DOI 10.37882/2223-2966.2025.05-2.07

ОЦЕНКА ВРЕМЕНИ ВЫПОЛНЕНИЯ ЗАПРОСОВ ИЗМЕНЕНИЯ И УДАЛЕНИЯ В NOSQL И ОБЪЕКТНО-РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ

ESTIMATION OF EXECUTION TIME FOR MODIFICATION AND DELETION QUERIES IN NOSQL AND OBJECT-RELATIONAL DATABASE

B. Goryachkin
A. Alekseev

Summary. *Problem statement*. To store different types of data, different databases are used, operations to which are performed at different speeds. It is necessary to optimize the execution time of user queries to different databases.
Goal. To increase the efficiency of user interaction with different types of databases.
Results. The time of operations for changing and deleting data from different types of databases is considered, depending on the amount of data, as well as the type of data.
Practical significance. The data was obtained and recommendations for the use of databases were developed based on them, depending on their number and types. This will allow you to select the most efficient databases and distribute data across them on a case-by-case basis.

Keywords: lots of databases, PostgreSQL, Cassandra, MongoDB, Neo4j.

Горячкин Борис Сергеевич
кандидат технических наук, доцент, Московский государственный технический университет им. Н.Э. Баумана
bsgor@mail.ru
Алексеев Андрей Сергеевич
Московский государственный технический университет им. Н.Э. Баумана
asalexeev2001@mail.ru

Аннотация. *Постановка проблемы*. Для хранения разнотипных данных используются различные базы данных, операции к которым выполняются с разной скоростью. Необходимо оптимизировать время выполнения запросов пользователя к разным базам данных.
Цель. Повысить эффективность взаимодействия пользователя с разнотипными базами данных.
Результаты. Рассмотрено время проведения операций изменения и удаления данных из разнотипных баз данных в зависимости от количества данных, а также типа данных.
Практическая значимость. Получены данные и разработаны на их основе рекомендации по использованию баз данных в зависимости от их количества и типов. Это позволит выбрать наиболее эффективные базы данных и распределение данных по ним в каждом конкретном случае.

Ключевые слова: множество баз данных, PostgreSQL, Cassandra, MongoDB, Neo4j.

Введение

В настоящее время наблюдается продолжающийся рост систем, поддерживающих огромный объем реляционных и нереляционных форм данных. Примерами моделей данных, которые поддерживают многомодельные базы данных, являются документные, графические, реляционные и колоночные модели. Эти модели используются для решения различных задач и оптимизации работы с данными в зависимости от их структуры, объема и требований к обработке. Каждая модель данных имеет свои преимущества и подходит для конкретных сценариев использования. Поэтому необходимо знать в каких базах данных наиболее быстрые запросы для определённых типов данных.

Постановка задачи

В рамках данного исследования будет осуществлен анализ NoSQL и объектно-реляционных систем управления базами данных, каждая из которых имеет свои уникальные достоинства и недостатки [1]. Главная цель исследования заключается в выявлении наиболее эффективной системы управления базами данных через сравнительный анализ времени выполнения запросов на изменение и удаление, в графовой (Neo4J), документной (MongoDB), объектно-реляционной (PostgreSQL) и колоночной (Cassandra) системах для кластеров баз данных [2], [3], а также сравнительный анализ времени выполнения запросов при изменении количества столбцов (PostgreSQL), полей (MongoDB), колонок (Cassandra), атрибуты (Neo4J) и их типов данных.

Практический эксперимент проводится на операционной системе Ubuntu. Поскольку СУБД выбирается не под какую-то конкретную тему, то для эксперимента была создана БД, содержащая информацию о блюдах, пользователях и их заказах.

Описание работы БД

База данных (БД) — это структурированная система для хранения, организации и управления данными.

В контексте аналитики БД используется для хранения больших объемов данных, которые затем анализируются для извлечения полезной информации, принятия решений и прогнозирования.

Основные компоненты БД:

1. **Таблицы**: Основные структуры для хранения данных. Данные организованы в строки (записи) и столбцы (поля).
2. **Индексы**: Структуры, ускоряющие поиск данных в таблицах.
3. **Запросы**: Команды на языке SQL (или другом языке запросов), которые извлекают, изменяют или удаляют данные.
4. **Транзакции**: Группы операций, которые выполняются как единое целое (например, вставка, обновление, удаление).
5. **Репликация и шардирование**: Механизмы для масштабирования и повышения отказоустойчивости БД.

Аналитика на основе БД:

1. **Агрегация данных**: Подсчет сумм, средних значений, максимумов, минимумов и других метрик.
2. **Фильтрация данных**: Выборка данных по определенным критериям (например, временной интервал, категория).
3. **Группировка данных**: Анализ данных по группам (например, по пользователям, регионам).
4. **Визуализация данных**: Представление данных в виде графиков, диаграмм и отчетов.

Модельное время выполнения запросов

Время выполнения запросов зависит от множества факторов, включая объем данных, сложность запроса, производительность сервера БД, наличие индексов и оптимизацию запросов. Ниже приведены примеры модельного времени выполнения для различных типов запросов.

1. Простой SELECT-запрос (поиск по индексу)

Пример запроса: SELECT * FROM users WHERE id = 12345;

Время выполнения: O(log n), где n — количество записей в таблице.

2. SELECT-запрос с фильтрацией (без индекса)

Пример запроса: SELECT * FROM users WHERE age > 30;

Время выполнения: O(n), где n — количество записей в таблице.

3. Агрегация данных (GROUP BY)

Пример запроса: SELECT country, COUNT(*) FROM users GROUP BY country;

Время выполнения: O(n), где n — количество записей в таблице.

4. JOIN-запрос (соединение таблиц)

Пример запроса: SELECT u.name, o.order_id FROM users u JOIN orders o ON u.id = o.user_id;

Время выполнения: O(n * m), где n и m — количество записей в таблицах users и orders.

5. Сложный аналитический запрос (оконные функции)

Пример запроса: SELECT user_id, SUM(amount) OVER (PARTITION BY user_id ORDER BY date) AS running_total FROM transactions;

Время выполнения: O(n log n), где n — количество записей в таблице.

Факторы, влияющие на время выполнения запросов

- Объем данных: Чем больше данных, тем дольше выполняется запрос.
- Индексы: Наличие индексов значительно ускоряет поиск и фильтрацию данных.
- Сложность запроса: Запросы с JOIN, GROUP BY, оконными функциями выполняются дольше.
- Производительность сервера: Скорость CPU, объем оперативной памяти и тип хранилища (HDD/SSD) влияют на время выполнения.
- Параллелизм: Возможность выполнения запросов в параллельном режиме (например, в распределенных БД).
- Кэширование: Кэширование результатов запросов может значительно ускорить повторное выполнение.

Определение аналитической зависимости времени выполнения запроса от его параметров

Учитывая, что в практических системах баз данных имеется как минимум несколько тысяч записей, вычисления аналитических зависимостей времени выполнения запросов от числа обрабатываемых строк будут проводиться при условии, что количество строк превышает 1000. Кроме того, в реальных системах запросы к базам данных обрабатываются на нескольких серверах, что требует учета времени распределения и передачи к серверам.

Время выполнения запроса к базе данных $\tau_{запр}$ рассчитывается по следующей формуле (1):

$$\tau_{запр} = \tau_{обращ} + \tau_{расп} + \tau_{выполн}, \quad (1)$$

где $\tau_{обращ}$ — это время, которое требуется для передачи запроса к базе данных и получения ответа от нее. Оно зависит от скорости сетевого соединения между приложением и базой данных; $\tau_{расп}$ — это время, которое требуется для распределения запроса между серверами; $\tau_{выполн}$ — это время, затраченное на выполнение самого запроса в сервере базы данных. Оно зависит от сложности запроса, объема данных и индексации.

Так как передача запроса и получение ответа происходят в одной и той же среде время передачи запроса и время получения ответа будут равны. Таким образом $\tau_{обращ}$ можно представить, как формулу 2:

$$\tau_{обращ} = 2 * \tau_{пути}, \quad (2)$$

где $\tau_{обращ}$ — это время пути запроса к базе данных.

Время выполнения запроса в сервере базы данных ($\tau_{выполн}$) равен сумме параметров, указанных в формуле 3:

$$\tau_{выполн} = \tau_{диск} + \tau_{индекс} + \tau_{СУБД}, \quad (3)$$

где $\tau_{диск}$ — время чтения с диска в оперативную память. Оно зависит от количества записей в таблице; $\tau_{индекс}$ — время выполнения, которое экономит использование индексов при запросе к серверу [4], [5]. Оно зависит от разных типов индексов; $\tau_{СУБД}$ — время стартовой задержки выполнения запроса сервером, которое зависит от категории СУБД и особенностей моделей данных.

На Рисунке 1 показана обобщенная схема выполнения запроса.

При изменении и удалении данных через API приложение взаимодействует с системой управления базами данных (СУБД), где координатор распределяет запросы между серверами [6]. СУБД передает запрос к базе данных, которая, в процессе чтения данных, обращается к дисковому накопителю, а затем, при наличии данных в кэш-памяти, использует ее. В случае выполнения операции записи обращение осуществляется к оперативной памяти с последующим просмотром необходимой информации в кэш-памяти.

Определим время выполнения запроса, основываясь на построении непрерывной функции с учетом промежуточных значений, используя формулы (1), (2) и (3). В случае значительного расхождения между экспериментальным и теоретическим временем выполнения запроса будет применен поправочный коэффициент.

Представим формулы, которые описывают процесс определения времени выполнения запроса для всех систем управления базами данных. Для запроса 1 (изменение данных) зависимость можно выразить формулами (4) — (7), где N обозначает количество строк в таблице.

В результате проведённых экспериментов было зафиксировано время передачи запроса от приложения к базе данных, а также время возвращения запроса от базы данных к приложению ($\tau_{пути}$) — 0,6 мс. Кроме того, было установлено время, затраченное координатором на распределение запроса по конкретному серверу ($\tau_{расп}$) — 0,5 мс.

Время чтения данных с диска в оперативную память ($\tau_{диск}$), представленное в формулах (4) — (11), описывается многочленом, который зависит от числа записей в таблице. Время обработки запроса сервером ($\tau_{индекс}$) в данном эксперименте считается равным нулю, поскольку индексация в базе данных не проводилась. В противном случае это время могло бы оказаться отрицательным, так как наличие индексов сокращает время выполнения запроса.

В связи с тем, что в аналитической зависимости имеется определенная погрешность, в параметре $\tau_{СУБД}$ введен корректирующий коэффициент, который уменьшает расхождение между $\tau_{ЭКСП}$ и $\tau_{запр}$.

Ниже представлены аналитические зависимости запроса на изменение для случаев, когда количество строк превышает 1000, количество строк не увеличено и типы данных строк не изменены (формулы (4) — (7)), основанные на формулах (1), (2) и (3), предназначенные для оценки эргономичности времени выполнения запроса. Формулы были выведены с применением метода регрессии для приближения функции с одной переменной [7] [8].

Методика проведения экспериментальных исследований производительности БД

При высокой нагрузке на один сервер целесообразно распределить её между несколькими серверами, организовав кластер. Применение кластерных систем управления базами данных и развертывание кластера баз данных на нескольких серверах позволяют снизить нагрузку на отдельные серверы и ускорить время обработки запросов [9].

В ходе практического эксперимента мы соберем данные для определения зависимости времени выполнения запроса от размера базы данных, процента записей, соответствующих заданному условию, количества полей, включенных в запрос, категории системы управления базами данных, количества столбцов в таблицах, типов данных столбцов, а также особенностей модели данных.

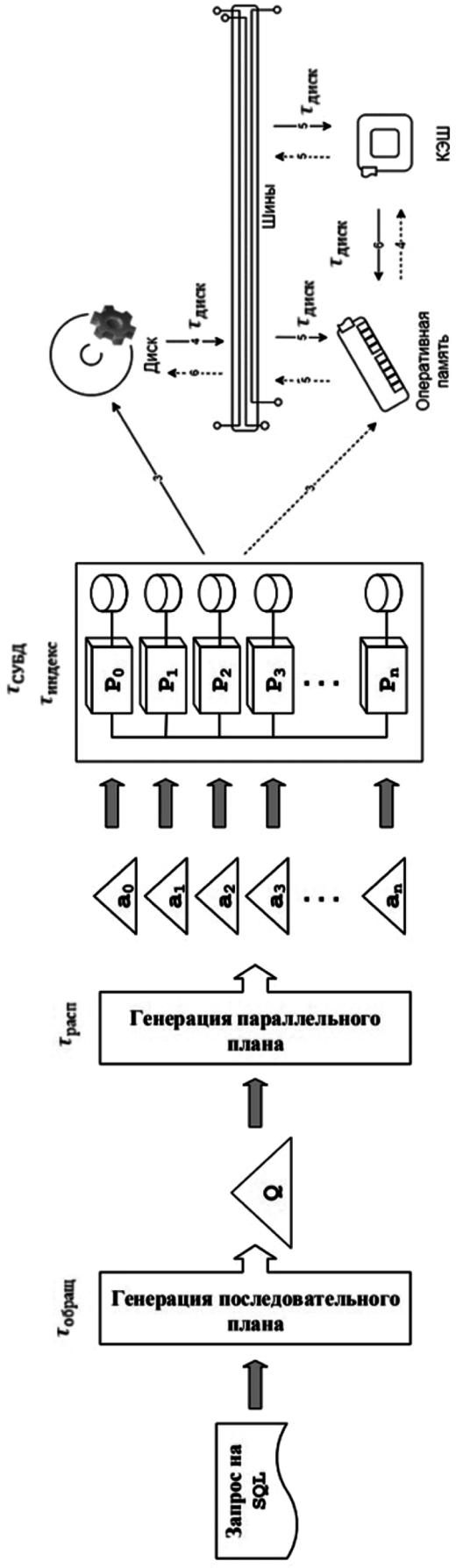

Рис. 1. Упрощенная схема выполнения запроса

В базе данных для проведения эксперимента находятся сущности User, Order и Dish, содержащие от 1 до 100000000 записей. Запросы включали операции изменения и удаления данных в каждой из трёх сущностей с целью анализа времени их выполнения. При проведении экспериментов с большими объемами данных (более 1000 строк) без изменения структуры базы данных наиболее производительна СУБД Cassandra. Худшее время показывает СУБД PostgreSQL. Такие же эксперименты были проведены для запросов на изменение и удаление к таблицам, где были изменены количество столбцов или типы данных. В случаях увеличения количества столбцов и изменения их типа на integer соотношение эффективности обработки запросов между СУБД не изменяются. В случае изменения типа столбцов на character varying СУБД Cassandra становится худшей, а СУБД Neo4J лучшей для выполнения обоих запросов.

Сравнение экспериментального и расчетного времени

После сравнения полученного времени выполнения запросов изменения и удаления для четырех СУБД при числе строк более 1000 было вычислено, что максимальная средняя ошибка аппроксимации в 16 % и 11 %. Эта ошибка является приемлемой. При анализе максимального объема данных, составляющего 100000000 строк, это означает, что 16 % и 12 % от минимального времени выполнения запроса составят соответственно 3,840 мс и 1578,77 мс. Эти значения превышают время, необходимое для зрительного восприятия человека (0,9–0,95 с). Однако, принимая во внимание латентный период реакции человека, мы считаем такие показатели удовлетворительными [10].

Таким образом, результаты, полученные на основе аналитической зависимости для запросов 1 и 2 при количестве строк свыше 1000, являются приемлемыми. Непрерывные функции, представленные в формулах (4) — (11), позволяют оценить время выполнения запросов на изменение и удаление данных с максимальной погрешностью в 16 %.

Заключение

В рамках исследования были проведены эксперименты для оценки времени выполнения запросов на изменение и удаление данных в четырех различных системах управления базами данных (СУБД). На основе полученных результатов разработаны рекомендации по выбору наиболее эффективной СУБД для работы с аналогичными базами данных. Установлено, что как в случае операций изменения, так и при удалении данных документная СУБД Cassandra демонстрирует наименьшее время выполнения запросов.

В данной работе также представлены формулы, описывающие аналитические зависимости времени выполнения запросов в зависимости от типа системы управления базами данных (СУБД) и объема данных в таблицах. Установлено, что результаты расчетов в значительной степени соответствуют экспериментально полученным данным. Авторы предлагают в качестве направлений для будущих исследований более детальный анализ аналитической зависимости для колоночных СУБД, а также дополнительные исследования, касающиеся использования индексов в запросах.

ЛИТЕРАТУРА

1. Виноградова М.В. «Постреляционные базы данных» Подготовительный материал. МГТУ им. Н.Э. Баумана, 2023.
2. Григорьев Ю.А. Оценка времени выполнения SQL-запросов к базам данных // Машиностроение и компьютерные технологии. 2012. №01.
3. Григорьев Ю.А., Ермаков Е.Ю. Анализ времени выполнения запроса в параллельном колоночном хранилище данных // Инженерный журнал: наука и инновации. 2013. №11 (23).
4. Виноградова М.В., Барашкова Е.С., Березин И.С., Ореликов М.Г., Лузин Д.С. Обзор системы полнотекстового поиска в постреляционной базе данных PostgreSQL // E-Scio. 2020. № 5 (44).
5. Морозов С.В., Нестеров С.А. Сравнительный анализ типов индексов в СУБД SQL Server И Postgresql. // Системный анализ в проектировании и управлении. 2024.
6. Плужников В.Л. Оценка времени выполнения запросов в параллельной системе баз данных // Машиностроение и компьютерные технологии. 2011. №06.
7. Бутырский Е.Ю., Кувалдин И.А., Чалкин В.П. Аппроксимация многомерных функций // НАУЧНОЕ ПРИБОРОСТРОЕНИЕ, 2010, том 20, № 2, с. 82–92
8. Онлайн калькуляторы PLANETCALC. URL: https://planetcalc.ru/5992/? (дата обращения: 10.12.2024).
9. Григорьев Ю.А., Плужников В.Л. Алгоритм выбора архитектуры параллельной системы баз данных по критерию стоимости // Машиностроение и компьютерные технологии. 2011. №12.
10. Горячкин Б.С. Эргономический анализ систем обработки информации и управления // Вестник евразийской науки. 2017. Т. 9. №3. С. 72.

Журнал «Современная наука: актуальные проблемы теории и практики»