Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 004.651

СРАВНИТЕЛЬНЫЙ АНАЛИЗ OLAP-СИСТЕМ ХРАНЕНИЯ ДАННЫХ ДЛЯ АНАЛИТИЧЕСКОГО СЕРВИСА

Аскеров Салех Теймур оглы,
Студент группы ИУК5-21М
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
askerovst1@student.bmstu.ru

Кондратьева Светлана Дмитриевна,
Доцент кафедры ИУК5 «Системы обработки информации»
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
sdkond@bmstu.ru

Романовский Илья Олегович,
Студент группы ИУК5-21М
Калужский филиал Московского государственного технического университета имени Н.Э. Баумана
romanovskiyio@student.bmstu.ru

Аннотация

В данной работе проводится экспериментальное сравнение двух OLAP-систем хранения данных для аналитического сервиса Wildberries: DuckDB и ClickHouse Server. Исследование направлено на оценку скорости загрузки данных, эффективности хранения и времени выполнения типовых аналитических SQL-запросов на наборах данных объёмом 100, 250 и 500 МБ. В эксперименте DuckDB рассматривается как embedded OLAP-база данных, работающая внутри процесса приложения, а ClickHouse — как отдельная серверная аналитическая СУБД. Полученные результаты показывают, что DuckDB быстрее выполняет одиночные локальные запросы, однако ClickHouse остаётся более обоснованным выбором для production-сервиса, где важны многопользовательская работа, разделение backend и хранилища, мониторинг, управление ресурсами и дальнейшее масштабирование.

Ключевые слова: Wildberries, OLAP, DuckDB, ClickHouse, аналитическое хранилище, marketplace analytics, MergeTree, embedded analytics, серверная аналитика, производительность запросов, масштабируемость.

COMPARATIVE ANALYSIS OF OLAP-DATA STORAGE SYSTEMS FOR ANALYTICAL SERVICES

Saleh T. Askerov,
Student of group IUK5-21M
Bauman Moscow State Technical University (Kaluga Branch)
askerovst1@student.bmstu.ru

Svetlana D. Kondratieva,
Associate Professor of the Department of IUK5 "Information Processing Systems"
Bauman Moscow State Technical University (Kaluga Branch)
sdkond@bmstu.ru

Ilya O. Romanovskiy,
Student of group IUK5-21M
Bauman Moscow State Technical University (Kaluga Branch)
romanovskiyio@student.bmstu.ru

ABSTRACT

In this paper, an experimental study is conducted on methods for optimizing the storage and processing of financial reports of a marketplace in an analytical pipeline based on DuckDB. Two models of data organization are considered: monolithic storage in the form of a single optimized Parquet file and partitioned storage in the form of a set of files separated by time intervals. The features of each model, the impact of the file structure on the performance of analytical queries and the cost of data preparation stages are analyzed. The experimental part includes testing typical financial queries (aggregations, ranking, window functions, time slices) in cold and warm data access modes. Particular attention is paid to estimating the median and percentile query execution time, as well as the stability of operation with increasing data volume. Based on the results obtained, recommendations are formulated for choosing a financial data storage model for analytics and reporting tasks.

Keywords: financial reports, DuckDB, Parquet, analytical data processing, storage optimization, database file structure, query performance, cold and warm modes, partitioning, aggregate queries.

Введение

Сервис аналитики Wildberries должен хранить и быстро обрабатывать большие объёмы данных: продажи, заказы, остатки, рекламные расходы, финансовые показатели и справочники товаров. Пользовательские сценарии включают построение отчётов по дням, расчёт эффективности рекламных кампаний, анализ прибыли, оценку складских остатков и сравнение периодов. Для таких задач важно выбрать OLAP-систему, которая обеспечит быстрое выполнение аналитических запросов и будет соответствовать архитектуре backend-сервиса [2].

DuckDB и ClickHouse решают похожий класс аналитических задач, но имеют разную архитектурную природу. DuckDB предназначен для embedded-аналитики и работает внутри процесса приложения [1]. ClickHouse работает как отдельная серверная OLAP-СУБД и подходит для централизованного аналитического хранилища. Поэтому сравнение

должно учитывать не только скорость одиночных запросов, но и эксплуатационную пригодность системы.

Используемые системы и критерии сравнения

DuckDB использовался как встроенная аналитическая база данных, подключаемая из Python-приложения. Для каждого набора данных создавался отдельный файл базы DuckDB, в который загружалась таблица sales.

ClickHouse использовался в режиме отдельного сервера [5]. Для эксперимента был поднят локальный ClickHouse Server, а загрузка и выполнение запросов проводились через clickhouse client по TCP-подключению. Таблица sales создавалась на движке MergeTree с сортировкой по дате, бренду и артикулу.

Критерии сравнения:

- Время загрузки CSV в физическое хранилище;
- размер данных на диске после загрузки;
- среднее время выполнения типовых аналитических SQL-запросов [4];
- поведение при росте объёма данных;
- архитектурная пригодность для production-сервиса аналитики [3].

Практическая часть

Эксперимент проводился на трёх CSV-наборах Wildberries-подобной структуры: 100, 250 и 500 МБ [6]. Данные содержат дату заказа, артикул, бренд, идентификатор рекламной кампании, склад, показы, клики, рекламные расходы, количество заказов, выручку, себестоимость и остатки. Для обеих систем использовалась одинаковая логическая таблица sales и одинаковый набор аналитических запросов.

Типовые запросы:

- Q1 — выручка, рекламные расходы, прибыль и ROAS по дням и брендам;
- Q2 — ROAS по рекламным кампаниям;
- Q3 — топ товаров по прибыли;
- Q4 — остатки, заказы и выручка по складам и дням.

Таблица 1 — Результаты загрузки и хранения

Данные	DuckDB загрузка, сек	DuckDB хранение, МБ	ClickHouse загрузка, сек	ClickHouse хранение, МБ
100 МБ	0,906	26,3	1,173	43,5
250 МБ	1,697	65,0	2,061	108,9
500 МБ	2,686	128,8	3,263	217,5

По результатам загрузки DuckDB показал немного меньшие значения времени и более компактное физическое хранение. На наборе 500 МБ загрузка в DuckDB заняла 2,686 секунды, а в ClickHouse Server — 3,263 секунды. Размер базы DuckDB составил 128,8 МБ, тогда как таблица ClickHouse заняла 217,5 МБ. Это указывает на эффективность DuckDB как локального embedded-хранилища для одиночных аналитических сценариев.

Таблица 2 — Результаты аналитических запросов на 500 МБ

Запрос	DuckDB, сек	ClickHouse Server, сек
Q1 daily brand	0,010	0,227
Q2 campaign ROAS	0,031	0,316
Q3 top products	0,106	0,398
Q4 warehouse daily	0,041	0,318

На одиночных локальных запросах DuckDB оказался быстрее ClickHouse Server. Особенно заметно это на запросах Q1 и Q2, где DuckDB выполнял расчёты за сотые доли секунды. ClickHouse Server также показал стабильное время, однако уступил DuckDB в выбранной постановке эксперимента. Такой результат объясняется тем, что DuckDB работает внутри процесса приложения и обращается к данным напрямую, без дополнительных сетевых вызовов, тогда как ClickHouse функционирует как отдельная СУБД с клиент-серверной архитектурой: каждый запрос требует установки соединения, передачи SQL по сети и обратной пересылки результата.

Масштабируемость

При росте данных с 100 до 500 МБ обе системы сохраняли работоспособность и предсказуемый рост времени выполнения. DuckDB оставался быстрее на одиночных запросах, однако ClickHouse Server показал стабильное серверное поведение и может быть дополнительно раскрыт в экспериментах с параллельными пользователями, постоянной фоновой загрузкой и материализованными витринами. Именно такие сценарии ближе к реальному сервису аналитики маркетплейса.

Заключение

В ходе выполнения научно-исследовательской работы было проведено экспериментальное сравнение DuckDB и ClickHouse Server как OLAP-систем хранения данных для аналитического сервиса Wildberries. Эксперимент включал загрузку CSV-наборов объёмом 100, 250 и 500 МБ, создание физических хранилищ и выполнение типовых аналитических SQL-запросов.

Несмотря на то что DuckDB показал лучшие результаты в простых одиночных тестах, это не означает, что он автоматически является лучшим выбором для production-сервиса аналитики Wildberries. В данном исследовании DuckDB следует рассматривать как оптимальное решение для локальной аналитики, MVP и сценариев обработки данных одного пользователя или одного продавца.

ClickHouse имеет преимущество в архитектурной пригодности для серверного аналитического продукта. Он работает как отдельное OLAP-хранилище, может обслуживать множество подключений, отделяет тяжёлые аналитические запросы от backend-приложения, предоставляет системные таблицы для наблюдения за запросами и ресурсами, поддерживает пользователей, профили и ограничения. Для сервиса Wildberries-аналитики это критично, потому что нагрузка формируется не одним локальным запросом, а множеством пользователей, регулярной загрузкой данных и повторяющимися dashboard-сценариями.

Таким образом, при разработке аналитического сервиса на базе OLAP-хранилища рекомендуется использовать ClickHouse в качестве основной системы управления данными — это позволяет обеспечить масштабируемость при росте числа пользователей и объёмов данных, высокую производительность повторяющихся аналитических запросов, поддержку потоковых обновлений, гибкое управление доступом и надёжность промышленной эксплуатации.

Список литературы:

1. Mark Raasveldt, Hannes Mühleisen. DuckDB: an Embeddable Analytical Database // Proceedings of the 2019 International Conference on Management of Data (SIGMOD ’19). – 2019. – Pp. 1981–1984. – DOI 10.1145/3299869.3320212.
2. Alexey Milovidov, Maxim Zaitsev. ClickHouse: A Column-Oriented Database Management System for Real-Time Analytics // Proceedings of the VLDB Endowment. – 2020. – Vol. 13, No. 12. – Pp. 3248–3261. – DOI 10.14778/3415478.3415560.

3. Daniel J. Abadi, Samuel Madden, Miguel Ferreira. Integrating Compression and Execution in Column-Oriented Database Systems // Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data. – 2006. – Pp. 671–682. – DOI 10.1145/1142473.1142548.
4. Abadi D. J., Madden S., Hachem N. Column-Stores vs. Row-Stores: How Different Are They Really? // Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. – 2008. – Pp. 967–980. – DOI 10.1145/1376616.1376712.
5. Pavlo A., Paulson E., Rasin A. et al. A Comparison of Approaches to Large-Scale Data Analysis // Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data. – 2009. – Pp. 165–178. – DOI 10.1145/1559845.1559865.
6. Boncz P., Kersten M. L., Manegold S. Breaking the Memory Wall in MonetDB // Communications of the ACM. – 2008. – Vol. 51, No. 12. – Pp. 77–85. – DOI 10.1145/1409360.1409380.

References:

1. Mark Raasveldt, Hannes Mühleisen. DuckDB: an Embeddable Analytical Database // Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19). – 2019. – Pp. 1981–1984. – DOI 10.1145/3299869.3320212.
2. Alexey Milovidov, Maxim Zaitsev. ClickHouse: A Column-Oriented Database Management System for Real-Time Analytics // Proceedings of the VLDB Endowment. – 2020. – Vol. 13, No. 12. – Pp. 3248–3261. – DOI 10.14778/3415478.3415560.
3. Daniel J. Abadi, Samuel Madden, Miguel Ferreira. Integrating Compression and Execution in Column-Oriented Database Systems // Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data. – 2006. – Pp. 671–682. – DOI 10.1145/1142473.1142548.
4. Abadi D. J., Madden S., Hachem N. Column-Stores vs. Row-Stores: How Different Are They Really? // Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. – 2008. – Pp. 967–980. – DOI 10.1145/1376616.1376712.
5. Pavlo A., Paulson E., Rasin A. et al. A Comparison of Approaches to Large-Scale Data Analysis // Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data. – 2009. – Pp. 165–178. – DOI 10.1145/1559845.1559865.
6. Boncz P., Kersten M. L., Manegold S. Breaking the Memory Wall in MonetDB // Communications of the ACM. – 2008. – Vol. 51, No. 12. – Pp. 77–85. – DOI 10.1145/1409360.1409380.