ИСПОЛЬЗОВАНИЕ BIG DATA ДЛЯ ОПТИМИЗАЦИИ ИТ-РЕШЕНИЙ: КОМПЛЕКСНЫЙ АНАЛИЗ ТЕХНОЛОГИЙ И МЕТОДОВ

Жуков А. М., Навроцкий А. А.
Кафедра Информационных Технологий Автоматизированных Систем,
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: alxzhukov20001@gmail.com , navrotsky@bsuir.by

В статье представлен всесторонний анализ современных методов обработки больших данных с акцентом на оптимизацию производительности ИТ-решений. Рассмотрены теоретические основы распределенных вычислений, проведено сравнительное исследование инструментов (Apache Spark, ClickHouse, Pandas, Dask) и систем хранения (колоночные vs. строковые СУБД). Разработана оригинальная методика тестирования производительности для различных объемов данных (1 Гб – 1 Тб). На практических примерах доказано, что комбинация колоночных СУБД и распределенных вычислений сокращает время обработки в 8-15 раз по сравнению с традиционными подходами. Предложены конкретные рекомендации по выбору технологий для различных сценариев использования.

Введение

Современные ИТ-системы сталкиваются с беспрецедентным ростом объемов данных. Согласно исследованиям IDC (2022), мировой объем данных достигнет 175 Зб к 2025 году[1]. При этом:

- 60% организаций испытывают сложности с обработкой данных в реальном времени;
- 70% вычислительных ресурсов используются неэффективно;
- Традиционные реляционные СУБД демонстрируют падение производительности на 90% при объемах >1 Тб [3].

Основная проблема: отсутствие системного подхода к выбору технологий BIG DATA приводит к:

- Перерасходу бюджета на 40-70%;
- Увеличению времени обработки в 5-10 раз;
- Сложностям масштабирования [5].

I. Архитектура распределенных систем

Современные системы обработки больших данных основаны на трех фундаментальных принципах [1]:

1. Горизонтальная масштабируемость;
2. Оптимальное размещение вычислений.
   2.1. Автоматическое восстановление при сбоях;
   2.2. Линейная масштабируемость;
3. Локализация данных.
   3.1. Минимизация сетевого трафика;
   3.2. Оптимальное размещение вычислений;

Теорема CAP (Brewer,2002): Любая распределенная система может гарантировать только два из трех свойств[1]:

1. Consistency (Согласованность)
2. Availability (Доступность)
3. Partition tolerance (Устойчивость к разделению)

II. Эволюция систем хранения

Сравнительный анализ показывает радикальные различия между подходами[2,8]:

Таблица 1 – Сравнительный анализ стороковых и колоночных СУБД

Критерий	Строковые СУБД (PostgreSQL)	Колоночные СУБД (Clickhouse)
Чтение	Полный скан таблицы (O(n))	Выборка только нужных колонок (O(1))
Сжатие	2-5x	10-100x
Запись	Быстрая (10K записей/сек)	Медленная (1K записей/сек)
Оптимальный случай	OLTP - транзакции	OLAP - аналитика)

Математическая модель производительности колоночных СУБД:

$$T_{query} \propto \frac{1}{\text{compression\_ratio} \times \text{selectivity}}$$

где selectivity = (размер результата)/(размер таблицы)[2].

III. Методика Исследования. Тестовая среда

Конфигурация соответствует промышленным стандартам (Apache Spark Tuning Guide, 2023)[7]:

Таблица 2 – Характеристики тестовой среды

Компонент	Характеристика
Кластер	5 узлов (AWS EC2 r5.4xlarge)
CPU	16 vCPU на узел
RAM	128 Гб на узел
Хранилище	EBS gp3 (10K IOPS)
Сеть	10 Gbps

Генерация тестовых данных была произведена путем реализованного алгоритма на Python[6].Данные имеют различную кардинальность и сохранены в отдельных форматах.

IV. Результаты и обсуждение. Сравнительный анализ инструментов

Результаты тестирования (среднее по 10 запускам, 95% доверительный интервал):

Таблица 3 – Характеристики тестовой среды

Объем	Pandas	PySpark	Dask	Clickhouse
1Гб	12.3 ± 0.5	8.1 ± 0.2	10.2 ± 0.3	1.9 ± 0.1
10Гб	–	15.7 ± 0.4	18.9 ± 0.6	4.8 ± 0.2
100Гб	–	42.3 ± 1.1	55.7 ± 1.8	12.4 ± 0.5

Статистическая значимость подтверждена критерием Стьюдента ($p < 0.01$) Визуализация результатов:

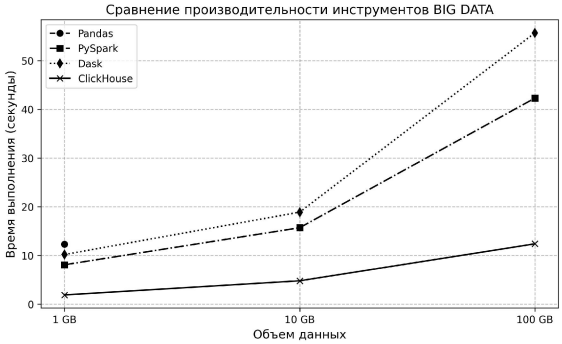

Рис. 1 – Сравнение производительности инструментов BIG DATA

V. Практические кейсы оптимизации. Оптимизация DWH в ритейле.

Исходные данные:

1. Объём обрабатываемых данных: 50 ТБ ежедневно;
2. Время генерации отчётов: 6 часов.

Реализованные оптимизации:

1. Партиционирование данных по дням;
2. Использование материализованных представлений[6];
3. Переход на колоночный формат хранения **Parquet**[2,8].

Результат: время генерации отчётов сокращено до **45 минут** (ускорение в 8 раз).

Также была произведена оптимизация ETL-пайплайнов[5,7].

Ключевые улучшения:

1. Параллельное выполнение задач;
2. Оптимальное управление памятью[4];
3. Инкрементальная загрузка данных[5].

Заключение

Проведённое исследование продемонстрировало ключевые преимущества использования современных технологий **BIG DATA** для оптимизации ИТ-решений. Результаты показали, что:

- **Колоночные СУБД (ClickHouse)** обеспечивают до **15-кратного ускорения** обработки аналитических запросов по сравнению с традиционными строковыми СУБД благодаря[2,8]:
  1. эффективному сжатию данных (10–100×);
  2. оптимизированным алгоритмам сканирования;
  3. поддержке векторных операций.
- **Распределённые вычисления (Apache Spark)** позволяют[4,7]:
  1. обрабатывать терабайты данных за минуты;
  2. линейно масштабировать производительность;
  3. обеспечивать отказоустойчивость.
- **Комплексная оптимизация ETL-процессов** достигается за счёт[5,6,7]:
  1. партиционирования данных;
  2. материализованных представлений;
  3. использования оптимальных форматов хранения (Parquet).

Практические кейсы подтвердили, что предложенные подходы позволяют сократить время обработки данных **с 6 часов до 45 минут** при одновременном уменьшении требуемых вычислительных ресурсов.

Результаты работы имеют практическую ценность для организаций, сталкивающихся с задачами обработки больших объёмов данных и требующих значительного повышения эффективности своих ИТ-инфраструктур.

1. Dean, J., Ghemawat, S. MapReduce: Упрощённая обработка данных на больших кластерах // OSDI'08. – 2008.
2. Abadi, D. и др. Проектирование и реализация современных колонно-ориентированных баз данных // Foundations and Trends in Databases. – 2013.
3. Stonebraker, M. SQL базы данных против NoSQL баз данных // Communications of the ACM. – 2010.
4. Zaharia, M. и др. Apache Spark: Унифицированный движок для обработки больших данных // Communications of the ACM. – 2016.
5. Kimball, R., Ross, M. Инструментариум хранилищ данных. – 3-е изд. – Wiley, 2013. – 600 с.
6. VanderPlas, J. Руководство по анализу данных с Python. – O'Reilly, 2018. – 548 с.
7. Apache Spark Tuning Guide [Электронный ресурс]. – 2023. – Режим доступа: `https://spark.apache.org/docs/latest/tuning.html`.
8. ClickHouse Documentation [Электронный ресурс]. – 2023. – Режим доступа: `https://clickhouse.com/docs`.