



ОТКРЫТАЯ НАУКА
издательство

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.65.52

РАЦИОНАЛЬНОЕ ИСПОЛЬЗОВАНИЕ НЕСКОЛЬКИХ ТИПОВ БАЗ ДАННЫХ В СЛОЖНЫХ ПРОЕКТАХ

¹ Буркалов А.М., Демонов Л.А.

ФГБОУ ВО "ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ, Иркутск, Россия (664074, Иркутская область, город Иркутск, ул. Лермонтова, д. 83), e-mail: ¹artiom.burkalov@mail.ru

В данной работе проводится сравнительный анализ производительности, масштабируемости, гибкости и совместимости SQL и NoSQL баз данных в контексте управления проектами и задачами. Описаны тестовые сценарии для вставки, обновления, удаления и поиска данных, создания зависимостей между задачами, хранения комментариев и файлов, а также выполнения аналитических запросов и генерации отчетов. Исследование включает измерение времени выполнения операций, что позволяет выявить преимущества и недостатки каждого типа баз данных для различных задач.

Ключевые слова: Производительность, масштабируемость, SQL, NoSQL, управление задачами, тестовые сценарии, аналитические запросы, время выполнения операций.

RATIONAL USE OF SEVERAL TYPES OF DATABASES IN COMPLEX PROJECTS

¹ Bukalov A.M., Demonov L.A.

IRKUTSK NATIONAL RESEARCH TECHNICAL UNIVERSITY, Irkutsk, Russia (664074, Irkutsk Lermontova st., 83), e-mail: ¹artiom.burkalov@mail.ru

This paper provides a comparative analysis of the performance, scalability, flexibility and compatibility of SQL and NoSQL databases in the context of project and task management. Test scenarios for inserting, updating, deleting and searching data, creating dependencies between tasks, storing comments and files, as well as performing analytical queries and generating reports are described. The study includes measuring the execution time of operations, which allows you to identify the advantages and disadvantages of each type of database for different tasks.

Keywords: Performance, scalability, SQL, NoSQL, task management, test scenarios, analytical queries, operation execution time.

Базы данных SQL и NoSQL уже долгое время являются объектом активного научного исследования. Реляционные базы данных, такие как PostgreSQL и Oracle, были разработаны на основе принципов, изложенных в оригинальной статье Эдгара Ф. Кодда, который считается основоположником теории реляционных баз данных. Его работа заложила основы для понимания структурированных данных и реляционных моделей, что стало основой для современных СУБД. Исследования производительности и масштабируемости SQL и NoSQL баз данных выявили ключевые различия между ними. В работе «Predictive Performance Comparison Analysis of Relational & NoSQL Graph Databases» анализируются производительность и прогнозируемость различных типов баз данных, показывая, что NoSQL базы данных часто превосходят реляционные в условиях работы с большими объемами данных и при выполнении сложных запросов. [1] Другие исследования, такие как статья «А

Comparative Study of SQL and NoSQL Databases for Big Data», опубликованная в журнале MDPI, подробно рассматривают преимущества и недостатки SQL и NoSQL баз данных в контексте больших данных. Эти работы подчеркивают, что NoSQL базы данных, такие как MongoDB и Neo4j, предлагают значительные преимущества в гибкости и масштабируемости по сравнению с традиционными реляционными базами данных. В книге, опубликованной на платформе Springer, также рассматриваются различные аспекты использования SQL и NoSQL баз данных в сложных проектах. Эти исследования показывают, что графовые базы данных, такие как OrientDB, эффективны для моделирования сложных зависимостей между данными, что делает их полезными для специфических приложений, таких как социальные сети и системы рекомендаций. [2]

Цель настоящего исследования — изучить производительность и эффективность различных типов баз данных (PostgreSQL, OrientDB, MongoDB и Rebis) в контексте решения одной сложной задачи, разделенной на подзадачи с различными требованиями. Это позволит определить оптимальные комбинации баз данных для обеспечения максимально эффективной работы системы. В отличие от существующих исследований, которые фокусируются на сравнении производительности и масштабируемости баз данных в общем контексте, данное исследование сосредоточено на оптимизации использования различных типов баз данных для одной комплексной задачи. [3] Разбиение задачи на подзадачи и детерминированный подход к выбору баз данных позволят выявить наиболее эффективные комбинации для каждого конкретного сценария, обеспечивая максимальную производительность и эффективность работы системы в целом.

В данной работе все тесты проводились с использованием следующих баз данных: PostgreSQL, MongoDB, Rebis и OrientDB. Этот выбор обусловлен различием их внутренней структуры, что позволяет всесторонне изучить преимущества и недостатки различных типов баз данных в контексте выполнения сложных задач.

PostgreSQL — это мощная реляционная база данных с открытым исходным кодом, основанная на принципах, заложенных в оригинальной статье Эдгара Ф. Кодда. PostgreSQL известна своей надежностью, масштабируемостью и обширной функциональностью. Она поддерживает сложные запросы, транзакции и строгую целостность данных, что делает её идеальной для приложений, требующих высокой степени структурированности и надежности данных. [4]

MongoDB — это документно-ориентированная NoSQL база данных, которая использует формат BSON для хранения данных. MongoDB обеспечивает гибкость и масштабируемость, что делает её подходящей для работы с большими объёмами данных и разнообразными структурами данных. Она широко используется в проектах, где требуется быстрая обработка данных и возможность горизонтального масштабирования.

Rebis — это перспективная база данных, которая также относится к категории NoSQL. Rebis разработана для эффективного управления большими объёмами данных и поддерживает гибкую схему данных. Её выбор для исследования основан на необходимости изучения производительности и возможностей менее распространённых NoSQL баз данных в сравнении с более популярными решениями.

OrientDB — это многофункциональная NoSQL база данных, сочетающая в себе возможности документно-ориентированной и графовой базы данных. Она поддерживает работу с графами, что делает её особенно полезной для моделирования сложных зависимостей между данными. OrientDB позволяет эффективно управлять данными, связанными через множество взаимосвязей, что актуально для задач, связанных с социальными сетями, рекомендационными системами и управлением проектами.

Оборудование

Тесты проводились на следующем оборудовании: процессор Intel® Core™ i5-11400H 11-го поколения с тактовой частотой 2.70 GHz, 16 ГБ оперативной памяти, графический процессор RTX 3050 TI для ноутбуков и хранилище на основе NVMe SSD. [5] Тестовая среда была настроена с использованием Docker на операционной системе Linux. Скрипты тестирования были написаны на языке программирования Python, а для визуализации и анализа метрик использовалась библиотека Matplotlib.

Методология

Эффективное использование баз данных с различной внутренней структурой демонстрировалось на примере сложной задачи, разбитой на четыре подзадачи. В качестве примера была выбрана задача создания системы управления проектами. Эта система предназначена для организации и координации работы проектных команд, а также для обеспечения прозрачности и эффективности процессов. Исследование охватывает управление пользователями и проектами, граф зависимости задач, хранение комментариев и файлов, а также аналитику и отчетность.

Конкретизация тестов и структуры данных

В рамках управления пользователями и проектами были протестированы операции вставки, обновления, удаления и поиска данных. Структура данных включает пользователей с полями `id`, `name` и `email`, а также проекты с полями `id`, `name` и `description`. В процессе тестирования производилась вставка 10 000 записей пользователей и 1 000 проектов. Также были проведены тесты на обновление 20% существующих записей и удаление 10% данных. Для проверки функциональности поиска осуществлялся поиск пользователей по имени и проектов по ID. Метрики включают время выполнения запросов на вставку, обновление и удаление данных, а также время поиска по ключевым полям. Для анализа графа зависимости задач структура данных включает задачи с полями `id`, `name`, `description` и `status`, а также зависимости, представленные полями `task_id` и `depends_on_task_id`. Тесты охватывали создание 10 000 зависимостей между задачами, обновление 20% существующих зависимостей и поиск всех зависимостей для одной задачи. Основные метрики включают время выполнения запросов на создание и обновление зависимостей, а также время выполнения поиска зависимостей. В тестировании хранения комментариев и файлов рассматривается структура данных, включающая комментарии с полями `id`, `task_id`, `user_id` и `text`, а также файлы с полями `id`, `task_id`, `file_name` и `file_data`. В тестах проводилась вставка 50 000 комментариев и 5 000 файлов различных типов и размеров. [6] Дополнительно были выполнены обновления 20% комментариев и файлов и их удаление на 10%. Поиск данных включал поиск комментариев

по ключевым словам и файлов по имени. Временные метрики включают время выполнения запросов на вставку, обновление и удаление комментариев и файлов, а также время поиска по ключевым словам и имени файлов. Аналитика и отчетность включают выполнение различных аналитических запросов, таких как подсчет количества задач по статусу, анализ активности пользователей и зависимостей задач, а также генерацию отчетов по различным метрикам. Метрики в этом разделе фиксируют время выполнения аналитических запросов и генерацию отчетов, позволяя оценить эффективность обработки данных и генерации отчетных материалов.

Настройка тестовой среды

Для обеспечения объективности тестирования было использовано идентичное оборудование для каждой базы данных, чтобы исключить влияние аппаратных различий. Конфигурация тестовых сред была унифицирована, а тесты проводились в одинаковых условиях, что позволило получить сопоставимые результаты и корректные данные для анализа.

Результаты тестов

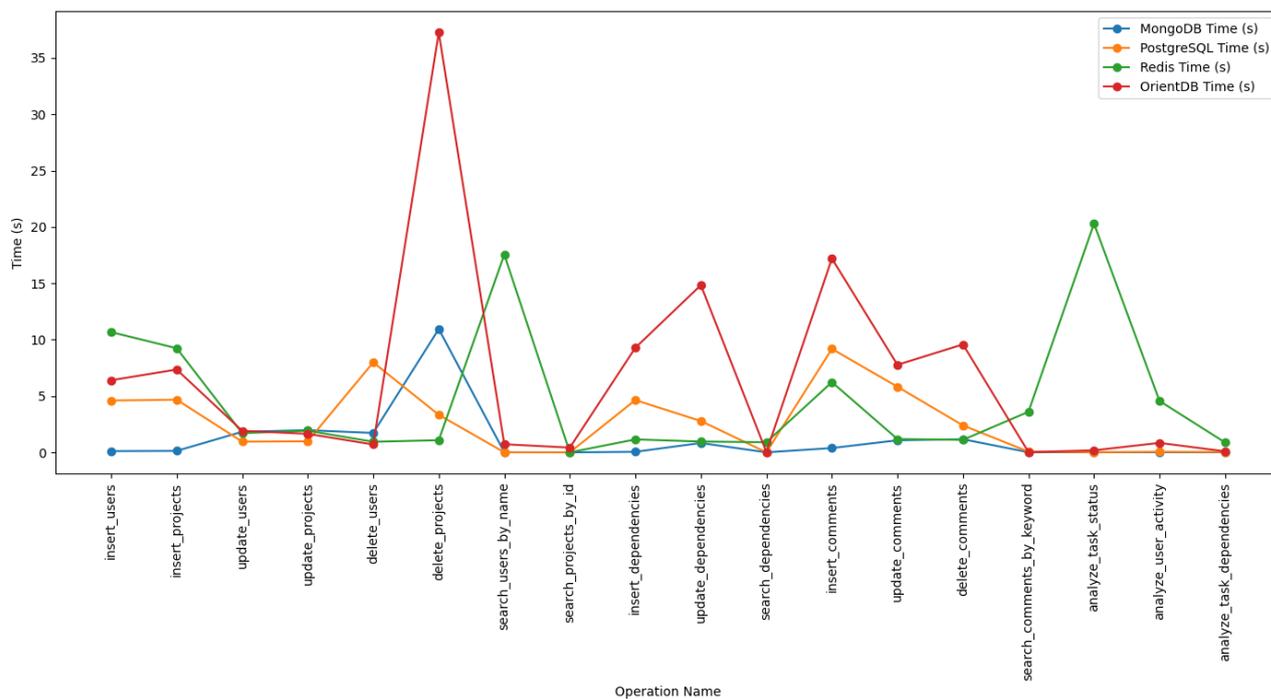


Рисунок 1 - График отображающий скорость работы четырёх типов баз данных: PostgreSQL, MongoDB, Redis, OrientDB

Сравнение баз данных на основании графика:

1. MongoDB:

- Быстро справляется с операциями вставки и поиска, особенно если сравнивать с PostgreSQL и Redis.
- Устойчив к изменениям данных, как видно из быстрого времени обновления и удаления.

- Сравнительно медленнее в операциях анализа.
2. **PostgreSQL:**
- Хорошо справляется с обновлением данных, даже быстрее Redis и OrientDB.
 - Сравнительно медленнее в операциях вставки и удаления по сравнению с MongoDB.
 - Эффективен в поиске, особенно в операциях, связанных с точным соответствием.
3. **Redis:**
- Отличается высокой производительностью в поиске и анализе задач, но имеет длинное время выполнения операций вставки и удаления.
 - Часто медленнее по сравнению с другими системами при выполнении сложных операций, таких как удаление и анализ.
4. **OrientDB:**
- Замечен как самый медленный во многих операциях, особенно в операциях удаления и анализа.
 - Быстро выполняет поисковые запросы и анализ задач, но его производительность в других операциях значительно ниже.

Таким образом, выбор подходящей базы данных зависит от конкретных требований приложения и типов операций. MongoDB и PostgreSQL показывают хорошую производительность в различных областях, Redis полезен для высокоскоростного поиска и анализа, тогда как OrientDB может быть менее предпочтителен для операций, требующих высокой скорости.[7]

Заключение

В данном исследовании была проведена оценка производительности и эффективности различных баз данных в контексте создания системы управления проектами, включающей планирование задач, отслеживание прогресса, управление командой и отчетность. Задача была разделена на четыре подзадачи, каждая из которых требовала использования разных типов баз данных для оптимального выполнения.

PostgreSQL продемонстрировала высокую скорость выполнения операций вставки, обновления и поиска данных, что делает её идеальной для структурированных данных. Однако значительные вариации во времени удаления пользователей указывают на необходимость оптимизации настройки базы данных для конкретных сценариев использования. PostgreSQL полностью поддерживает принципы ACID (атомарность, согласованность, изолированность, долговечность), обеспечивая надежность транзакций и согласованность данных. Это делает её идеальной для критически важных приложений, где необходимо гарантировать целостность данных.

OrientDB продемонстрировала свою эффективность при работе с графами зависимостей, обеспечивая хорошие результаты в создании и обновлении связей между задачами. Это подтверждает её пригодность для представления сложных связей и зависимостей. OrientDB также поддерживает ACID-транзакции, что позволяет выполнять надежные операции в графовой модели данных. Это важно для приложений, требующих точных и согласованных связей между данными.

MongoDB показала себя как надежное решение для хранения и поиска неструктурированных данных, таких как комментарии и файлы, обеспечивая гибкость и

масштабируемость. MongoDB изначально не была ориентирована на полную поддержку ACID, но с версией 4.0 введена поддержка многодокументных транзакций, что улучшило её способность обеспечивать согласованность данных в более сложных сценариях.

Redis продемонстрировала высокую скорость выполнения аналитических запросов, что делает её подходящей для задач, требующих быстрой обработки и анализа данных. Хотя Redis известна своей высокой производительностью, она обеспечивает поддержку транзакций, но не гарантирует полной реализации всех принципов ACID. В Redis транзакции атомарны и изолированы, однако долговечность данных может зависеть от настроек устойчивости (persistence settings).

Оптимальный выбор базы данных зависит от специфических требований каждой подзадачи. Использование гибридных архитектур, комбинирующих различные типы баз данных, позволяет добиться максимальной производительности и эффективности системы в целом, учитывая требования к надежности и согласованности данных.

Список литературы

1. Джани, Д. Х., и Праджапати, Б. Р. (2018). Сравнительное исследование NoSQL и реляционных баз данных. Получено из ResearchGate (https://www.researchgate.net/publication/326019759_A_Comparative_Study_of_NoSQL_and_Relational_Database).
2. Шрихарша, А. Б. и Кумар, С. В. (2017). Сравнительный анализ производительности реляционных баз данных и графовых баз данных NoSQL. Получено из ResearchGate (https://www.researchgate.net/publication/317312130_Predictive_Performance_Comparison_Analysis_of_Relational_NoSQL_Graph_Databases).
3. Торки, А. М., & Абдо, А. А. (2021). Сравнительное исследование реляционной базы данных MySQL и базы данных MongoDB NoSQL. Извлечено из [ResearchGate](https://www.researchgate.net/publication/352039638_Comparative_Study_Between_the_MySQL_Relational_Database_and_the_MongoDB_NoSQL_Database).
4. IJRAR. (2018). Сравнительный анализ баз данных SQL и NoSQL. Взято из [IJRAR](http://ijrar.com/upload_issue/ijrar_issue_237.pdf).
5. АрангоДБ. (2018). Тест производительности NoSQL 2018: MongoDB, PostgreSQL, OrientDB, Neo4j, ArangoDB. Извлечено из ArangoDB (<https://arangodb.com/2018/02/nosql-performance-benchmark-2018-mongodb-postgresql-orientdb-neo4j-arangodb/>).
6. Перван Г., Роблес Р. Дж. (2022). Большие данные и сравнительный анализ производительности: реляционные и альтернативные Базы данных NoSQL. В: Материалы Международной конференции по большим данным и бенчмаркингу эффективности. Извлечено из Springer (https://link.springer.com/chapter/10.1007/978-3-031-03884-6_14).
7. Шрихарша, А. Б. и Кумар, С. В. (2017). Сравнительный анализ производительности реляционных баз данных и графовых баз данных NoSQL. Получено из ResearchGate (https://www.researchgate.net/publication/317312130_Predictive_Performance_Comparison_Analysis_of_Relational_NoSQL_Graph_Databases).

References

1. Jani, D. H., & Prajapati, B. R. (2018). A Comparative Study of NoSQL and Relational Database. Retrieved from ResearchGate (https://www.researchgate.net/publication/326019759_A_Comparative_Study_of_NoSQL_and_Relational_Database).
 2. Shreeharsha, A. B., & Kumar, S. V. (2017). Predictive Performance Comparison Analysis of Relational & NoSQL Graph Databases. Retrieved from ResearchGate (https://www.researchgate.net/publication/317312130_Predictive_Performance_Comparison_Analysis_of_Relational_NoSQL_Graph_Databases).
 3. Torkey, A. M., & Abdo, A. A. (2021). Comparative Study Between the MySQL Relational Database and the MongoDB NoSQL Database. Retrieved from [ResearchGate](https://www.researchgate.net/publication/352039638_Comparative_Study_Between_the_MySQL_Relational_Database_and_the_MongoDB_NoSQL_Database).
 4. IJRAR. (2018). Comparative Analysis of SQL and NoSQL Databases. Retrieved from [IJRAR](http://ijrar.com/upload_issue/ijrar_issue_237.pdf).
 5. ArangoDB. (2018). NoSQL Performance Benchmark 2018: MongoDB, PostgreSQL, OrientDB, Neo4j, ArangoDB. Retrieved from ArangoDB (<https://arangodb.com/2018/02/nosql-performance-benchmark-2018-mongodb-postgresql-orientdb-neo4j-arangodb/>).
 6. Pervan, G., & Robles, R. J. (2022). Big Data and Performance Benchmarking: Relational vs. NoSQL Databases. In: Proceedings of the International Conference on Big Data and Performance Benchmarking. Retrieved from Springer (https://link.springer.com/chapter/10.1007/978-3-031-03884-6_14).
 7. Shreeharsha, A. B., & Kumar, S. V. (2017). Predictive Performance Comparison Analysis of Relational & NoSQL Graph Databases. Retrieved from ResearchGate (https://www.researchgate.net/publication/317312130_Predictive_Performance_Comparison_Analysis_of_Relational_NoSQL_Graph_Databases).
-