

ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ВИКОНАННЯ SQL-ЗАПИТІВ

Вінницький національний технічний університет

Анотація

Проаналізовано основні причини зниження продуктивності виконання запитів до бази даних та проблеми, що виникають під час спроб користувачів оптимізувати запит власноруч.

Ключові слова: *SQL-запит, продуктивність, база даних, СКБД.*

Abstract

Analyzed the main reasons for the decrease in database query performance and problems witch occurred when users try to optimize query by himself.

Keywords: *SQL- query, performance, database, DBMS.*

Вступ

Згідно основних принципів програмування [1], будь-яка система повинна мати можливості масштабування та розширення, в результаті чого з'являється необхідність зберігати велику кількість даних. Одним з найпопулярніших методів зберігання великої кількості даних є бази даних.

База даних – 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 [2]. Для доступу до даних в БД використовуються системи керування баз даних, які створюють класифікацію архітектури баз даних: ієрархічна, мережна та реляційна. Для вибору необхідної архітектури необхідно оцінити які розміри даних будуть лежати в базі, якого типу транзакції будуть виконуватись та архітектура системи, що розробляється.

На сьогоднішній день, найбільш популярною архітектурою для бази даних ж реляційний підхід, згідно якого дані зберігаються у вигляді різних таблиць. Для доступу до даних використовують декларативну мову програмування SQL. У декларативних мовах програмування, користувач не описує детальний алгоритм виконання, а лише бажану відповідь, в результаті чого багато користувачів вважає, що вони не несуть відповідальності за час виконання запиту.

Отже, метою дослідження є підвищення продуктивності виконання SQL-запитів у середовищі MS SQL шляхом розробки та використання спеціалізованої програми, що дозволяє зменшити час виконання запитів.

Об'єкт дослідження – процес оптимізації SQL-запитів.

Предмет дослідження – алгоритми та засоби для аналізу продуктивності SQL-запитів.

Результати дослідження

Перш за все необхідно розглянути, що називають критичними запитами. Такі запити можна розділити на дві категорії за частотою та часом виконання [3]:

- Виконуються досить рідко, але їх тривалий час виконання помітно неозброєним поглядом.
- Часто виконуються запити, які самі по собі не займають багато часу, але через те, що виконуються часто, загальний час їх виконання стає критичним.

Тривале виконання запитів може створити досить серйозну проблему, особливо коли доступ до бази відбувається асинхронно. В такому випадку один неоптимізований запит може сповільнити роботу всієї системи через довготривале блокування таблиць, які використовуються у оптимізованих запитах [4]. Звісно для вирішення цієї проблеми можна просто використовувати вбудовані методи, як, наприклад, у середовищі MS SQL таблична підказка WITH(NOLOCK) [5], але це призведе до проблеми брудного читання.

Для доступу до даних всередині реалізації СКБД використовують індекси. Індекс є структурою на диску, яка пов'язана з таблицею і прискорює отримання рядків з таблиці. Індекс містить ключі, побудовані з одного або декількох стовпців в таблиці [6]. Ці ключі зберігаються у вигляді структури збалансованого дерева, яка підтримує швидкий пошук рядків по їх ключовим значенням.

Індекси поділяють на кластерні (автоматично створюється для кожної таблиці) та некластерні (створюються лише за вимоги користувача). Створювати некластерний індекс потрібно лише за потреби по полю, що забезпечить мінімальну щільність розподілу. В іншому випадку, після кожної операції, що змінює дані в таблиці, буде відбуватись перебудова всіх індексів пов'язаних з цією таблицею, що призведе до значної затримки під час виконання даних операцій.

Ще одним інструментом, який може сповільнити запит є табличні підказки. Перед виконанням запиту вбудований механізм створює план виконання і вибір фізичних операторів. Однак користувач може власноруч задати бажаний йому оператор, зазвичай це роблять при створенні з'єднань.

Однак практика показує, що табличні підказки частіше призводять до збільшення часу виконання запитів [7]. Насамперед, це пов'язане з тим, що користувач, який бажає використання «кращих» операторів не звертає уваги на причину, за якою SQL SERVER не включив цей оператор у план виконання. Наприклад, багато користувачів знає, що найшвидшим оператором з'єднання є «MERGE JOIN», однак для його використання необхідно, щоб обидві таблиці були відсортовані за полем, по якому відбувається з'єднання, тому при використанні табличної підказки для використання цього оператора, може відбутись сортування обох таблиць. В результаті ми отримуємо запит в декілька разів повільніший за аналогічний з використання неефективного на перший погляд оператора.

Для допомоги користувачам уникнути можливих проблем з оптимізацією запитів, розроблено програмний додаток для дослідження продуктивності SQL-запитів [8]. Основним функціоналом додатка є аналіз та оцінка вхідного запита та плану виконання в середовищі MS SQL.

Висновок

В результаті дослідження було виявлено найпоширеніші помилки, що допускають користувачі, під час спроб оптимізувати запит. Розроблено додаток для дослідження продуктивності SQL-запитів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. S.O.L.I.D: The First 5 Principles of Object Oriented Design [Електронний ресурс] – Режим доступу: <https://scotch.io/bar-talk/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>
2. ISO/IEC 2382:2015, Information technology — Vocabulary — Part 1: Terms and definitions
3. Как думать на SQL? [Електронний ресурс] – Режим доступу: <https://habr.com/ru/post/305926/>
4. Кучерявий І.В. Оптимізація запитів до баз даних [Електронний ресур]/ І.В.Кучерявий, А.І.Веренько О.В.Романюк // Матеріали XLIX Науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії (2020). – Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2020/paper/view/9493/7929>
5. Hints (Transact-SQL) – Table [Електронний ресурс] – Режим доступу: <https://docs.microsoft.com/en-us/sql/t-sql/queries/hints-transact-sql-table?view=sql-server-ver15>
6. Clustered and Nonclustered Indexes Described [Електронний ресурс] – Режим доступу: <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described?view=sql-server-ver15>
7. Index Hints: Helpful or Harmful? [Електронний ресурс] – Режим доступу: <https://www.brentozar.com/archive/2013/10/index-hints-helpful-or-harmful/>
8. Свіжак В.В. Програмний додаток для дослідження продуктивності SQL-запитів [Електронний ресур]/ В.В.Свіжак, О.В.Романюк // Матеріали XLIX Науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії (2020). – Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2020/paper/view/9110/7918>

Свіжак Віктор Вячеславович, студент групи ІПІ-16б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: viktor.svizhak@gmail.com

Романюк Оксана Володимирівна, к.т.н., доцент кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця, e-mail: romaniukoksanav@gmail.com

Viktor Svizhak, student of group ІПІ-16b, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: viktor.svizhak@gmail.com

Oksana Romaniuk – Candidate of Technical Sciences, Associate Professor of the Software Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: romaniukoksanav@gmail.com