П. А. Арсенович

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ КОЛОНОЧНЫХ И СТРОКОВЫХ ФОРМАТОВ ПРИ МАСШТАБИРОВАНИИ ОЗЕР ДАННЫХ

Представлен сравнительный анализ производительности и ресурсоемкости форматов хранения больших данных (JSON, CSV, Apache Avro, Apache Parquet) в архитектуре «озер данных». Целью исследования является оценка деградации временной эффективности при росте объема хранимой информации.

Введение

Концепция «озер данных» (Data Lakes) на базе таких платформ, как Apache Hadoop, стала стандартом для аналитики больших данных [1]. Однако в этих системах нет строго установленного формата файлов. Разработчикам приходится выбирать между текстовыми (JSON, CSV), линейно-ориентированными и колоночными форматами [2].

Ошибка в выборе критически влияет на производительность аналитических запросов и стоимости хранения: серьезной проблемой становится время поиска информации и объем занятого дискового пространства. Кроме того, любая система обработки данных неизбежно сталкивается с постоянным ростом их объема, поэтому эффективность форматов необходимо оценивать именно в динамике масштабирования.

I. Методология и архитектура форматов

В экосистеме больших данных форматы хранения классифицируются по своей внутренней структуре. Форматы JSON и CSV являются простыми текстовыми форматами. CSV хранит данные в виде строк со значениями, разделенными запятыми, а JSON – в виде пар «ключ-значение», что делает его наиболее избыточным. В свою очередь, Apache Avro представляет собой бинарный линейно-ориентированный (строковый) формат.

В отличие от них, Apache Parquet является бинарным колоночно-ориентированным форматом, который оптимизирован специально для аналитических платформ [3]. Для более глубокого понимания эффективности Apache Parquet следует рассмотреть его внутреннюю структуру [4]. Архитектура использует гибридную модель: файл физически делится на логические группы строк, которые внутри разбиты на колоночные блоки, а те, в свою очередь, делятся на страницы. Такая архитектура позволяет аналитическому движку считывать только запрошенные столбцы и пропускать целые блоки данных, что радикально ускоряет выполнение запросов.

Для оценки эффективности форматов был развернут локальный экспериментальный стенд с использованием контейнеризации и фреймворка Apache Spark. В рамках исследования были сгенерированы синтетические наборы данных объемом 1, 5, 10 и 25 млн записей. Эксперимент представлял собой многократный запуск запросов различной сложности к данным в форматах JSON, CSV, Apache Avro и Apache Parquet. Нагрузочное тестирование включало аналитические запросы на фильтрацию по условию и агрегацию, после чего замерялись объем занимаемого пространства и время выполнения запроса.

II. Результаты эксперимента

В ходе исследования был проведен сравнительный анализ занимаемого объема данных и времени выполнения запросов. Анализ показал, что формат Apache Parquet существенно снижает объем хранимой информации по сравнению с текстовыми форматами за счет колоночно-ориентированного хранения и встроенных алгоритмов сжатия. В таблице 1 представлена динамика изменения объема файлов при масштабировании набора данных.

Таблица 1 – Оценка ресурсоемкости хранения данных (в МБ)

Формат	1M	5M	10M	25M
JSON	135.18	684.37	1370.85	3458.93
CSV	69.37	355.35	712.82	1813.84
Avro	26.98	142.15	287.81	732.70
Parquet	20.79	108.59	219.17	559.41

Как видно из таблицы 1, текстовый формат JSON демонстрирует наихудшие показатели из-за дублирования ключей в каждой записи. CSV экономит пространство за счет их отсутствия, однако колоночный формат Parquet все равно значительно выигрывает: при масштабировании до 25 млн записей он продемонстрировал сжатие в 6.2 раза по отношению к JSON и в 3.2 раза по отношению к CSV.

При оценке временной эффективности замерялось время выполнения операций фильтрации и агрегации. Результаты нагрузочного тестирования представлены в таблицах 2 и 3.

Таблица 2 – Время выполнения фильтрации (в секундах)

Формат	1M	5M	10M	25M
JSON	0.37	1.19	2.42	7.66
CSV	0.28	0.70	1.74	4.03
Avro	0.11	1.07	1.27	1.97
Parquet	0.05	0.08	0.11	0.26

Таблица 3 – Время выполнения агрегации (в секундах)

Формат	1M	5M	10M	25M
JSON	0.47	1.56	3.42	9.05
CSV	0.31	0.95	1.65	3.97
Avro	0.12	0.43	0.83	2.01
Parquet	0.08	0.16	0.46	0.84

Анализ таблиц 2 и 3 демонстрирует превосходство колоночного формата Parquet на объеме 25 млн строк.

При фильтрации Parquet показал время 0.26 сек против 7.66 сек у JSON и 4.03 сек у CSV. Такое превосходство объясняется возможностью использовать метаданные для пропуска нерелевантных страниц без их полного считывания с диска [5]. При агрегации деградация производительности Avro (2.01 сек против 0.84 сек у Parquet) вызвана необходимостью считывать всю строку целиком для получения значений отдельных полей. Низкая производительность CSV и JSON также обусловлена затратами ресурсов на построчный парсинг текстовых данных.

Для наглядной оценки динамики деградации производительности при масштабировании объема хранимой информации было вычислено усредненное время выполнения аналитических запросов (агрегации и фильтрации). Визуализация результатов представлена на рисунке 1.

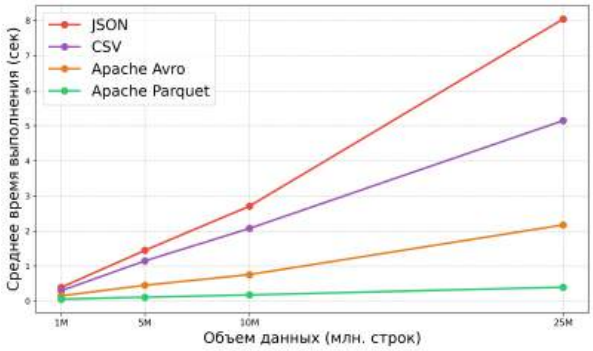

Рис. 1 – Усредненная деградация производительности аналитических запросов

Анализ графика на рисунке 1 наглядно подтверждает гипотезу о деградации производительности: текстовые (JSON, CSV) и бинарный строковый (Avro) форматы демонстрируют быстрый, практически линейный рост времени обработки, что выражается в крутом угле наклона их графиков. В свою очередь, колоночный формат (Parquet) сохраняет высокую эффективность при масштабировании, поскольку его время выполнения растет крайне медленно, а график остается практически пологим.

Итоговый рейтинг эффективности форматов в динамике роста объема данных выстраивается в следующую последовательность (от наиболее эффективного к наименее): $Parquet < Avro < CSV < JSON$.

III. Заключение

В ходе исследования была доказана важность выбора формата хранения данных при проектировании масштабируемых аналитических систем. Экспериментальные данные подтвердили, что использование колоночно-ориентированного формата Apache Parquet позволяет не только радикально сократить потребление дискового пространства, но и многократно ускорить выполнение аналитических запросов.

Установлено, что при увеличении объема данных строковые и текстовые форматы подвержены быстрой деградации производительности при выполнении агрегации и фильтрации. Таким образом, для построения эффективных «озер данных» целесообразно использовать колоночные форматы, в то время как форматы JSON, CSV и Avro следует оставить для задач обмена сообщениями, интеграции и потоковой передачи данных.

Список литературы

1. Vohra, D. Apache Parquet / D. Vohra // Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools. – Berkeley, CA : Apress, 2016. – С. 325-335.
2. Белов, В. А. Оценка эффективности обработки больших объемов данных в реляционных и колоночных форматах / В. А. Белов, Д. Ю. Ильин, Е. В. Никульчев // Вычислительные технологии. – 2022. – Т. 27, № 3. – С. 46-65.
3. Boufea, A. Managing Variant Calling Files the Big Data Way: Using HDFS and Apache Parquet / A. Boufea, R. Finkers, M. van Kaauwen [et al.] // Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT '17). – 2017. – С. 219-226.
4. Ахмедов, Э. Р. Формат Parquet: обзор структуры и методы оптимизации для высокой производительности в распределённых системах / Э. Р. Ахмедов, И. В. Хлудов // Universum: технические науки. – 2025. – №4(133). – С. 35-38.
5. Braams, B. Predicate Pushdown in Parquet and Apache Spark : Master's Thesis / B. Braams. – Amsterdam : Vrije Universiteit Amsterdam, 2018. – 89 с.

Арсенович Павел Александрович, студент кафедры информационных технологий автоматизированных системы БГУИР, arsenpashamail@gmail.com.

Научный руководитель: Шабанович Роман Александрович, ассистент кафедры информационных технологий автоматизированных системы БГУИР, r.shabanovich@bsuir.by.