

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ РЕЛЯЦИОННЫХ И ВРЕМЕННЫХ БД ПРИ ОБРАБОТКЕ ТЕЛЕМЕТРИИ IOT-СИСТЕМ

В работе проводится практический сравнительный анализ производительности реляционной СУБД и специализированной Time-Series СУБД. Исследование базируется на данных аппаратного комплекса экологического мониторинга (ESP8266) и анализирует скорость записи и выполнения аналитических запросов при различной интенсивности нагрузки.

Введение

Стремительное развитие интернета вещей (IoT) приводит к экспоненциальному росту объемов телеметрической информации [1]. Специфика IoT-данных заключается в их потоковой природе – это непрерывные временные ряды (Time-Series). Главной инженерной проблемой становится выбор СУБД, так как классические реляционные системы часто демонстрируют деградацию производительности при масштабировании [2]. Целью данной работы является практическое сравнение PostgreSQL и InfluxDB в условиях реальной потоковой нагрузки.

I. Методика исследования

Для проведения исследования был спроектирован испытательный стенд на базе микроконтроллера NodeMCU (ESP8266), который является распространенным стандартом для встраиваемых систем [3]. К модулю подключен комплекс сенсоров: датчик DHT11, фоторезистор и микрофон. Стенд настроен на опрос датчиков с отправкой данных по HTTP. Для отказоустойчивости применен механизм локального циклического буферизирования на SD-карте, минимизирующий риск потери пакетов на «краевом» (Edge) уровне [4].

В PostgreSQL спроектирована реляционная схема, где каждая метрика представлена отдельным столбцом, а первичным ключом выступает Timestamp (B-Tree индекс). В InfluxDB применена линейная структура: параметры записываются как Fields, а идентификатор устройства — как Tag, что изначально оптимизировано для потоковой работы с временными рядами [5].

II. Анализ производительности

Тестирование проводилось по направлениям: скорость записи (Write Workload) и скорость аналитических выборок. Генератор нагрузки отправлял пакеты с частотой 100, 1000 и 5000 записей в секунду. Для аналитики применялись типичные IoT-выборки:

вычисление скользящего среднего, поиск пиковых значений и даунсэмплинг (Downsampling).

Результаты показали, что InfluxDB превосходит PostgreSQL в скорости записи за счет отказа от механизмов блокировок строк и транзакционной целостности (ACID). Выполнение оконных функций в InfluxDB происходит в среднем в 3–5 раз быстрее благодаря колоночному принципу хранения и встроенным алгоритмам компрессии данных. PostgreSQL сохраняет преимущество только при необходимости сложных JOIN-операций со статическими бизнес-данными.

Заключение

В ходе исследования доказано, что для систем мониторинга, где преобладают операции вставки и агрегации по временным окнам, целесообразно использовать специализированные Time-Series СУБД. Установлено, что InfluxDB обеспечивает выполнение аналитических запросов в среднем в 3–5 раз быстрее по сравнению с реляционными аналогами. Спроектированный аппаратный комплекс успешно подтвердил способность обеспечивать базы данных непрерывным потоком информации.

Список литературы

1. Макаров, В. В., Смирнов, А. А. Архитектура высоконагруженных систем интернета вещей. – М. : Издательство «Радио и связь», 2022. – 256 с.
2. Gomez, J., Perez, L. Performance Evaluation of Relational and Time-Series Databases for IoT Data Management // Journal of Database Management. – 2023. – Vol. 34, No. 2. – P. 45-62.
3. Иванов, П. С. Проектирование встраиваемых систем на базе микроконтроллеров семейства ESP8266. – СПб. : БХВ-Петербург, 2021. – 320 с.
4. Chen, Y. [et al.] Edge Computing and Local Data Buffering in Unreliable IoT Networks // IEEE Internet of Things Journal. – 2022. – Vol. 9, No. 14. – P. 12345-12355.
5. Naqvi, S., Fakhri, A. Benchmarking Time-Series Databases: InfluxDB vs. Relational Alternatives // Data Engineering Proceedings. – 2023. – Vol. 11, No. 4. – P. 112-128.

Горбель Алексей Артёмович, студент факультета информационных технологий и управления, alekseygorbel@gmail.com.

Научный руководитель: Лаппо Александр Игоревич, старший преподаватель кафедры ИТАС, lappo@bsuir.by.