

ВЛИЯНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ НА ПРОИЗВОДИТЕЛЬНОСТЬ SQL БАЗЫ ДАННЫХ

Невейков А.С.

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Научный руководитель: Тонкович И.Н. – к.х.н., доцент, доцент кафедры ПИКС

Аннотация. Экспериментально исследовано влияние использования параллельного выполнения запросов на производительность базы данных. Установлено, что использование нескольких потоков при выполнении команд может увеличить скорость выполнения запроса в четыре раза. Предложен способ ускорения выполнения запросов к базе данных с использованием многопоточности.

Ключевые слова: многопоточность, база данных, SQL-запросы, производительность

Введение. В настоящее время с увеличением объемов данных и требований к скорости обработки информации важность эффективного управления базами данных становится все более актуальной. Одним из ключевых факторов, влияющих на производительность баз данных, является способность обрабатывать запросы параллельно. В связи с этим возникает необходимость исследования потенциальных методов оптимизации выполнения запросов для улучшения производительности систем управления базами данных.

В данном исследовании автором показано, как зависит производительность реляционной СУБД Oracle DB от количества обрабатываемых данных при использовании нескольких потоков для выполнения запроса.

Основная часть. При оценке влияния многопоточной обработки на скорость выполнения запроса необходимо решить следующие две задачи:

- обеспечить достаточную полноту тестирования;
- определить одинаковые условия для каждого эксперимента.

В качестве примера рассматривалась реляционная СУБД Oracle DB версии 18с. При проведении экспериментов использовалась выборка на 1 000 000 записей. Поскольку при выполнении запросов СУБД Oracle DB подгружает в кэш часто выполняемые операции, то после каждой операции таблица удалялась и пересоздавалась заново.

Результаты проведенных исследований представлены в таблице 1.

Таблица 1– Результаты проведенных экспериментов

№ эксперимента	Описание эксперимента	Среднее время выполнения запроса (секунд)
1	2	3
Э1	Получение тестовой выборки с условием и сортировкой с использованием одного потока	0.296
Э2	Получение тестовой выборки с условием и сортировкой с использованием четырех потоков	0.311
Э3	Получение тестовой выборки только оператором SELECT с использованием одного потока	0.038
Э4	Получение тестовой выборки только оператором SELECT с использованием четырех потоков	0.054
Э5	Удаление данных из тестовой выборки с использованием одного потока	1.762
Э6	Удаление данных из тестовой выборки с использованием четырех потоков	1.862
Э7	Генерация тестовой выборки с использованием одного потока	23.409

Продолжение таблицы 1

Э8	Генерация тестовой выборки с использованием четырех потоков	5.511
Э9	Генерация тестовой выборки с использованием двух потоков	5.364
Э10	Генерация тестовой выборки с использованием шестидесяти четырех потоков	7.084
Э11	Генерация тестовой выборки с использованием двухсот пятидесяти шести потоков	5.962
Э12	Генерация тестовой выборки с использованием двух тысяч сорока восьми потоков	5.959
Э13	Удаление таблицы с использованием четырех потоков	0.015
Э14	Удаление таблицы с использованием одного потока	0.018

Как видно из таблицы, использование многопоточной обработки при выполнении SQL-запросов эффективнее для больших объемов данных.

Результаты обусловлены тем, что использование нескольких потоков позволяет выполнить несколько операций одновременно, что особенно важно при выполнении операций, требующих интенсивную обработку данных. Это позволяет распределить нагрузку на процессор и использовать вычислительные ресурсы более оптимально. Также в многопоточной среде возможна параллельная загрузка данных из памяти, кэширование и другие оптимизации, что способствует более эффективному использованию ресурсов. Нелинейное изменение времени в экспериментах Э8-Э12 обусловлено наличием в Oracle DB планировщика задач, который просчитывает неэффективность создания большого числа потоков и при попытке создать 256 или 2048 находит оптимальное значение.

На выборке в 1 млн записей использование многопоточности увеличило производительность на 4,9% при запросе выборки с условием и сортировкой в рамках одного потока; на 29.7% – при запросе выборки только оператором SELECT; на 5.4% – при удалении данных из тестовой выборки, в 4.24 раза – при генерации тестовой выборки. Обнаружена нелинейная зависимость между количеством потоков и скоростью выполнения запросов.

Заключение. Выполнен анализ воздействия многопоточной обработки запросов на быстродействие базы данных. Установлено, что прирост производительности составляет 4.9%. Определено нелинейное изменение времени при попытке создать 256 или 2048 потоков, что обусловлено наличием в Oracle DB планировщика задач, защищающим базу данных от неэффективного использования ресурсов.

Предложено использование многопоточной обработки запросов для СУБД Oracle DB. Найден наиболее оптимизируемый тип запросов – генерация данных.

Список литературы

1. Документация OracleDB 18c – [Электронный ресурс]. Режим доступа: <https://docs.oracle.com/en/database/oracle/oracle-database/18/>.

UDC 621.3.049.77–048.24:537.2

PARALLEL COMPUTATIONS IMPACT ON SQL DATABASE PERFORMANCE

Niaveikau A.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Tonkavich I.N. – Cand. of Che., associate professor, associate professor of the department of ICSD

Annotation. The influence of parallel query execution on database performance is experimentally investigated. It has been determined that utilizing multiple threads during command execution can increase query execution speed by four times. A method for accelerating query execution in a database using multithreading is proposed.

Keywords: multithreading, databases, SQL-queries, performance