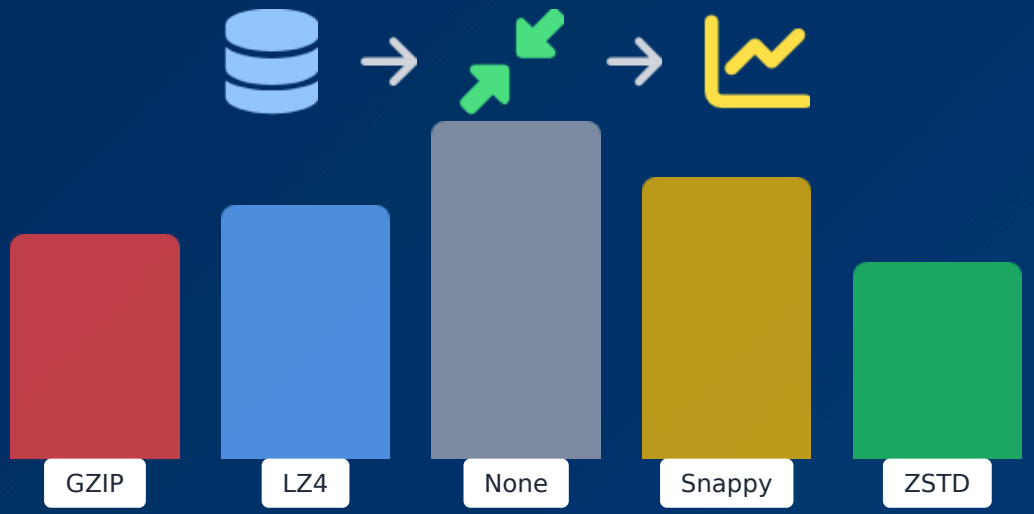
# Анализ производительности Trino

## с различными алгоритмами сжатия




Сравнительный анализ эффективности алгоритмов сжатия

Операции загрузки данных и выполнения запросов

Scale Factor: 1 (SF1)

Дата отчета: 28.07.2025

Усредненные результаты 3 тестов

# Методология исследования

## Подход к тестированию

- Данные: Набор данных TPC-H с масштабным коэффициентом 1 (SF1)
- Тестовые прогоны: Три последовательных бенчмарка для каждого алгоритма
- Операции: Загрузка данных (copy) и выполнение аналитических запросов (query)
- Метрики: Время выполнения (duration_ms) и размер файлов (total_files_size)
- Анализ: Усреднение результатов по трем тестовым прогонам

## Процесс тестирования

### Цикл тестирования

| GZIP | LZ4 | None | Snappy | ZSTD |
|---|---|---|---|---|
| Тест 1-3 | Тест 1-3 | Тест 1-3 | Тест 1-3 | Тест 1-3 |

### Тестируемые операции

**COPY**
25 таблиц загружено

**QUERY**
22 запроса выполнено

### Анализ данных

Для каждого алгоритма сжатия рассчитаны:

Время выполнения Размер данных (МБ) Степень сжатия

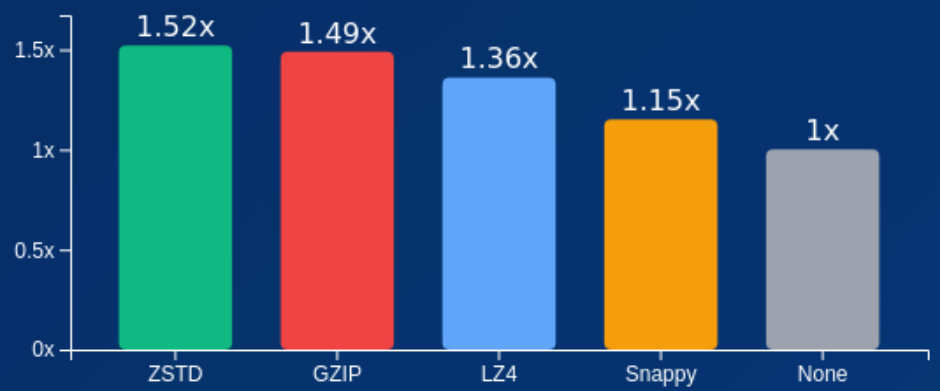
Источник данных: test_results_raw_1-2-3.csv

# Обзор производительности алгоритмов сжатия

## Сводные показатели алгоритмов сжатия

| Алгоритм | Размер данных (МБ) | Степень сжатия | Время загрузки (сек) | Время запросов (сек) | Общее время (сек) |
|---|---|---|---|---|---|
| ZSTD | **255.9** | **1.52x** | 231.5 | 116.4 | **347.9** |
| LZ4 | 285.3 | 1.36x | 234.3 | **115.5** | 349.8 |
| GZIP | 260.7 | 1.49x | 239.9 | 128.0 | 367.9 |
| Snappy | 338.4 | 1.15x | 241.2 | 117.1 | 358.3 |
| None | 388.3 | 1.00x | 249.8 | 121.7 | 371.5 |

### Степень сжатия (относительно None)




### Сравнительные характеристики

Наилучшая степень сжатия:

- **ZSTD (1.52x)** — экономия 132.4 МБ (34.1%)

Наибыстрейшая загрузка данных:

- **ZSTD (231.5 сек)** — быстрее на 18.3 сек (7.3%)

Наибыстрейшее выполнение запросов:

- **LZ4 (115.5 сек)** — быстрее на 6.2 сек (5.1%)

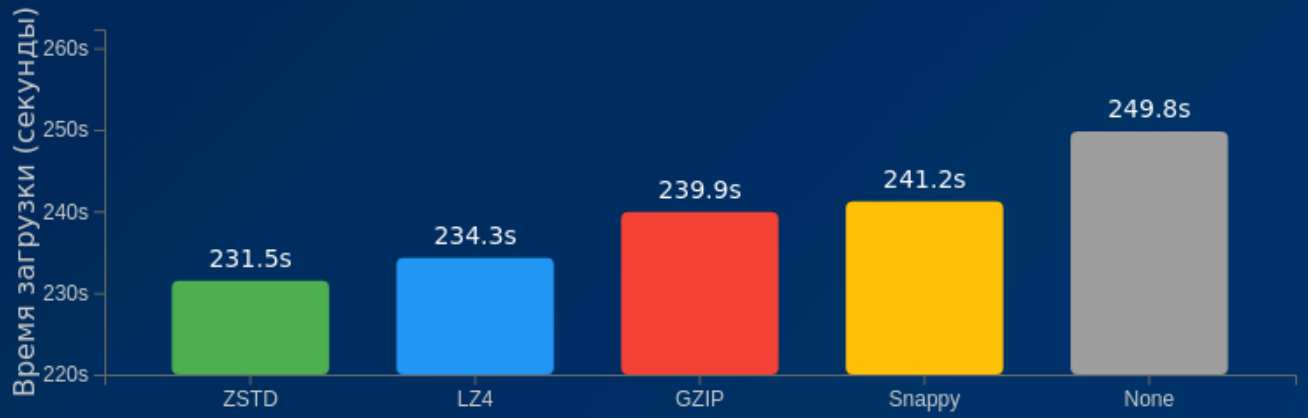
Наилучшее общее время:

- **ZSTD (347.9 сек)** — быстрее на 23.6 сек (6.4%)

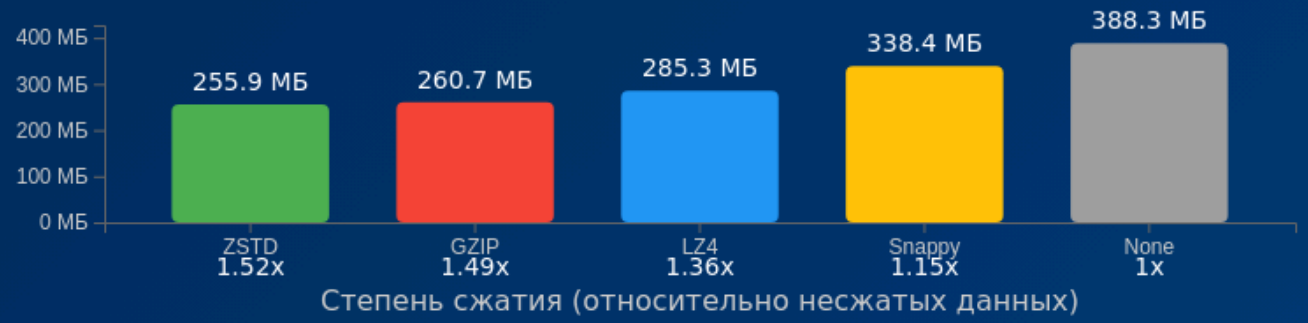
На основе усредненных данных трех тестовых прогонов

# Производительность загрузки данных (операции COPY)

## Время загрузки данных




## Эффективность сжатия и размер данных




## Ключевые выводы

### Лидер по скорости загрузки

ZSTD показал наилучшее время загрузки **231.5** секунд, что на **7.3%** быстрее несжатых данных.

### Лучшая степень сжатия

ZSTD также обеспечил наилучшую компрессию (**1.52x**), уменьшив размер данных до **255.9 МБ**.

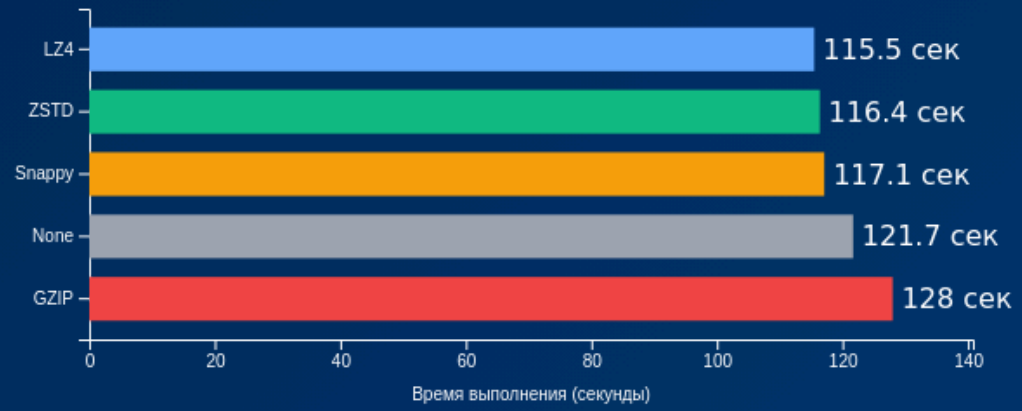
### Почему сжатие ускоряет загрузку?

Узким местом в тестовой конфигурации является пропускная способность сети. Уменьшение объема передаваемых данных дает больший выигрыш, чем затраты процессорного времени на компрессию.

Результаты усреднены по трем тестовым прогонам

# Анализ производительности запросов

## Время выполнения запросов




### Среднее время на запрос

| LZ4 | ZSTD | Snappy | None | GZIP |
|---|---|---|---|---|
| 5.25 с | 5.29 с | 5.32 с | 5.53 с | 5.82 с |
| (самый быстрый) | | | | (самый медленный) |

## Ключевые выводы

### LZ4 — лидер по скорости чтения

LZ4 выполнил все 22 запроса за 115.5 секунд, что делает его самым быстрым алгоритмом для аналитических операций. Этот алгоритм оптимизирован для максимальной скорости распаковки.

### Эффективный баланс ZSTD

ZSTD отстает от LZ4 всего на 0.9 секунды (116.4 с), но при этом обеспечивает гораздо лучшую степень сжатия. Это делает его привлекательным для смешанных нагрузок.

### Недостатки GZIP

GZIP показал самое медленное время выполнения запросов (128.0 с) несмотря на хорошую степень сжатия. Это связано с высокими требованиями к CPU для декомпрессии данных.

### Особенности без сжатия

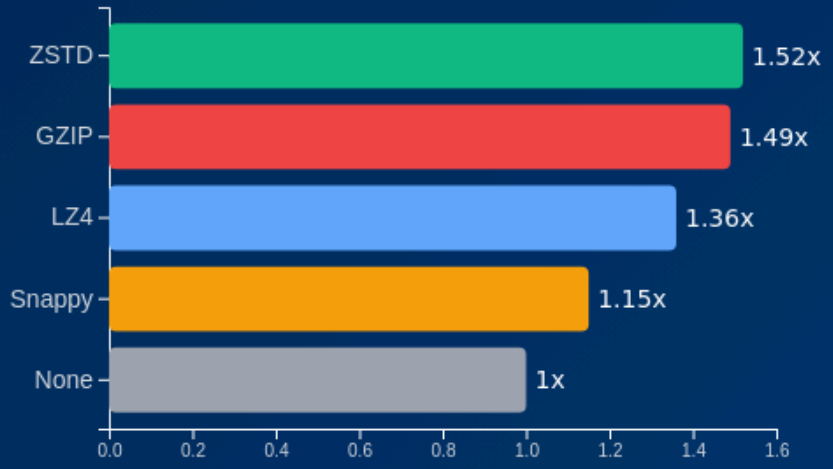
Вариант без сжатия (None) показал среднюю производительность (121.7 с), уступая быстрым алгоритмам, так как ему приходится читать с диска больший объем данных.

Выполнено 22 аналитических запроса для каждого алгоритма сжатия

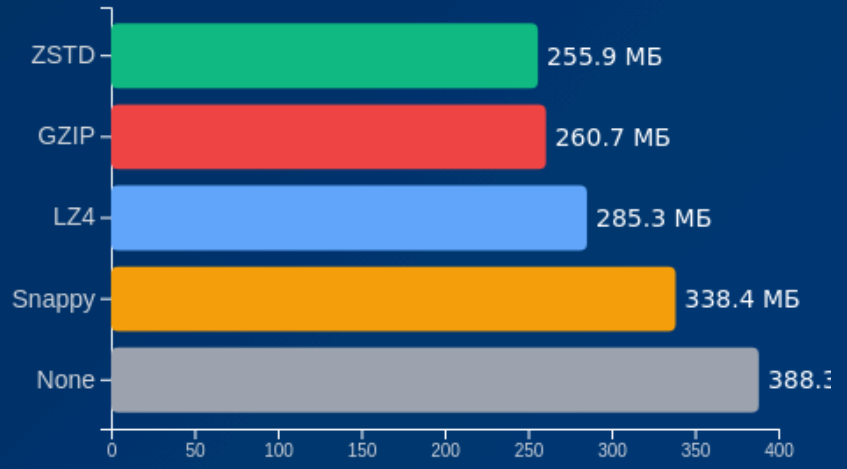
# Сравнение эффективности сжатия

Степень сжатия показывает, насколько эффективно алгоритм уменьшает размер данных относительно несжатого формата (None = 1.00x).

## Степень сжатия (выше = лучше)




## Фактический размер данных (МБ)




### Ключевые наблюдения

- ZSTD обеспечивает наилучшую степень сжатия (1.52x), позволяя хранить на 52% больше данных в том же объеме.
- GZIP немного уступает ZSTD с показателем 1.49x.
- Snappy показывает наименьшую эффективность сжатия (1.15x) среди тестируемых алгоритмов.

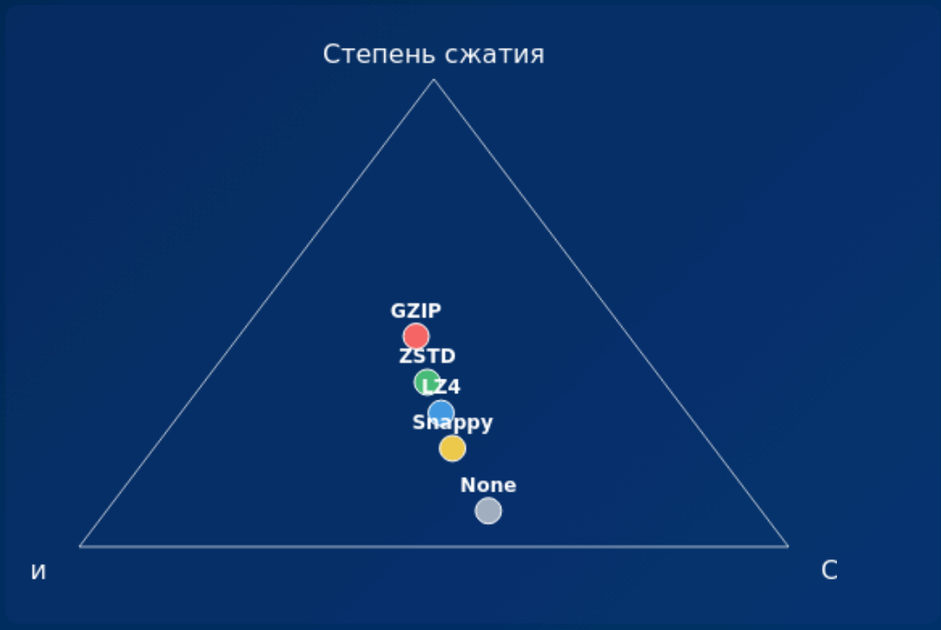
### Экономия места

- ZSTD: экономия **132.4 МБ**
- GZIP: экономия **127.6 МБ**
- LZ4: экономия **103.0 МБ**
- Snappy: экономия **49.9 МБ**

Данные усреднены по трем тестовым прогонам

# Компромиссы производительности

## ⚖ Баланс производительности




Положение алгоритма на диаграмме отражает его относительную эффективность по трем ключевым параметрам.
Чем ближе к вершине треугольника, тем лучше алгоритм по соответствующему параметру.

## Оптимальные сценарии использования

### ZSTD

**Лучший выбор**

Степень сжатия (1.52x) — Отлично

Скорость загрузки (231.5с) — Отлично

Скорость запросов (116.4с) — Очень хорошо

✓ Оптимально для: Смешанных нагрузок (частая запись и чтение)

### LZ4

**Лидер по чтению**

Степень сжатия (1.36x) — Хорошо

Скорость загрузки (234.3с) — Очень хорошо

Скорость запросов (115.5с) — Отлично

✓ Оптимально для: Нагрузок с интенсивным чтением данных

### GZIP

**Для архивации**

Степень сжатия (1.49x) — Очень хорошо

Скорость загрузки (239.9с) — Средне

Скорость запросов (128.0с) — Низкая

✓ Оптимально для: Архивации и холодных данных

### Snappy

Сжатие: 1.15x | Загрузка: 241.2с | Запросы: 117.1с

ℹ Компромиссный вариант без выраженных преимуществ

### None (без сжатия)

Сжатие: 1.00x | Загрузка: 249.8с | Запросы: 121.7с

ℹ Для систем с ограниченными ресурсами CPU

⚖ Данные усреднены по трем тестовым прогонам

# Ключевые выводы и рекомендации

## Сравнительный анализ алгоритмов

| ZSTD | LZ4 | GZIP | Snappy | None |
|---|---|---|---|---|
| Лучший универсальный выбор | Лидер по скорости чтения | Для архивации данных | Компромиссный вариант | Без сжатия |

### ZSTD — Лучший выбор

- Наилучшая степень сжатия: **1.52x**
- Самая высокая скорость загрузки: **231.5 с**
- Высокая скорость запросов: **116.4 с**
- Рекомендуется для смешанных нагрузок (запись и чтение)

### LZ4 — Для быстрого чтения

- Самая высокая скорость запросов: **115.5 с**
- Хорошая скорость загрузки: **234.3 с**
- Достаточная степень сжатия: **1.36x**
- Идеально для аналитических систем с приоритетом скорости запросов

### GZIP — Для архивации

- Хорошая степень сжатия: **1.49x**
- Средняя скорость загрузки: **239.9 с**
- Низкая скорость запросов: **128.0 с**
- Подходит для редко используемых данных и холодного хранения

### Snappy — Компромиссный вариант

Не показал выдающихся результатов ни в одной из метрик. Степень сжатия **1.15x**, время запросов **117.1 с**, время загрузки **241.2 с**.

### None — Без сжатия

Оправдан только в системах с крайне ограниченным CPU. Время запросов **121.7 с**, время загрузки **249.8 с**, объем данных **388.3 МБ**.

### Итоговая рекомендация

Для большинства сценариев использования Trino **рекомендуется ZSTD** как алгоритм, предлагающий наилучший баланс между степенью сжатия, скоростью записи и скоростью чтения.

Рекомендации основаны на усредненных результатах трех тестов

# Описание тестовой среды

## Аппаратное обеспечение

**Процессор:** MacBook Pro, 6 ядер

**Оперативная память:** 32 ГБ

## Среда виртуализации

**Контейнеризация:** Podman

**Выделенная память:** 24 ГБ

## Конфигурация Trino

**Координатор:** 1 узел

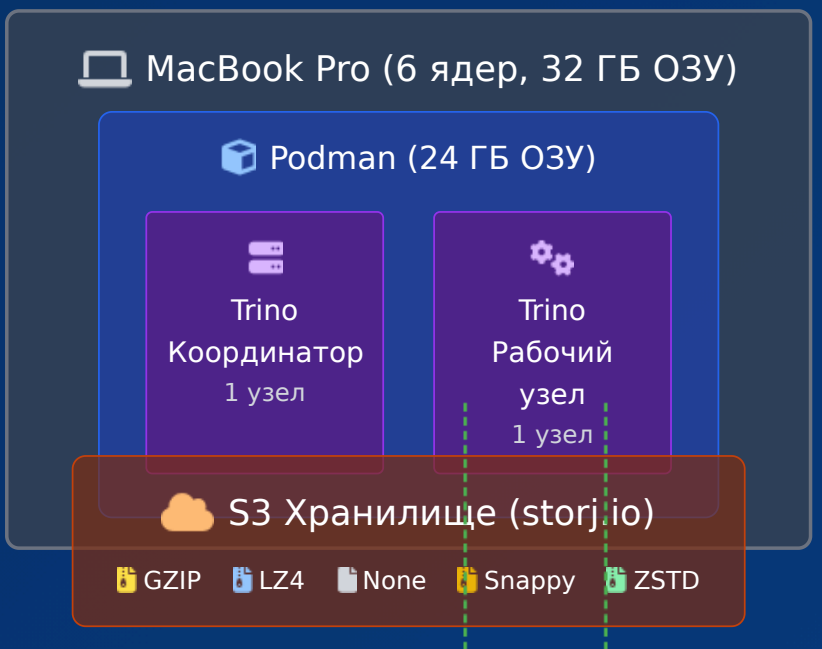
**Рабочий узел:** 1 узел

## Хранилище данных

**Тип хранилища:** Объектное хранилище S3

**Провайдер:** storj.io

## Архитектура тестовой среды




### Дополнительная информация

Все тесты проводились в контролируемой среде с использованием данных TPC-H масштабного фактора 1 (SF1). Результаты представляют средние значения по трем последовательным тестовым прогонам.