УДК 378.147:004.94 **doi: 10.26906/SUNZ.2019.1.125**

Т. С. Нікітіна, О. І. Морозова

Національний аерокосмічний університет імені М. Є. Жуковського «ХАІ», Харків, Україна

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРОДУКТИВНОСТІ БАЗ ДАНИХ SQL ТА NOSQL

В роботі було проведено короткий аналіз функцій баз даних SQL та NoSQL, були приведені їх основні відмінності. На сьогоднішній день існують два найбільш поширених типу систем управління даними: реляційні бази даних та NoSQL. Існує величезне різноманіття моделей даних та API (Application Programming Interface) запитів для NoSQL. Зокрема для порівняння були обрані Apache Cassandra, DynamoDB, MongoDB. Модель даних та функціональність Apache Cassandra має схожість з іншими масштабованими сховищами. Оновлення та угруповання стовпців кешується в оперативній пам'яті, після чого скидаються на диск. Основною метою роботи було порівняння продуктивності реляційних SQL баз даних та NoSQL, на прикладі PostgreSQL, MySQL, Apach Cassandra, MongoDB, Amazon DynamoDB. Для тестування продуктивності було розроблено окремий програмний продукт. Основним предметом дослідження є продуктивність базових операцій цих систем. Результати про продуктивність кожної з них були отримані за допомогою розробленої системи тестування, адаптованої для потреб дослідження. Розроблена система тестування дозволила тестувати швидкість виконання складних аналітичних операцій, робити додаткові налаштування, використовувати великий обсяг даних. Система була розширена для виконання тестування розширеного набору операцій над схемою даних, що містить зв'язки між таблицями. Ця система тестування містить набір готових навантажень, які покривають основні аспекти функціонування й підтримують створені користувачем навантаження. За допомогою системи тестування були отримано дані про продуктивність представлених систем управління базами даних для набору різних запитів. Для аналізу продуктивності вимірювався час відгуку систем на запит – час між початком запиту й одержанням відповіді. Порівнювалися два види показників – середній відгук по виконані операції й деталізований аналіз. Отримані дані були представлені у вигляді діаграм, і по ним був зроблений висновок про продуктивність баз даних SQL та NoSQL. Вибір баз даних повинен максимально ґрунтуватися на типі вирішуваних завдань й також повинен враховувати обсяги даних, час відгуку системи.

Ключові слова: NoSQL база даних, реляційна база даних, тестування, продуктивність.

Вступ

У різних областях діяльності накопичується величезна кількість даних, що веде до посилення вимог до їх обробки та зберігання, зокрема до продуктивності систем управління базами даних (БД). Дана проблема особливо актуальна для даних, що вимагають глибокого аналізу. У зв'язку з цією ситуацією з'являються нові підходи до побудови таких систем, які повинні подолати недоліки існуючих. На сьогоднішній день існують два найбільш поширених типу систем управління даними: реляційні БД (РБД) [1, 2] та NoSQL БД, різні в багатьох аспектах роботи.

Такі кардинальні відмінності в питаннях, як надійність, гнучкість, узгодженість даних і масштабованість, вимагають ретельного аналізу різних моментів функціонування систем, особливо продуктивності. Однак існуючі дослідження на цю тему не в повній мірі охоплюють питання продуктивності двох підходів, обмежуючись порівнянням операцій, наданих NoSQL системами. У той же час великий спектр операцій, який реалізує реляційна система управління базами даних (СУБД), вимагає обчислень на стороні користувача, при роботі з NoSQL системою, що може привести до значних відмінностей в продуктивності. Мета даної роботи – провести дослідження продуктивності операцій СУБД цих двох типів систем.

В рамках роботи було проведено порівняний аналіз реляційної та NoSQL БД на прикладі MySQL, PostgreSQL, Apache Cassandra, MongoDB, Amazon DynamoDB. Основним предметом дослідження є продуктивність базових операцій цих систем. Результати про продуктивність кожної з них були отримані за допомогою розробленої системи тестування, адаптованої для потреб дослідження.

Підсумком роботи стали дані про продуктивність СУБД MySQL, PostgreSQL, Cassandra, DynamoDB, MongoDB отримані з використанням системи тестування. Система була розширена для виконання тестування розширеного набору операцій над схемою даних, що містить зв'язки між таблицями. На підставі отриманих даних про продуктивність операцій були зроблені висновки про ефективність досліджуваних СУБД.

Реляційні бази даних

РБД засновані на реляційній моделі та теорії множин. Таблиця складається з безлічі кортежів (записів), атрибути яких відповідають стовпцям. Така модель даних дуже точна та добре структурована. РБД гарантує високу надійність транзакцій завдяки повній підтримці чотирьох властивостей ACID (Atomicity, Consistency, Isolation, Durability):

- атомарність (Atomicity): якщо будь-яка частина транзакції не виконується, то не виконується транзакція цілком;
- узгодженість (Consistency): якщо БД знаходилася в узгодженому стані до виконання транзакції, то після виконання вона також буде знаходитися в узгодженому стані;
- ізольованість (Isolation): безліч транзакцій, що виконуються одночасно, не впливають на хід роботи один одного. Іншими словами, паралельні транзакції повинні бути серіалізовані;
- довговічність (Durability): зміни, вчинені транзакцією, будуть залишатися в системі не дивлячись на будь-які збої.

В роботі пропонується аналіз продуктивності РБД PostgreSQL [3] та MySQL. PostgreSQL є об'єктно-реляційною системою управління БД. Система розробляється більше 15 років та має перевірену архітектуру, яка заробила високу репутацію надійності, цілісності даних й точності.

Також PostgreSQL цілком підтримує ACID та стандарт ANSI-SQL. PostgreSQL містить в собі нетривіальні можливості, такі як управління конкурентним доступом за допомогою багатоверсійності (MVCC), повернення до станом в певний момент часу (Point-in-time recoverty), простір таблиць, асинхронні реплікації, внутрішні транзакції (точки збереження), резервне копіювання під час виконання, планувальник й оптимізатор запитів та ін. Всі ці функції РБД PostgreSQL дозволяють бути хорошою альтернативою NoSQL-системам в плані масштабованості, зберігаючи можливість складних глибоких аналітичних запитів за допомогою SQL.

MySQL – вільна реляційна система управління БД. Розробка та підтримка MySQL здійснює корпорація Oracle. Продукт поширюється як під GNU (General Public License), так і під власною комерційною ліцензією. MySQL є рішенням для малих і середніх додатків. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або видалені клієнти, проте в дистрибутив входить бібліотека внутрішнього сервера, що дозволяє включати MySQL в автономні програми. Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць.

Всі ці функції РБД PostgreSQL та MySQL дозволяють бути хорошою альтернативою NoSQL-системам в плані масштабованості, зберігаючи можливість складних глибоких аналітичних запитів за допомогою SQL.

NoSQL бази даних

В останні роки було розроблено велику кількість нових БД (рис. 1), які надають гарне горизонтальне масштабування для простих операцій читання, запису для БД, розподілених на безлічі серверів, орієнтованих на Big Data [4]. На відміну від них, традиційні БД дають менше можливостей до масштабування.

Рис. 1. Приклади NoSQL-систем

Безліч нових БД визначаються терміном «NoSQL». Термін «NoSQL», що розшифровується як «Not Only SQL» або «Not Relational». Зазвичай такі системи задовольняють такими ознаками: наявність засобів розподілу навантаження на безліч серверів; можливість розподілу даних на безліч серверів; простий протокол виклику операцій; більш слабка модель паралелізму, ніж ACID-транзакції; ефективне використання розподілених індексів й оперативної пам'яті для зберігання даних; відсутність фіксованої схеми даних.

NoSQL-системи зазвичай не задовольняють властивостям ACID-транзакцій: допускається узгодженість в кінцевому рахунку. Пропонується модель «BASE» (Basically Available, Soft state, Eventually consistent, узгодженість в кінцевому рахунку) в протилежність ACID.

Ідея полягає в тому, що, відмовившись від обмежень ACID, можна домогтися набагато кращої продуктивності й масштабованості. Більшість систем різняться в ступенях відмови від ACID. Прихильники NoSQL часто посилаються на CAP-теорему, яка стверджує, що система може задовольняти лише двом з трьох таких властивостей:

- узгодженість - дані завжди однакові для всіх реплік;
- доступність - дані завжди доступні користувачеві;
- стійкість до поділу - система БД продовжує коректну роботу не дивлячись на відмову мережі або вузлів.

У NoSQL системах зазвичай опускається узгодженість, але припущення можуть бути складніше. Найчастіше моделі даних в NoSQL-системах розбиваються на наступні категорії [4–7]:

- сховища типу «ключ-значення» (Key-value stores): зберігають значення й ідентифікатор для пошуку, заснований на заданому ключі;
- документно-орієнтовані сховища (Document stores): система зберігає дані в формі документів, які індексуються;
- сховища, що розширюються записом (Extensible records stores): записи в таких сховищах можуть бути розподілені вертикально та горизонтально по вузлах.

Існує величезне різноманіття моделей даних та API (Application Programming Interface) запитів для NoSQL БД. Зокрема для порівняння були обрані Apache Cassandra [6], DynamoDB [5], MongoDB [7]. Модель даних та функціональність Apache Cassandra [4] має схожість з іншими масштабованими сховищами. Оновлення та угруповання стовпців кешується в оперативній пам'яті, після чого скидаються на диск.

Присутня підтримка розподілу даних та механізми реплікації, автоматичне виявлення відмов вузлів та відновлення. Однак й модель паралелізму в Cassandra слабкіша, ніж в інших системах, в слідстві відсутності механізмів блокування асинхронного оновлення реплік. Для обробки даних Cassandra підтримує CQL – Cassandra Query Language, заснований на SQL.

Amazon DynamoDB [5] є продуктом компанії Amazon, основа якого є хмарне сховище даних типу «ключ-значення», де клієнт оплачує трафік, а не обсяг даних. Додатково компанія надає можливість розгорнути БД локально. Схемою зберігання даних в цій системі є запис, що містить ключ та деякий набір іменованих атрибутів будь-якої розмірності.

Серед особливостей DynamoDB можна виділити автоматичне кешування таблиць в оперативній пам'яті (Amazon DynamoDB Accelerator, DAX), масштабування в хмарі по заданій пропускній спроможності та можливість створення додатків-тригерів, що реагують на зміни даних. В даній роботі в якості системи тестування продуктивності обраних БД було створено окремий програмний продукт.

Аналіз продуктивності баз даних різних типів

Розроблено програму для тестування продуктивності БД, що дозволяє отримати дані щодо сильних та слабких сторін СУБД. Система тестування містить набір готових навантажень, які покривають основні аспекти функціонування (читання, запис, видалення, оновлення) й підтримують створені користувачем навантаження. Тестування проводилося на наступній конфігурації: одна машина (16-ти ядерний, Intel Core i9 процесор, 8GB RAM, диск розміру 450 GB) для запуску однієї з систем. Над БД виконувалися наступні операції: створення таблиці, запису даних, оновлення даних, читання даних, видалення даних, видалення таблиці. Операції виконувалися над БД наступних розмірів – середнім (500 Мб), великим (15 Гб).

Для аналізу продуктивності вимірювався час відгуку систем на запит – час між початком запиту й одержанням відповіді. Порівнювалися два види показників – середній відгук по виконані операції й деталізований аналіз.

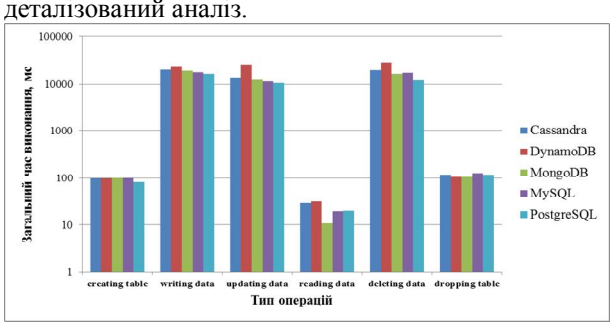

Рис. 2. Аналіз часу виконання операцій БД для середнього обсягу даних (500 Мб)

На рис. 2 представлено діаграму, яка відображує час відгуку системи в залежності від SQL операцій та типу БД для середнього обсягу даних (500 Мб). Для середнього обсягу даних РБД та NoSQL дають схожі часові відгуки.

Операції запису. Операції запису, оновлення, видалення даних потребують найбільшого часу на виконання для всіх типів БД. Найбільший час виконання операції запису, оновлення записів у DynamoDB при всіх обсягах даних. При середньому обсязі даних PostgreSQL та MySQL показує найкращий час виконання, ніж Cassandra, MongoDB. На великому обсязі даних швидше працює MongoDB (рис. 3).

Операції читання записів. Операції читання записів в БД PostgreSQL та MySQL виконуються швидше ніж NoSQL системах для середнього обсягу даних. DynamoDB значно програє в продуктивності ніж Cassandra, MongoDB та показує рівний результат при великому обсязі даних. Cassandra, MongoDB працюють значно шивдше при великому обсязі даних для простих операцій читання.

Операція поновлення записів. Згідно результатів тестування продуктивність БД DynamoDB нижче інших систем. Найкращий показник продуктивності має БД MongoDB та Cassandra для великого обсягу даних. Операції Update в БД PostgreSQL та MySQL виконуються швидше ніж в NoSQL системах для середнього обсягу даних.

Операції підрахунку з угрупованням. У зв'язку з тим, що дана операція виконується в PostgreSQL, MySQL на стороні сервера, а для NoSQL вона реалізована на стороні користувача. PostgreSQL демонструє кращу продуктивність, ніж NoSQL системи. В свою чергу, продуктивність Cassandra, MongoDB вище, ніж продуктивність DynamoDB.

Пропускна здатність PostgreSQL та MySQL зберігається при малому та середньому обсягах даних, знижується при великому обсязі.

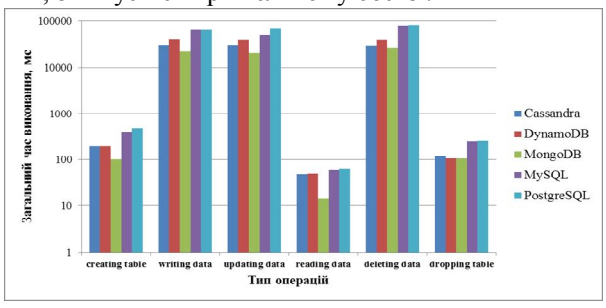

Рис. 3. Аналіз часу виконання операцій БД для великого обсягу даних (15 Гб)

Висновки

З результатів тестування продуктивності систем можна зробити висновок, що PostgreSQL, MySQL виконує прості операції читання, вставки й поновлення запису не гірше, а в деяких випадках краще, досліджуваних NoSQL систем. Операція читання з'єднаних таблиць виконується в реляційній БД повільніше за рахунок об'єднання таблиць всередині. Таке рішення, в свою чергу, несе збільшення обсягу збережених даних за рахунок надмірності. Операція підрахунку з угрупованням записів виконується PostgreSQL, MySQL швидше, так як при роботі з NoSQL основні обчислення залишалися на стороні користувача. Пропускна здатність PostgreSQL та MySQL зберігається при малому та середньому обсягах даних, знижується при великому обсязі. Операції читання записів в БД PostgreSQL та MySQL виконуються швидше ніж NoSQL системах для середнього обсягу даних, але Cassandra, MongoDB працюють значно швидше при

великому обсязі даних для простих операцій читання. Для операцій видалення при середньому обсязі даних PostgreSQL та MySQL показує найкращий час виконання. На великому обсязі даних швидше працює NoSQL системи працюють краще.

Таким чином, можна зробити висновок, що системи орієнтовані на Big Data можуть значно ефективніше працювати з NoSQL системами. Однак, типові рішення й системи для невеликих обсягів данних можуть мати ефективну продуктивність при роботі з БД MySQL, PostgreSQL. Таким чином, вибір БД повинен максимально грунтуватися на типі вирішуваних завдань й також повинен враховувати обсяги даних, час відгуку системи.

СПИСОК ЛІТЕРАТУРИ

1. Дейт К. Дж. Введение в системы баз данных = Introduction to Database Systems. - М.: Вильямс, 2005. - 1328 с.
2. Коваленко А.А. Сучасний стан та тенденції розвитку комп'ютерних систем об'єктів критичного застосування / А.А. Коваленко, Г.А. Кучук // Системи управління, навігації та зв'язку. – Полтава . ПНТУ, 2018. – Вип. 1(47). – С. 110-113.
3. PostgreSQL. [Електронний ресурс] - Режим доступу: - http://www.postgresql.org/.
4. Apache HBase. [Електронний ресурс] - Режим доступу: - https://hbase.apache.org/..
5. Amazon DynamoDB. [Електронний ресурс] - Режим доступу: - https://aws.amazon.com/documentation/dynamodb/.
6. Apache Cassandra. [Електронний ресурс] - Режим доступу: - http://cassandra.apache.org/. – 25.01.2019.
7. MongoDB Atlas. [Електронний ресурс] - Режим доступу: - https://www.mongodb.com/. – 25.01.2019.

Рецензент: д-р техн. наук, проф. К. С. Козелкова,
Державний університет телекомунікацій, Київ
Received (Надійшла) 11.12.2018
Accepted for publication (Прийнята до друку) 23.01.2019

Сравнительный анализ производительности баз данных sql и nosql

Т. С. Никитина, О. И. Морозова

В работе было проведено краткий анализ функций баз данных SQL и NoSQL, были приведены их основные отличия. На сегодняшний день существуют два наиболее распространенных типа систем управления данными: реляционные базы данных и NoSQL. Существует огромное многообразие моделей данных и API (Application Programming Interface) запросов для NoSQL. В частности, для сравнения были выбраны Apache Cassandra, DynamoDB, MongoDB. Модель данных и функциональность Apache Cassandra имеет сходство с другими масштабируемыми хранилищами. Обновления и группировки столбцов кэшируется в оперативной памяти, после чего сбрасываются на диск. Основной целью работы было сравнение производительности реляционных SQL баз данных и NoSQL, на примере PostgreSQL, MySQL, Apach Cassandra, MongoDB, Amazon DynamoDB. Для тестирования производительности был разработан отдельный программный продукт. Основным предметом исследования является производительность базовых операций этих систем. Результаты о производительности каждой из них были получены с помощью системы тестирования, адаптированной для нужд исследования. Разработанная система тестирования позволила тестировать скорость выполнения сложных аналитических операций, делать дополнительные настройки, использовать большой объем данных. Система была расширена для выполнения тестирования расширенного набора операций над схемой данных, содержащий связи между таблицами. Эта система тестирования содержит набор готовых нагрузок, которые покрывают основные аспекты функционирования и поддерживают созданные пользователем нагрузки. С помощью системы тестирования были получены данные о производительности представленных систем управления базами данных для набора различных запросов. Для анализа производительности измерялся время отклика систем на запрос - время между началом запроса и получением ответа. Полученные данные были представлены в виде диаграмм, и по ним был сделан вывод о производительности баз данных SQL и NoSQL. Выбор баз данных должен максимально основываться на типе решаемых задач и также должен учитывать объемы данных, время отклика системы.

Ключевые слова: NoSQL база данных, реляционная база данных, тестирование, производительность.

Comparative analysis of the Sql and nosql database productivity

T. S. Nikitina, O. I. Morozova

A brief analysis of SQL and NoSQL database functions was performed in this work, their main differences were cited. To date, there are two most common types of data management systems: relational databases and NoSQL. There is a huge variety of data models and API (Application Programming Interface) requests for NoSQL. In particular, for comparison, Apache Cassandra, DynamoDB, MongoDB were selected. The data model and functionality of Apache Cassandra are similar to other scalable repositories. Updates and grouping of columns are cached in RAM and then reset to disk. The main purpose of the work was to compare the performance of relational SQL databases and NoSQL databases, for example, PostgreSQL, MySQL, Apach Cassandra, MongoDB, Amazon DynamoDB. A separate software product was developed for testing the performance. The main subject of the study is the performance of these systems basic operations. Performance results for each of them were obtained using testing system adapted for research purposes. The developed system of testing allowed testing the speed of complex analytical operations, making additional settings, using the large amount of data. The system has been expanded for testing the extended set of operations over a data plan that contains links between tables. This testing system contains a set of ready-made loads that cover the main aspects of the operation and support user-generated load. The testing system received data on the performance of the presented database management systems for a set of different queries. For performance analysis, the system response time (the time between the beginning of the request and the response) was measured on request. Two types of indicators were compared - the average response to completed operations and detailed analysis. The obtained data was presented in the form of diagrams, and by them was made a conclusion on the performance of SQL and NoSQL databases. The choice of databases should be based as much on the type of tasks to be solved and should also take into account the amount of data and the system response time.

Keywords: NoSQL database, relational database, testing, performance.