УДК 004.428

А. Ю. Веретко

ДОСЛІДЖЕННЯ ПЕРЕВАГ NOSQL БАЗ ДАНИХ ПРИ РОЗРОБЦІ СЕРВЕРНИХ ЧАСТИН ІНФОРМАЦІЙНИХ СИСТЕМ

Вінницький національний технічний університет

Анотація

Розглянуто можливості реалізації NoSQL баз даних у розробці серверного додатку. Досліджено переваги у використанні перед SQL базами даних. Головні вимоги та критерії до досліджуваної предметної області: швидкість розгортання, економія ресурсів, складність розробки, швидкість виконання, складність у підтримці.

Ключові слова: NoSQL, SQL, серверний додаток.

Abstract

The possibilities of implementation of NoSQL databases in server application development are considered. The advantages of using SQL databases are explored. The main requirements and criteria for the studied subject area: speed of deployment, cost savings, complexity of development, speed of implementation, complexity of support.

Keywords: NoSQL, SQL, server application.

База даних (англ. database) – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування (за стандартом ISO/IEC 2382:2015). В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки. [1]

Реляційна база даних — база даних, заснована на реляційній моделі даних. Слово «реляційний» походить від англ. relation. Для роботи з реляційними БД застосовують реляційні СКБД. Інакше кажучи, реляційна база даних — це база даних, яка сприймається користувачем як набір нормалізованих відношень різного ступеня.

Реляційна база даних є сукупністю елементів даних, організованих у вигляді набору формально описаних таблиць, з яких дані можуть бути доступними або повторно зібрані багатьма різними способами без необхідності реорганізації таблиць бази даних.Оскільки типові задачі математичного аналізу можуть потребувати візуалізації у вигляді складних графіків функцій, які можуть потребувати високої деталізації, то найоптимальнішою платформою для вирішення такого типу задач є персональний комп’ютер. Тому було розглянуто бібліотеки для платформи .NET. [2]

NoSQL (зазвичай розшифровується як англ. non SQL або англ. non relational, іноді англ. not only SQL) — база даних, яка забезпечує механізм зберігання та видобування даних відмінний від підходу таблиць-відношень в реляційних базах даних. Подібні бази даних існували вже в другій половині 1960-х років, але тоді вони ще не здобули гучне ім'я «NoSQL», одержане після сплеску популярності на початку 21-ого століття, що був спричинений потребами Web 2.0 компаній, такими як Facebook, Google, та Amazon.com. NoSQL бази даних все більше і більше використовуються в задачах із застосуванням великих даних та real-time web-застосунках. NoSQL системи також називають «Not only SQL» (англ. not only SQL — не тільки SQL) для підкреслення того, що вони можуть підтримувати SQL-подібну структуру та мову запитів. [3]

NoSQL бази даних своєю філософією та структурою нагадують будову звичної усім файлової системи (ФС) NTFS. У не реляційних БД подібно NTFS також можна прослідкувати ієрархічну будову проекту. Подібно директоріям и піддиректоріям NoSQL БД мають документи та підколекції (рядки та підтаблиці, в аналогії з SQL БД). Проте, не реляційні БД мають суттєву перевагу у вигляді вільного вмісту документу. В той час як у реляційних моделях БД рядок має чітко визначені стовпці (поля в аналогії з NoSQL).

Не реляційні БД поділяються за типами: «Ключ-значення: кеш», «Ключ-значення: сховище»,

«Сервер структурованих даних», «Кортеж: сховище», «Об'єктна база даних», «Документ: сховище» та «Широко-колонкове сховище». [3]

Для роботи з NoSQL базою даних є достатній вибір СКБД. Розглянемо найпопулярніші рішення у нереляційних БД типу «Документ: сховище».

MongoDB — документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

При розробці автори виходили з необхідності спеціалізації баз даних, завдяки чому їм вдалося відійти від принципу «один розмір під усе». За рахунок мінімізації семантики для роботи з транзакціями з'являється можливість вирішення цілого ряду проблем, пов'язаних з нестачею продуктивності, причому горизонтальне масштабування стає простішим. Використовувана модель документів зберігання даних (JSON/BSON) простіше кодується, простіше управляється (у тому числі за рахунок застосування так званого «безсхемного стилю» (англ. schemaless style), а внутрішнє угруповування релевантних даних забезпечує додатковий виграш в швидкодії. Нереляційний підхід досить зручний для створення баз даних, у яких горизонтальне масштабування означає розгортання на множині машин. Можливість забезпечувати найкращу продуктивність повинна існувати паралельно з підтримкою більшої функціональності, ніж це дозволяє використання пар «ключ-значення» (у чистому вигляді). Технологія баз даних має працювати скрізь, починаючи з серверів користувача та віртуальних машин і закінчуючи хмарними технологіями. [4]

Apache CouchDB (Cluster Of Unreliable Commodity Hardware) — розподілена документо-орієнтована система управління базами даних класу NoSQL-систем, що не вимагає опису схеми даних. Запити до CouchDB та індексація даних можуть виконуватися згідно з парадигмою MapReduce, використовуючи для формування логіки вибірки даних мову JavaScript.

CouchDB зберігає дані в форматі впорядкованого списку і дозволяє проводити часткову реплікацію даних між декількома БД в режимі «майстер-майстер» з одночасним виявленням і вирішенням конфліктних ситуацій. Кожен сервер зберігає свій локальний набір даних, синхронізований з іншими серверами, які можуть переводитися в offline-режим і періодично реплікувати зміни. Зокрема, така можливість робить CouchDB привабливим рішенням для організації синхронізації налаштувань програм між різними комп'ютерами (наприклад, це використовувалося для синхронізації вмісту адресної книги ПК с мобільним телефоном через сервіс Ubuntu One. [5]

Firebase — це платформи розробки мобільних та веб- застосунків. Firebase розвивається з 2011 року компанією Firebase Inc., яку придбав Google у 2014.

Firebase надає в режимі реального часу базу даних та бекенд як службу. Ця служба надає розробникам застосунків API, який дозволяє синхронізувати дані застосунків між клієнтами та зберігати їх у хмарі Firebase. Компанія також надає клієнтські бібліотеки, які дозволяють інтеграцію із застосунками Android, iOS, JavaScript / Node.js, Java, Objective-C, Swift. База даних також доступна через REST API та прив'язки до декількох сценаріїв JavaScript, таких як AngularJS, React, Ember.js та Backbone.js. REST API використовує протокол подій із сервером, який є інтерфейсом для створення HTTP-з'єднань для отримання push-повідомлень від сервера. Розробники, які використовують Realtime Database, можуть захищати свої дані за допомогою правил безпеки, що застосовуються на сервері. [6]

Дані СКБД широко використовуються у розробці як об’ємних, так і простих проектів. Повертаючись до безпосередньо переваг варто визначити ряд наступних фактів:

- Простота роботи. Багато NoSQL баз даних, в основному сховищі виду “ключ-значення” мають у порівнянні з реляційними базами даних дуже сильно урізану функціональність, яка їм просто не потрібна для виконання поставлених завдань. В такому випадку оператору бази даних не потрібно глибоких знань досить потужного і гнучкого механізму роботи з SQL-запитами. Це дуже сильно знижує вхідний поріг для початку роботи з NoSQL сховищами.
- Простіший синтаксис запитів — менше помилок. Для спрощення роботи з базою даних деякими розробниками використовується ORM (Object-Relational Mapping) (Object-Relational Mapping) — це технологія, що дозволяє автоматично транслювати операції з об’єктами в запити до бази даних. Найчастіше подібні рішення працюють неефективно і плодять безліч непотрібних або відверто помилкових запитів. Не можна сказати, що

розробники ORM погано виконують свою роботу, просто завдання занадто складне. Мова SQL універсальна, проте дуже ємна, для повноцінної роботи з нею необхідний певний багаж знань. При цьому власні мови запитів сучасних NoSQL сховищ набагато більше підходять для виконання простих маніпуляцій з базою даних.

- Реплікація — це копіювання даних при їх оновленні на інші сервера. Цей механізм дозволяє домогтися більшої відмовостійкості і масштабованості системи. Прийнято виділяти два види реплікації: master-slave і peer-to-peer. Перший тип має на увазі наявність одного майстер-сервера і кількох дочірніх серверів. Запис може проводитися тільки на майстер-сервер, а він в свою чергу передає зміни на дочірні машини. Цей тип реплікації дає хорошу масштабованість на читання (читання може відбуватися з будь-якого вузла мережі), але не дозволяє масштабувати операції записи. Запис йде тільки на один майстер-сервер. Також такий варіант організації реплікації передбачає складності в разі несправності майстер-сервера. В такому випадку повинен відбуватися автоматичний або ручний вибір нового майстер-сервера з решти. Другий тип peer-to-peer — припускає, що всі вузли рівні в можливості обслуговувати запити на читання і запис. Інформація про оновлення даних передається від сервера до сервера по колу.
- Шаринг — це поділ масиву інформації з різних вузлів мережі, коли кожен вузол відповідає тільки за певний набір даних і обробляє запити на читання і запис, що відносяться тільки до цього набору даних.

Різкий стрибок популярності NoSQL баз даних і пов'язані з ним історії використання нереляційних СУБД показали світу IT важливість реалістичної оцінки пріоритетів компанії. Деякі вендори успішно впровадили у себе NoSQL сховища і отримали помітне зниження витрат і підвищення якості їх додатків. Інші зазнали невдачі, пізно зрозумівши, що прийняте рішення їм не підходить. А треті просто залишилися зі своїми технологіями. Реляційні або нереляційні бази даних не єдиний вибір, який належить зробити компанії. Не менш важливим є і вибір між конкретними системами і конкретними стратегіями роботи з ними. [7]

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. База даних [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/База_даних.
2. Реляційна база даних [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Реляційна_база_даних.
3. NoSQL [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/NoSQL.
4. MongoDB [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/MongoDB.
5. Apache CouchDB [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Apache_CouchDB.
6. Firebase [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Firebase.
7. NOSQL — ПЕРЕВАГИ ТА НЕДОЛІКИ НЕРЕЛЯЦІЙНИХ БАЗ ДАНИХ [Електронний ресурс] – Режим доступу до ресурсу: https://www.quality-assurance-group.com/nosql-perevagy-ta-nedoliky-nerelyatsijnyh-baz-danyh/.

Веретко Андрій Юрійович — студент групи 1ПІ-18мс, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: veretkoandr@gmail.com.

Науковий керівник – ***Катєльніков Денис Іванович***, кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: fuzzy2dik@gmail.com.

Veretko Andrii Yuriyovich — student of Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: veretkoandr@gmail.com.

Supervisor– ***Katielnikov Denys Ivanovych***, PhD, Associate Professor of Software Engineering Department, Vinnytsia National Technical University, Vinnytsia, E-mail: fuzzy2dik@gmail.com.