durability)** гарантує постійне зберігання в пам'яті результатів успішно завершених транзакцій, навіть після потенційних збоїв.

Коли набір даних організований за допомогою формально описаних таблиць, до елементів такої БД легко отримати доступ, створити нові дані, розширити структуру БД. Сьогодні такий тип зберігання інформації є найпоширенішим.

Однак, щоб відповідати потребам сучасних додатків, які оперують великими обсягами неструктурованих даних, часто вдаються до використання нереляційної БД. Нереляційні БД також відомі як NoSQL (не тільки SQL). Ці БД забезпечують гнучке масштабування, тобто вони

збільшуються прозоро, додаючи новий вузол і зазвичай розробляються з використанням недорогого обладнання.

На відміну від реляційних NoSQL бази даних часто базується на властивостях BASE [6]:

- Базова доступність (*basic availability*) – кожен запит гарантовано завершиться (успішно чи неуспішно).
- Гнучкий стан (*soft state*) - стан системи може змінюватися з часом, навіть без введення нових даних, для досягнення узгодженості даних у часі.
- Кінцева узгодженість (*eventual consistency*) – дані можуть залишатись неузгодженими, але через деякий час приходять у стан узгодженості.

Однак системи на основі BASE не можуть застосовуватись в додатках, де узгодженість даних є надзвичайно важливою, наприклад банківська та біржова сфери. У цьому сенсі реляційна транзакційна система забезпечує вищий рівень надійності даних.

Реляційні бази даних використовують структуровану мову запитів SQL, яка є універсальною стандартизованою та містить багатий функціонал для маніпулювання даними, складання звітів, управління транзакціями та інше. NoSQL бази даних використовують власний API та власну мову запитів, яка насправді є урізаною версією SQL [7]. Оскільки велика кількість NoSQL баз даних відносяться до виду «ключ-значення», то обмежений функціонал, якого цілком достатньо для роботи з такими базами даних, спрощує роботу оператора бази даних і визначає низький поріг входження. Крім того, простіший синтаксис зумовлює меншу кількість помилок. Однак, NoSQL додатки є вузькоспеціалізованими і в перспективі розвитку та розширення бази даних можуть не забезпечити нові вимоги до додатку, на відміну від реляційних баз даних, які використовують повноцінну SQL.

З лавиноподібним зростанням інформації в світі і необхідності її обробляти за розумний час постала проблема вертикальної масштабованості – зростання швидкості процесора зупинився на 3.5 ГГц, швидкість читання з диска також зростає тихими темпами, а ціна потужного сервера завжди більше сумарної ціни декількох простих серверів. У цій ситуації звичайні реляційні бази, навіть групові на масиві дисків, не здатні вирішити проблему швидкості, масштабованості і пропускну здатності. Єдиний вихід із ситуації – горизонтальне масштабування, коли кілька незалежних серверів з'єднуються швидкою мережею і кожен володіє/обробляє тільки частину даних і/або тільки частину запитів на читання-оновлення. У такій архітектурі для підвищення потужності сховища (ємності, часу відгуку, пропускну спроможності) необхідно лише додати новий сервер в кластер. Крім того, такий підхід вимагає менше фінансових ресурсів, ніж придбання більш потужного сервера.

Помітними перевагами NoSQL бази даних також є простіші процедури шарингу (розподіл інформації по різних вузлах мережі) та реплікації (копіювання даних на інші сервери), які є невід'ємними складовими розподілених систем.

На відміну від реляційних баз даних, процес розробки NoSQL баз даних є простішим, оскільки не потрібно визначати схему даних та зв'язки між даними. Однак, часто саме на етапі проектування визначаються потенційно слабкі місця обраного технічного рішення, що дозволяє вчасно відкоригувати стратегію розробки.

Реляційні бази даних є транзакційними. Транзакції дають можливість відмінити або відновити операції, якщо в системі сталась якась помилка. У випадку з нереляційними базами даними, розробнику потрібно самому дізнаватись, де є помилка та виправляти її. Також відсутність транзакцій може спровокувати втрату даних. Коли обидва розробники беруть якийсь запис з бази, обробляють його і додають до бази даних, то зміни одного з розробників затруть даними іншого розробника. Тобто не підтримується коректне опрацювання паралельної роботи різних користувачів з одним і тим же елементом даних.

Реляційні бази даних є типізованими, в той час коли нереляційні бази даних є динамічними. З однієї сторони це є перевагою, оскільки в реляційній базі зберігається об'єкт і він має чітку структуру, а в нереляційній базі даних в колекції можуть зберігатись 2 абсолютно різних об'єкта. А недоліком реляційної бази даних є те, що вона не така стійка до змін, як нереляційна база даних.

За рахунок відсутності зв'язків між даними, нереляційні бази даних забезпечують значно вищу швидкість пошуку та видачі результатів, порівняно з реляційними, що є без сумніву їх основною перевагою.

Висновки

В результаті проведеного аналізу було виявлено, що головними перевагами реляційної бази даних є те, що вони забезпечують високу надійність та узгодженість даних, є універсальними та підходять для рішення переважної більшості задач, де немає великої кількості даних. Реляційні бази даних є добре перевіреним інструментом, в той час як нереляційні бази даних є відносно новими рішеннями. Нереляційні бази даних є вузькоспеціалізованими, найкраще підходять для організації неструктурованих даних, забезпечують лінійну масштабованість та високу швидкість обробки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. IDC: объем всей хранимой в мире информации будет удвоен к 2023 году [Електронний ресурс] – Режим доступу: <https://regnum.ru/news/it/2574265.html>
2. Варламов Н. Взрывной рост трех V, или Большие данные и их хранение [Електронний ресурс] – Режим доступу: <http://lib.secuteck.ru/articles2/ip-security/vzryvnoy-rost-treh-v--ili-bolshie-dannye-i-ih-hranenie>
3. Объем данных в мире к 2025 году может вырасти до 175 зеттабайт [Електронний ресурс] – Режим доступу: <http://www.iksmedia.ru/news/5548977-Obem-dannyx-v-mire-k-2025-godu-mozh.html#ixzz6HONqsfhB>
4. DB-Engines [Електронний ресурс] – Режим доступу: <https://db-engines.com/>
5. Петух А.М. Базы данных. Мови запитів, управління транзакціями, розподілена обробка даних [Електронний навчальний посібник] / А.М. Петух, О.В. Романюк, О.Н. Романюк. – Режим доступу: https://web.posibnyky.vntu.edu.ua/fitki/11petuh_bazdanyh_movy_zalitiv/
6. NoSQL [Електронний ресурс] – Режим доступу: <https://ru.wikipedia.org/wiki/NoSQL>
7. Преимущества и недостатки нереляционных баз данных [Електронний ресурс] – Режим доступу: <https://veesp.com/ru/blog/sql-or-nosql>

Ковтун Богдан Валентинович – студент групи ЗПІ-18б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: kirpich1337228@gmail.com

Манич Андрій Михайлович – студент групи ЗПІ-18б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: anderss1918@gmail.com

Романюк Оксана Володимирівна – канд. техн. наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, e-mail: romaniukoksnav@gmail.com

Bogdan Kovtun - student of group ЗПІ-18b, Faculty of Information Technology and Computer Engineering, Vinnitsa National Technical University, Vinnitsa, e-mail: kirpich1337228@gmail.com

Andriy Manich - student of group ЗПІ-18b, Faculty of Information Technology and Computer Engineering, Vinnitsa National Technical University, Vinnitsa, e-mail: anderss1918@gmail.com

Oksana V. Romaniuk – Cand. Sc. (Eng), Assistant Professor of the Software Chair, Vinnitsia National Technical University, Vinnitsia, e-mail: romaniukoksnav@gmail.com.