

*Шаров С.В., канд. пед. наук, доцент; Петровський В.В., магістрант
Мелітопольський державний педагогічний університет імені Богдана Хмельницького,
м. Мелітополь
Кафедра інформатики і кібернетики*

ОГЛЯД НЕРЕЛЯЦІЙНИХ БАЗ ДАНИХ

Актуальність. Сьогодні більшість організацій та підприємств використовують значні обсяги інформації, що зберігаються у базах даних (БД) та обробляються системами управління базами даних (СУБД). Звісно, найбільш розповсюдженими є реляційні БД, такі як Microsoft SQL Server, Oracle, MySQL або Access. Але в той же час, на ринку програмних продуктів існує значний пласт баз даних, які не використовують реляційну модель даних, винайдену та обґрунтовану Е. Коддом. Мова йде про NoSQL бази даних, які мають свої переваги, недоліки та відповідну сферу застосування.

Мета статті полягає у висвітленні особливостей різних типів нереляційних баз даних NoSQL.

Виклад основного матеріалу. Термін NoSQL (Not Only SQL) не має загальновизнаного визначення. Він був введений Еріком Евенсом наприкінці 90-х років, проте набув свого сучасного сенсу з 2009 року. Слід зазначити, що поява нереляційних баз даних була викликана декількома проблемами у використанні реляційних баз даних у межах роботи веб-додатків, а саме:

- значна кількість високонавантажених веб-додатків обробляють такі об'єми даних, які на порядок більше тих, що можуть бути оброблені реляційними базами даних;
- реляційні системи управління базами даних не завжди здатні забезпечити одночасний доступ до даних мільйонам користувачів Internet;
- на сьогодні існує потреба в обробці нетабличних даних [4].

На відміну від реляційних баз даних, NoSQL не гарантує дотримання вимог ACID (Atomicity, Consistency, Insulation, Durability – Атомарність, Узгодженість, Ізольованість, Довговічність). Замість цього Еріком Брюєром, автором теореми CAP (евристичне твердження про те, що у будь-якій реалізації розподілених обчислень можливо забезпечити не більше двох з трьох властивостей), було висунуто набір вимог до баз даних NoSQL:

- базова доступність (basic availability) – кожен запит гарантовано завершується (успішно чи безуспішно);
- гнучкий стан (soft state) – стан системи може змінюватися з часом, навіть без введення нових даних, для досягнення узгодження даних;
- узгодженість у кінцевому результаті (eventual consistency) – дані будуть обов'язково узгодженими через деякий час [7].

До основних видів NoSQL баз даних відносяться наступні: сховища даних «ключ-значення», стовбцеві (колоночні) СУБД, документо-орієнтовані СУБД, графо-орієнтовані СУБД, об'єктно-орієнтовані СУБД. Дамо коротку характеристику кожного з цих баз даних.

За думкою деяких адміністраторів баз даних, сховища типу «ключ-значення» є дуже простою моделлю даних. У ній всі ключі представлені у

формі значень, як в хеш-таблиці, що дозволяє отримати досить гнучку побудову даних. Представником такого типу БД є Redis – він досить швидкий, в ньому можна робити вибір між надійністю та швидкістю. Також серед переваг Redis можна виділити високу продуктивність, роботу зі списками та стеками, наявність клієнтських API для різних платформ (PHP, Ruby, Python, Perl, Java) [3, с. 358].

Також зарекомендували себе хранилища даних за моделлю «ключ-значення» MemcacheDB та BerkeleyDB. Перевагами СУБД MemcacheDB можна назвати високу швидкість запису/читання даних, високу надійність та швидкість доступу до інформації, можливість реплікацій, підтримка транзакцій, наявність офіційного релізу. Досить перспективним є база даних BerkeleyDB, яка має вже кілька версій і продовжує розвиватися. До переваг даного продукту варто віднести підтримку реплікації та транзакцій. Вона часто використовується як платформа для зберігання даних, на якій будується операційна та інтерфейсна частини.

Головна ідея стовбцевих СУБД полягає у зберіганні даних не по рядках, як це реалізовано у реляційних СУБД, а по стовбцях. Стовбцеві сховища засновані на зчитуванні стовбців з багатьох рядків одночасно, тому одиницею виміру є група стовбців. БД даного типу складається у більшості випадків із 2 рівнів агрегатних станів, які, в свою чергу, складаються із асоціативного масиву розширених значень [5, с. 43]. З точки зору клієнта дані представляються у вигляді таблиці, але фізично ці таблиці є сукупністю колонок, кожна з яких, по суті, являє собою таблицю з одного поля. При цьому фізично на диску значення одного поля зберігаються послідовно один за одним. Така організація даних призводить до суттєвого зниження завантаженості на сервер БД при обробці даних. Крім того, при стовпчиковому зберіганні з'являється можливість компресії даних, так як в одній колонці таблиці дані, як правило, однотипні. Недоліком СУБД такого типу є низька швидкість виконання операцій запису даних при збільшенні об'єму БД.

Під документо-орієнтованими СУБД розуміються комп'ютерні системи, розроблені для зберігання, одержання та управління документо-орієнтованої чи слабоструктурованої інформації. Ці системи оперують абстрактним поняттям «Документ», в якому відбувається інкапсуляція та кодування інформації, що зберігається у стандартному форматі (XML, YAML, JSON, BSON, бінарні формати PDF, документи Microsoft Office та ін.). Документи усередині документно-орієнтованих БД деяким чином схожі на записи реляційних БД, але є більш гнучкими. Вони не вимагають наявності одних і тих же розділів, частин, ключів тощо. Документи адресуються у БД допомогою унікального ключа, який представляє конкретний документ. Часто роль ключа виконує звичайний рядок, шлях до файлу тощо. Зазвичай, документо-орієнтовані СУБД будують індекси за таким ключам, що дозволяє досить швидко отримати документ з бази даних. Прикладами документно-орієнтованих СУБД є MongoDB, IBM Lotus Notes, CouchDB, Oracle NoSQL та ін. [2, с. 42].

MongoDB є СУБД з відкритим вихідним кодом, яка розроблена на мові C++. Вона управляє наборами документів, які зберігаються у двійковому вигляді у форматі BSON. Ця СУБД не підтримує транзакції та не вимагає опису схеми таблиць. Робота з документами проводиться у режимі асинхронної

реплікації «ведучий – ведений», при цьому документ оновлюється цілком. Багато користувачів працюють з одним документом одночасно. Основна ідея CouchDB полягає у збереженні даних в формі документів, що дає можливість багаторазового використання [6, с. 148].

У графо-орієнтованій БД дані представлені у вигляді набору вузлів, кожний з яких може зберігати дані. Крім того, кожен вузол може мати деяку кількість зв'язків з вузлами-нащадками. Часто вузол містить список зв'язків з численними вузлами того ж типу, що дозволяє представити зв'язки у відношенні «багато-до-багатьох». До переваг графових моделей БД, у порівнянні з реляційною моделлю даних, слід віднести пошук шляхів, виділення співтовариств, гнучку схему даних, можливість уніфікувати зберігання різнорідних об'єктів. Прикладами графових СУБД є AllegroGraph, BigData, InfiniteGraph [1, с. 25].

Об'єктно-орієнтована база даних (ООБД) – база даних, яка складається з методів та класів. ООБД працюють з об'єктами баз даних так само, як з об'єктами при об'єктно-орієнтованому програмуванні. Деякі ООБД розроблені для щільної взаємодії з такими мовами програмування, як Java, C #, C ++; інші мають свої власні мови програмування. Прикладами ООБД є Jasmine, ObjectDB, Caché, Matisse.

Висновки. Отже, сьогодні існує певна кількість баз даних, які використовують нереляційний підхід для збереження та обробки даних (MongoDB, IBM Lotus Notes, CouchDB, Oracle NoSQL, Jasmine, ObjectDB, Caché, Matisse). Всі вони отримали загальну назву NoSQL баз даних. Відносно недорогі та високопродуктивні, бази даних типу NoSQL мають горизонтальну масштабованість даних; гнучкі схеми та моделі даних; прості низькорівневі інтерфейси для запитів. Однак, їх розвиток ще продовжується та потребує подальшого обґрунтування і теоретичної розробки.

Література

1. Гортон И. Системы для больших данных: конвергенция архитектур / И. Гортон, Д. Клейн // Открытые системы. СУБД. – 2015. – № 3. – С. 24-28.
2. Курцев Ю.В. Виды NoSql и их сравнение с реляционными базами данных / Ю.В. Курцев, Г.П. Цапко // Молодежь и современные информационные технологии: сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых (г. Томск, 12-14 ноября 2014 г.). – Томск: Изд-во ТПУ, 2014. – С. 42-43.
3. Редмонд Э. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL / Эрик Редмонд, Джим. Р. Уилсон. – М.: ДМК Пресс, 2013. – 384 с.
4. Селезнев К. От SQL к NoSQL и обратно [Электронный ресурс] / Константин Селезнев. – Режим доступа: <http://www.osp.ru/os/2012/02/13014127>.
5. Фаулер М. NoSQL: новая методология разработки нереляционных баз данных / Мартин Фаулер, Прамодкумар Дж. Садаладж. – М.: ООО «И.Д. Вильямс», 2013. – 192 с.
6. Хоган Б. Книга веб-программиста. Секреты профессиональной разработки веб-сайтов / Б. Хоган, К. Уоррен, М. Уэбер, К. Джонсон, А. Годин. – СПб.: Питер, 2013. – 288 с.
7. NoSQL [Электронный ресурс] – Режим доступа: <http://en.wikipedia.org/wiki/NoSQL>.