Московский государственный технический университет им. Н.Э. Баумана

Исследование производительности постреляционных баз данных с применением технологий тестирования

18.06.2025

Постановка задачи

- Проблема:
- Цель исследования:
- Объекты исследования:
- Предмет исследования:

- **Лавинообразный рост объемов данных:** Ежедневно генерируются огромные массивы информации (к 2025 г. – 175 зеттабайт).
- **Критичность выбора СУБД:** От выбора системы управления базами данных напрямую зависят производительность, масштабируемость и общая эффективность приложений.
- **Многообразие СУБД:** Наряду с традиционными реляционными СУБД (как PostgreSQL), широкое распространение получили NoSQL решения (документо-ориентированные как MongoDB, колоночные как Cassandra), каждое со своими архитектурными особенностями.
- **Высокая цена ошибки:** Неправильный выбор СУБД ведет к проблемам производительности, масштабирования и высоким эксплуатационным расходам.
- YCSB.

- Провести **комплексное сравнительное исследование производительности и масштабируемости** постреляционных СУБД (MongoDB, Cassandra) и реляционной СУБД PostgreSQL.
- Оценить поведение СУБД при **обработке больших объемов данных** (~12 ГБ).
- Использовать **стандартизированные методы тестирования** (бенчмарк YCSB).
- Выявить **сильные и слабые стороны** каждой СУБД в различных сценариях использования (типах нагрузок).

- **PostgreSQL** (реляционная СУБД)
- **MongoDB** (документо-ориентированная NoSQL СУБД)
- **Cassandra** (колоночная NoSQL СУБД)

- Показатели **производительности** (пропускная способность, время отклика) и **масштабируемости** при различных рабочих нагрузках (CRUD, сканирование), генерируемых YCSB.

ПОСТАНОВКА ЗАДАЧИ

ПРОБЛЕМА, ЦЕЛЬ И ОБЪЕКТЫ ИССЛЕДОВАНИЯ

ПРОБЛЕМА

ЛАВИНООБРАЗНЫЙ РОСТ ДАННЫХ

Ежедневно генерируются огромные массивы информации (к 2025 г. – 175 зеттабайт)

КРИТИЧНОСТЬ ВЫБОРА СУБД

От выбора системы напрямую зависят производительность, масштабируемость и эффективность приложений

МНОГООБРАЗИЕ СУБД

Наряду с реляционными СУБД, широкое распространение получили NoSQL решения, каждое со своими особенностями

ВЫСОКАЯ ЦЕНА ОШИБКИ

Неправильный выбор ведет к проблемам производительности и высоким эксплуатационным расходам

РЕШЕНИЕ: YCSB BENCHMARK

ЦЕЛЬ ИССЛЕДОВАНИЯ

- Провести **комплексное сравнительное исследование** производительности постреляционных и реляционной СУБД
- Оценить поведение СУБД при обработке **больших объемов данных** (~12 ГБ)
- Использовать **стандартизированные методы** тестирования (YCSB)
- Выявить **сильные и слабые стороны** каждой СУБД в различных сценариях

ОБЪЕКТЫ ИССЛЕДОВАНИЯ

PostgreSQL
Реляционная СУБД

MongoDB
Документоориентированная NoSQL

Cassandra
Колоночная NoSQL

ПРЕДМЕТ ИССЛЕДОВАНИЯ

ПРОПУСКНАЯ СПОСОБНОСТЬ
ops/sec

ВРЕМЯ ОТКЛИКА
Latency

МАСШТАБИРУЕМОСТЬ
Threads

РАБОЧИЕ НАГРУЗКИ
YCSB A-F

Показатели при различных рабочих нагрузках (CRUD, сканирование), генерируемых YCSB

ПЕРЕЧЕНЬ РЕШЁННЫХ ЗАДАЧ

ЭТАПЫ ИССЛЕДОВАНИЯ И РЕАЛИЗАЦИИ

ИССЛЕДОВАНИЕ И АНАЛИЗ

Теоретическое обоснование

1. Исследование **архитектурных решений** и подходов к обработке больших объемов данных в PostgreSQL, MongoDB и Cassandra
2. Сравнительный **анализ технологий тестирования** СУБД и обоснованный выбор универсального бенчмарка YCSB
3. Анализ рынка **инструментов бенчмаркинга** баз данных и исследование тенденций их использования

ПОДГОТОВКА ДАННЫХ

Настройка окружения

4. Анализ и выбор реального **JSON-датасета** научных публикаций объемом ~12 ГБ для тестирования производительности
5. Разработка специфических **стратегий подготовки** и загрузки данных для каждой СУБД с учетом их архитектурных особенностей
6. Создание и прецизионная настройка **изолированного тестового окружения** на виртуальной машине

ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ

Оптимизация и настройка

7. Техническая реализация **загрузки больших объемов** данных с преодолением совместимости инструментов
8. Комплексная оптимизация **конфигурационных файлов** PostgreSQL, MongoDB и Cassandra для высокопроизводительных нагрузок
9. Детальная конфигурация **YCSB** и разработка стандартизированных тестовых сценариев с широким диапазоном параллелизма

ТЕСТИРОВАНИЕ И АНАЛИЗ

Сбор и визуализация результатов

10. Проведение комплексной **серии тестов** производительности с обеспечением статистической достоверности и контролем условий
11. Разработка специализированного **Python-скрипта** для автоматизированного сбора и обработки многомерных результатов
12. Создание интерактивных **дашбордов в Apache Superset** для визуализации данных и формулирование практических рекомендаций

📈 ВСЕГО ВЫПОЛНЕНО: 12 задач | 4 этапа | 3 СУБД | 189 тестов

АРХИТЕКТУРНЫЕ РЕШЕНИЯ СУБД

Тезис

Архитектура СУБД определяет системные возможности

ВАЖНО

Реляционные СУБД дают консистентность, NoSQL — масштабируемость и гибкость

Анализ архитектурных особенностей

PostgreSQL (реляционная СУБД)

- Реляционная модель данных с определенной схемой
- MVCC (Multiversion Concurrency Control) для изоляции транзакций
- Полная поддержка ACID-свойств (Atomicity, Consistency, Isolation, Durability)
- JSON-поддержка для работы с полуструктурированными данными
- Расширяемость через пользовательские типы данных и функции

1

Cassandra (колоночная СУБД)

- Колоночная модель данных для эффективности определенных типов запросов
- Распределенная архитектура без единой точки отказа
- Линейная масштабируемость при добавлении узлов
- Настраиваемая консистентность для каждой операции
- Оптимизация для записи - архитектура, ориентированная на высокую производительность операций записи
- Сравнение подходов к обработке данных:
- Реляционный подход (PostgreSQL): строгая схема, нормализация, SQL, транзакционность
- Документоориентированный подход (MongoDB): гибкая схема, вложенные документы, горизонтальное масштабирование
- Колоночный подход (Cassandra): денормализация, широкие строки, распределение данных

2

MongoDB (документноориентированная СУБД)

- Колоночная модель данных для эффективности определенных типов запросов
- Распределенная архитектура без единой точки отказа
- Линейная масштабируемость при добавлении узлов
- Настраиваемая консистентность для каждой операции
- Оптимизация для записи - архитектура, ориентированная на высокую производительность операций записи
- Сравнение подходов к обработке данных:
- Реляционный подход (PostgreSQL): строгая схема, нормализация, SQL, транзакционность
- Документоориентированный подход (MongoDB): гибкая схема, вложенные документы, горизонтальное масштабирование
- Колоночный подход (Cassandra): денормализация, широкие строки, распределение данных

3

ТЕХНОЛОГИИ ТЕСТИРОВАНИЯ СУБД

Обзор инструментов бенчмаркинга

СПЕЦИАЛИЗИРОВАННЫЕ БЕНЧМАРКИ

pgBench

только для PostgreSQL

Cassandra-stress

специально для Cassandra, CQL операции

MongoDB Benchmarking Tools

mongoperf для тестирования дисковой подсистемы

Apache JMeter

универсальный, но с ограничениями для NoSQL

УНИВЕРСАЛЬНЫЕ БЕНЧМАРКИ

TPC Benchmarks (TPC-C, TPC-H)

индустриальные стандарты, сложны в настройке

Sysbench

скриптуемый, ограниченная поддержка NoSQL

Обоснование выбора YCSB

КРОСС-ПЛАТФОРМЕННОСТЬ 1

- Поддержка всех трех исследуемых СУБД
- Единый инструмент и единые метрики для сопоставимости результатов

СТАНДАРТИЗИРОВАННЫЕ РАБОЧИЕ НАГРУЗКИ 2

- Workload A: 50% чтение / 50% обновление (Update heavy)
- Workload B: 95% чтение / 5% обновление (Read heavy)
- Workload C: 100% чтение (Read only)
- Workload D: 95% чтение / 5% вставка (Read latest)
- Workload E: 95% сканирование / 5% вставка (Short ranges scan)
- Workload F: 50% чтение / 50% чтение-модификация-запись

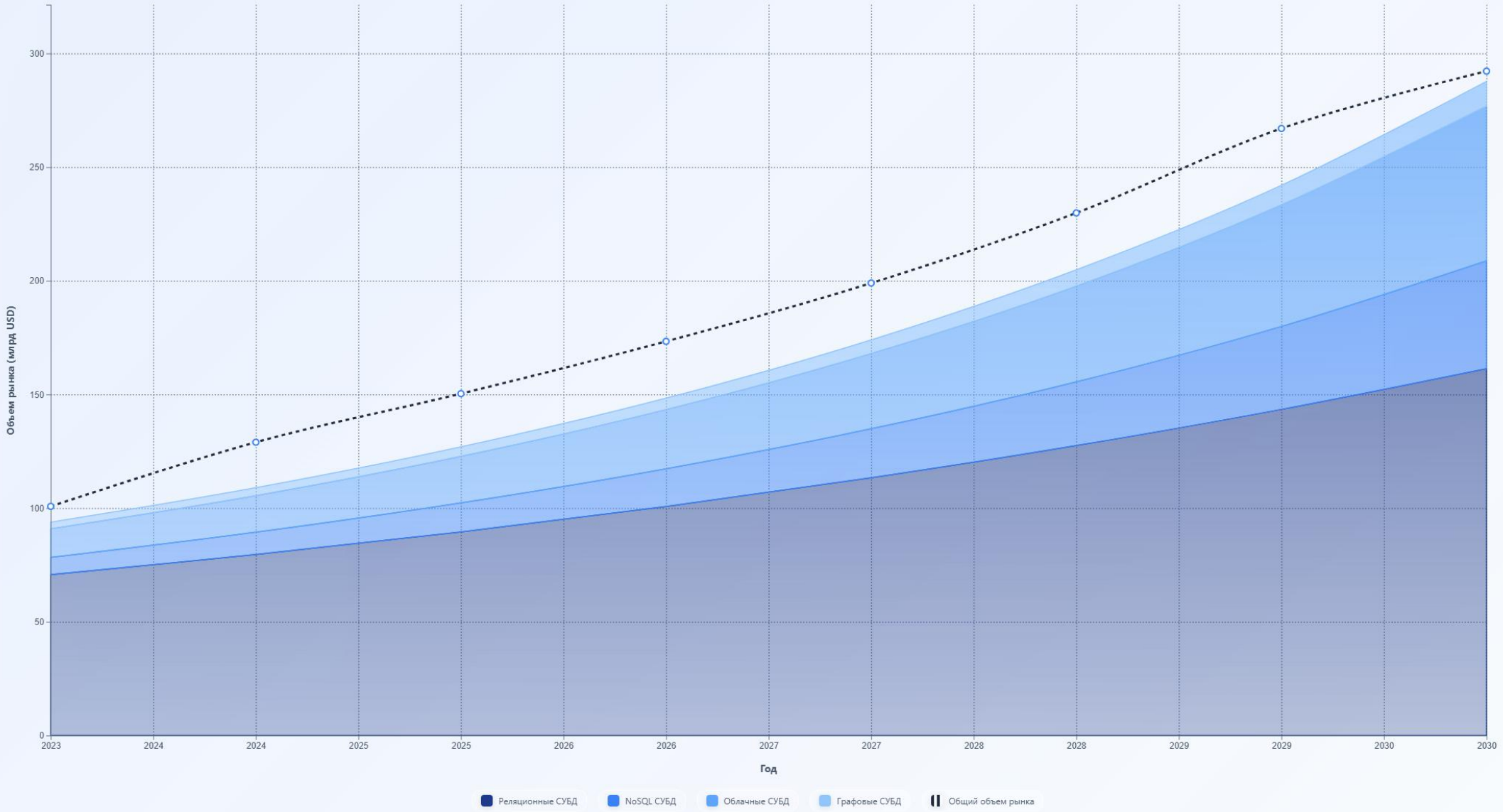
РЕЛЕВАНТНЫЕ МЕТРИКИ 3

- Пропускная способность (ops/sec)
- Задержки операций (среднее, перцентили P95, P99)
- Конфигурируемость параметров тестирования

АНАЛИЗ РЫНКА ИНСТРУМЕНТОВ БЕНЧМАРКИНГА

ГЛОБАЛЬНЫЕ ТЕНДЕНЦИИ И СТРУКТУРНЫЕ ИЗМЕНЕНИЯ РЫНКА СУБД

📈 Динамика роста рынка СУБД по типам (2023-2030)



🎯 КЛЮЧЕВЫЕ МЕТРИКИ

Общий рост рынка:
100.79 → 292.22 млрд USD

Темп роста (CAGR):
14.21% в год

NoSQL рост:
6-кратный

Облачные решения:
5.4x рост к 2030

📊 СТРУКТУРНЫЕ ИЗМЕНЕНИЯ

Реляционные СУБД
Снижение доли с 70% до 55%

NoSQL системы
Взрывной рост: 7.55 → 47.41 млрд USD

Облачные платформы
Удвоение доли рынка

Графовые БД
Новая ниша с 4x ростом

💡 ДРАЙВЕРЫ РОСТА

Цифровая трансформация предприятий

Экспоненциальный рост объемов данных

Потребность в масштабируемости и гибкости

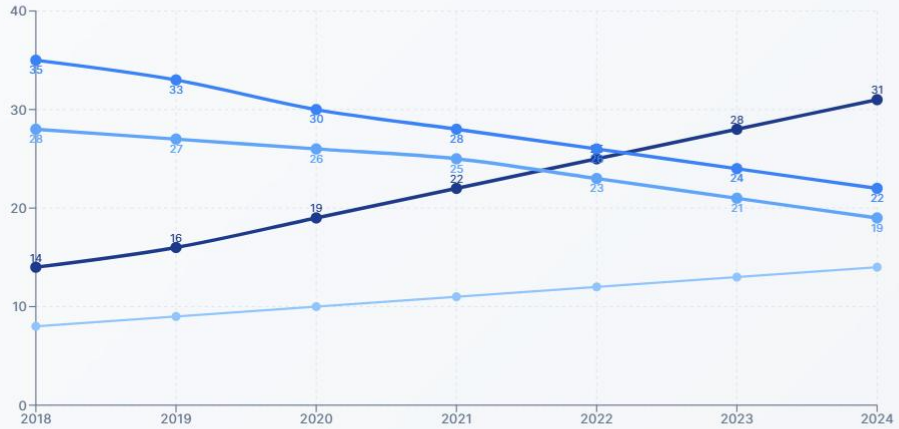
Развитие IoT и Big Data технологий

АНАЛИЗ РЫНКА ИНСТРУМЕНТОВ БЕНЧМАРКИНГА

Глобальные тенденции и структурные изменения рынка СУБД

🌍 Прогноз роста внедрения по регионам

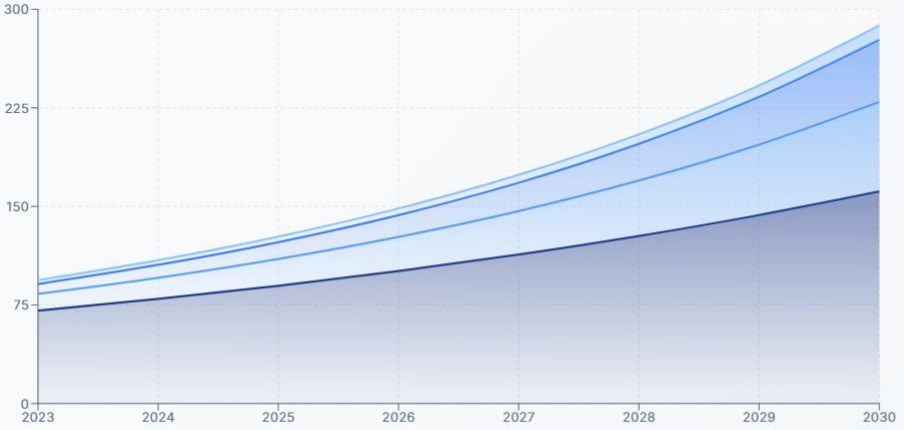
📈 Тренды популярности бенчмарков



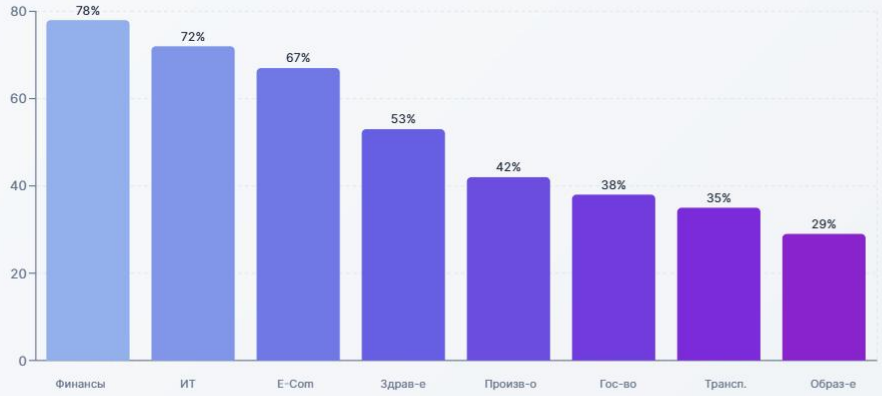
🎯 Рыночная доля

- TPC стандарты (42%)
- Open Source (YCSB) (31%)
- Вендорские (18%)
- Внутренние (9%)

💾 Динамика роста рынка СУБД по типам



🏢 Внедрение бенчмаркинга по отраслям



💰 **292.22B**
USD к 2030

📈 **+14.21%**
CAGR

🚀 **6x**
NoSQL рост

☁️ **5.4x**
Облачные

💡 Ключевые инсайты

- **31%** YCSB опережает TPC-H
- **2022** Точка смены парадигм
- **+77%** Азия лидирует по росту
- **78%** Финансы ведут внедрение
- **6x** Рост NoSQL к 2030

ВЫБОР И АНАЛИЗ ДАТАСЕТА

Характеристики и преимущества

Выбранный датасет

- Источник: Метаданные научных публикаций в JSON-формате
- Объём: ~12 ГБ
- Количество записей: 4,894,081
- Структура: Сложные вложенные JSON-документы с метаданными публикаций

Преимущества выбранного датасета

- Реалистичность: отражает типичную структуру данных современных приложений
- Сложность структуры: содержит вложенные объекты, массивы, различные типы данных
- Масштаб: достаточный объем для выявления характеристик производительности
- Универсальность: подходит для тестирования различных моделей данных

ВЫБОР И АНАЛИЗ ДАТАСЕТА

Детальное изучение данных позволяет выявить оптимальные архитектурные решения и избежать проблем производительности

Структура и особенности данных

Структура JSON-документов:

- Основные поля: title, year, authors, abstract, references
- Вложенные объекты: venue (место публикации), fos (области знаний)
- Массивы: authors, references, indexed_abstract
- Метаданные: идентификаторы, индексы, временные метки

Вызовы для различных СУБД

- MongoDB: прямая совместимость с JSON, сохранение структуры
- PostgreSQL: необходимость трансформации в реляционную структуру или использование JSONB
- Cassandra: требование денормализации и выравнивания вложенных структур

Значение для тестирования

- Проверка эффективности обработки реальных полуструктурированных данных
- Оценка производительности при различных паттернах доступа
- Выявление ограничений каждой архитектурной модели

СТРАТЕГИИ ПОДГОТОВКИ ДАННЫХ

Анализ архитектурных особенностей

PostgreSQL (реляционная СУБД)

- Реляционная модель данных с определенной схемой
- MVCC (Multiversion Concurrency Control) для изоляции транзакций
- Полная поддержка ACID-свойств (Atomicity, Consistency, Isolation, Durability)
- JSON-поддержка для работы с полуструктурированными данными
- Расширяемость через пользовательские типы данных и функции

1

Cassandra (колоночная СУБД)

- Колоночная модель данных для эффективности определенных типов запросов
- Распределенная архитектура без единой точки отказа
- Линейная масштабируемость при добавлении узлов
- Настраиваемая консистентность для каждой операции
- Оптимизация для записи - архитектура, ориентированная на высокую производительность операций записи
- Сравнение подходов к обработке данных:
- Реляционный подход (PostgreSQL): строгая схема, нормализация, SQL, транзакционность
- Документоориентированный подход (MongoDB): гибкая схема, вложенные документы, горизонтальное масштабирование
- Колоночный подход (Cassandra): денормализация, широкие строки, распределение данных

2

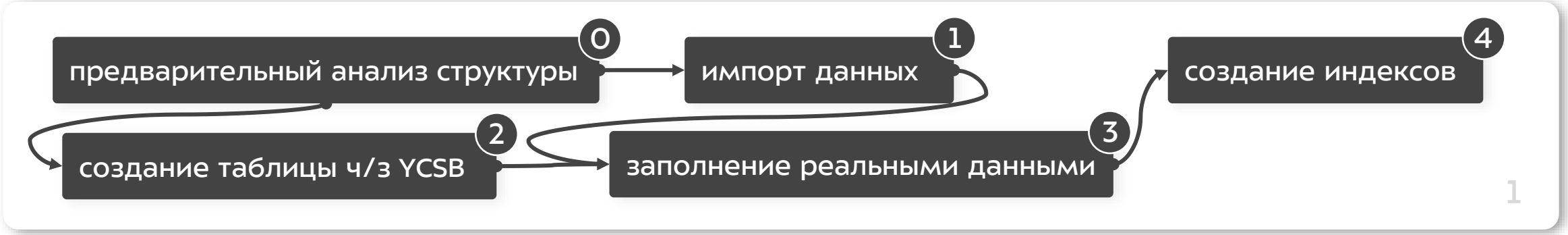
MongoDB (документноориентированная СУБД)

- Колоночная модель данных для эффективности определенных типов запросов
- Распределенная архитектура без единой точки отказа
- Линейная масштабируемость при добавлении узлов
- Настраиваемая консистентность для каждой операции
- Оптимизация для записи - архитектура, ориентированная на высокую производительность операций записи
- Сравнение подходов к обработке данных:
- Реляционный подход (PostgreSQL): строгая схема, нормализация, SQL, транзакционность
- Документоориентированный подход (MongoDB): гибкая схема, вложенные документы, горизонтальное масштабирование
- Колоночный подход (Cassandra): денормализация, широкие строки, распределение данных

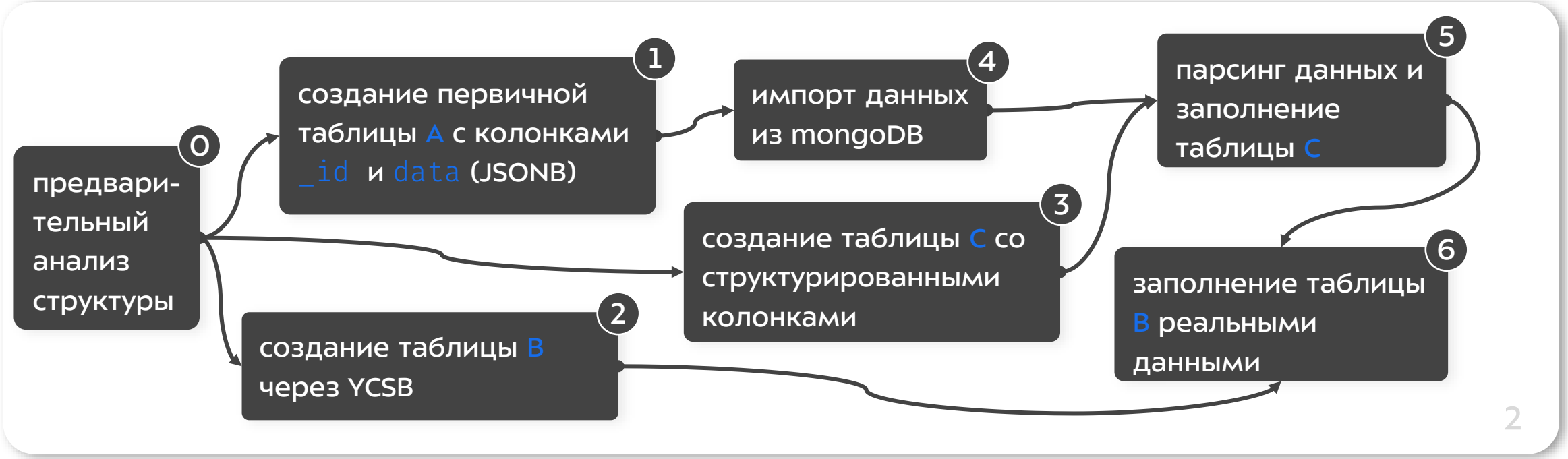
3

СТРАТЕГИИ ПОДГОТОВКИ ДАННЫХ

mongoDB, прямой ипорт JSON

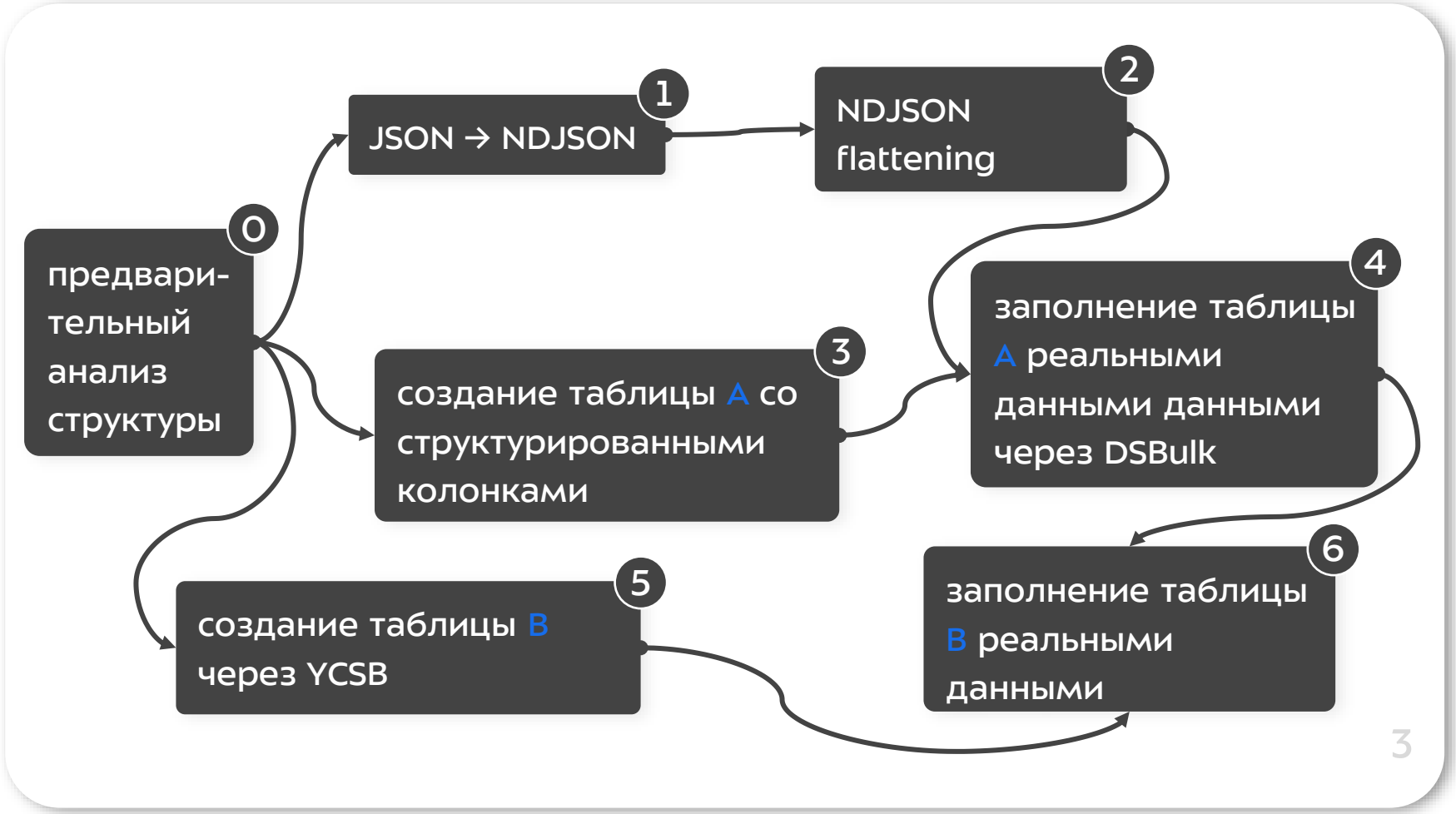


PostgreSQL, многоэтапный процесс



СТРАТЕГИИ ПОДГОТОВКИ ДАННЫХ

Cassandra, многоэтапный процесс



КЛЮЧЕВЫЕ РАЗЛИЧИЯ

MongoDB:

естественная совместимость

PostgreSQL:

баланс между структурированностью и гибкостью

Cassandra:

требование полной денормализации

АППАРАТНАЯ И ПРОГРАММНАЯ КОНФИГУРАЦИЯ

- Intel Core i9-12900H (16 виртуальных ядер, 2.9 ГГц)
- 24 ГБ DDR5 RAM, 4800Mhz
- NVMe SSD Western Digital S850NX (1512 ГБ)

ВИРТУАЛИЗАЦИЯ

- VMWare Workstation Pro 17

ПРОГРАММНОЕ ОКРУЖЕНИЕ

- Kubuntu 24.04 LTS
- MongoDB 8.0.6
- PostgreSQL 17.4
- Cassandra 4.1.8

ИНСТРУМЕНТЫ

- YCSB 0.17.0
- Python 3.9.21
- JDK 11.0.26

ТЕСТОВОЕ ОКРУЖЕНИЕ

ПРИНЦИПЫ ОБЕСПЕЧЕНИЯ ОБЪЕКТИВНОСТИ:

- Идентичное окружение для всех тестов
- "Холодный" старт перед каждым тестом
- Контролируемые условия и изоляция процессов

ОБЕСПЕЧЕНИЕ НАДЁЖНОСТИ РЕЗУЛЬТАТОВ

- Трехкратное повторение каждого теста
- Статистическая обработка результатов
- Контроль внешних факторов и системных ресурсов
- Документирование условий проведения тестов

СТАНДАРТИЗАЦИЯ ПРОЦЕДУР

- Одинаковая последовательность тестирования
- Фиксированные интервалы между тестами
- Автоматизация сбора метрик и результатов

Техническая реализация загрузки

Основные технические вызовы:

- Различия в форматах данных между СУБД
- Совместимость драйверов и API
- Оптимизация процесса загрузки для больших объемов
- Обеспечение целостности данных при трансформации

Результаты технической реализации:

- Успешная загрузка 4,894,081 записей во все три СУБД
- Сохранение целостности данных при всех трансформациях
- Подготовка единообразных тестовых таблиц

MongoDB

- Прямое использование mongoimport для JSON
- Настройка параметров импорта для оптимизации скорости
- Создание индексов после загрузки для минимизации времени

PostgreSQL

- Пакетная загрузка с использованием psycopg2-binary
- Оптимизация SQL-запросов для трансформации JSON в реляционную структуру
- Использование COPY для быстрой загрузки больших объемов

Cassandra

- Предобработка данных (JSON → NDJSON, выравнивание, валидация данных)
- Конфигурация DSBulk (создание конфигурационных файлов, настройка маппинга полей м/уисточником и целевой схемой, оптимизация параметров загрузки)

ОПТИМИЗАЦИЯ КОНФИГУРАЦИЙ СУБД

PostgreSQL
Реляционная СУБД

postgresql.conf

```
# Увеличение подключений для высокого параллелизма
max_connections = 500 # вместо 100
# аутентификация для тестирования
local all all trust
host all all 127.0.0.1/32 trust
```

MongoDB
Документоориентированная СУБД

mongod.conf

```
cacheSizeGB: 16 # увелечение кэша
maxIncomingConnections: 1000
wiredTigerConcurrentReadTransactions: 1000
wiredTigerConcurrentWriteTransactions: 1000
```

Cassandra
Колоночная СУБД

cassandra.yaml

```
# Увеличение подключений для высокого параллелизма
concurrent_reads: 256 # вместо 32
concurrent_writes: 256 # вместо 32
concurrent_materialized_view_writes: 256 # вместо 32
```

РЕЗУЛЬТАТ ОПТИМИЗАЦИИ

- Минимизация узких мест при высоком параллелизме
- Стабильная производительность на всех уровнях нагрузки
- Оптимальное использование аппаратных ресурсов

КОНФИГУРАЦИЯ YCSB

Параметры тестирования производительности

ОСНОВНЫЕ ПАРАМЕТРЫ

Record Count	4,894,081
Operation Count	4,894,081
Threads	[4,8,16,32,64,128,256]
Распределение	uniform, Zipfian, latest
Повторения	3x

ДРАЙВЕРЫ СУБД

MongoDB
Стандартный YCSB драйвер

Cassandra
CQL-драйвер

PostgreSQL
JDBC-драйвер

РАБОЧИЕ НАГРУЗКИ

Workload A
Baseline
50% read / 50% update

Workload B
Read-Heavy
95% read / 5% update

Workload C
Read-Only
100% read

Workload D
Read-Latest
95% read latest / 5% insert

Workload E
Scan-Heavy
95% scan / 5% insert

Workload F
Read-Modify-Write
50% read / 50% RMW

ПРОЦЕДУРА ТЕСТИРОВАНИЯ

- Трехкратное повторение каждой комбинации
- Перезапуск систем между тестами
- Продолжительность: ≥120 минут
- Идентичные параметры для всех СУБД
- Единая методология измерения
- Стандартизированная отчетность

КЛЮЧЕВЫЕ ОСОБЕННОСТИ

- Zipfian распределение имитирует "горячие" точки
- Минимизация влияния кэширования
- Обеспечение сопоставимости результатов
- Полный спектр нагрузок

ПРОВЕДЕНИЕ ТЕСТОВ

Методология и контроль качества

ПРОТОКОЛ ПРОВЕДЕНИЯ

Последовательность выполнения:

- Последовательное выполнение всех workload'ов (A-F)
- Для каждого workload'а: тестирование с 7 различными уровнями параллелизма
- 21 конфигурация на СУБД × 3 повторения = 63 теста на СУБД
- Общее количество тестов: 189 для трех СУБД

Контроль условий:

"Холодный" старт
Каждого теста

Мониторинг
Системных ресурсов

Фиксация условий
Состояние системы

Проверка корректности
Выполнения операций

СТАТИСТИЧЕСКАЯ ДОСТОВЕРНОСТЬ

Обеспечение надежности:

Трехкратное повторение
каждого теста

Расчет средних значений
и стандартных отклонений

Исключение выбросов
при анализе данных

Доверительные интервалы
для ключевых метрик

Контрольные проверки:

- Валидация результатов между повторными запусками
- Проверка согласованности метрик YCSB
- Анализ системных логов на предмет ошибок
- Верификация целостности данных после тестов

Документирование процесса:

ДЕТАЛЬНЫЕ ЛОГИ
Каждого запуска

КОНФИГУРАЦИИ
Воспроизводимость

ВРЕМЕННЫЕ МЕТКИ
Условия проведения

ПРОМЕЖУТОЧНЫЕ
Результаты

189

ОБЩЕЕ КОЛИЧЕСТВО ТЕСТОВ

`3 СУБД × 6 workloads × 7 threads × 3 повторения`

63

ТЕСТОВ НА СУБД

`21 конфигурация × 3 повторения`

Собираемые метрики:

Пропускная способность (ops/sec)

Задержки (среднее, мин, макс, P95, P99)

Системные метрики (CPU, память)

Дисковый I/O и сеть

1. Workload A-F
2. 7 уровней threads
3. 3 повторения
4. Анализ результатов

АВТОМАТИЗАЦИЯ СБОРА РЕЗУЛЬТАТОВ

PYTHON-СКРИПТ ДЛЯ ОБРАБОТКИ ДАННЫХ

parse_ycsb.py

Автоматизированный парсинг результатов YCSB тестирования

Функциональность скрипта:

- **Автоматическое извлечение** метрик из отчетов YCSB
- **Парсинг различных форматов** выходных данных
- **Агрегация результатов** по СУБД, workload'ам и потокам
- **Расчет статистических показателей** (среднее, медиана, стандартное отклонение)

Структура выходных данных:

CSV-файлы
Агрегированные результаты для каждой СУБД

Сводные таблицы
Данные для сравнительного анализа

JSON-файлы
Интеграция с системами визуализации

Извлекаемые метрики:

ОБЩИЕ МЕТРИКИ

- Время выполнения (RunTime)
- Пропускная способность (Throughput, ops/sec)

МЕТРИКИ СБОРКИ МУСОРА

- G1 Young Generation (количество, время, %)
- G1 Old Generation (количество, время, %)

ОПЕРАЦИИ YCSB

- READ, UPDATE, INSERT, SCAN
- READ-MODIFY-WRITE, CLEANUP

ЗАДЕРЖКИ ОПЕРАЦИЙ

- Среднее значение (AverageLatency)
- Минимум и максимум
- Перцентили: P95, P99

ВСЕГО МЕТРИК	ФОРМАТ ВЫВОДА
60+ параметров	CSV структура
АВТОМАТИЗАЦИЯ	**ТОЧНОСТЬ**
100% процесса	Микросекунды (µs)

ВИЗУАЛИЗАЦИЯ И РЕКОМЕНДАЦИИ

APACHE SUPERSET ДАШБОРДЫ

Apache Superset

Платформа для интерактивной визуализации данных

ТИПЫ СОЗДАННЫХ ВИЗУАЛИЗАЦИЙ:

- Линейные графики пропускной способности по количеству потоков
- Гистограммы сравнения производительности по workload'ам
- Тепловые карты задержек операций
- Сравнительные диаграммы масштабируемости СУБД

КЛЮЧЕВЫЕ ДАШБОРДЫ:

"Обзор производительности"
Сравнение всех СУБД по основным метрикам

"Анализ масштабируемости"
Поведение при увеличении параллелизма

"Детализация по операциям"
Задержки для различных типов операций

"Workload-специфичные паттерны"
Оптимальные сценарии для каждой СУБД

РЕКОМЕНДАЦИИ ПО ВЫБОРУ СУБД:

MongoDB

ОПТИМАЛЬНЫЕ СЦЕНАРИИ:

- Приложения с гибкой схемой данных
- Системы с преобладанием операций чтения
- Сложные вложенные документы

WORKLOAD D	WORKLOAD E
47.7k ops/sec	23.9k ops/sec

Cassandra

СИЛЬНЫЕ СТОРОНЫ:

- Системы с высокой интенсивностью записи
- Лидер по пропускной способности Workload A-C
- Низкие задержки операций обновления

WORKLOAD A-C	WORKLOAD E
24.2-29.2k ops/sec	1.81k ops/sec

ОГРАНИЧЕНИЯ:
Низкая эффективность сканирования

PostgreSQL

УНИВЕРСАЛЬНОСТЬ:

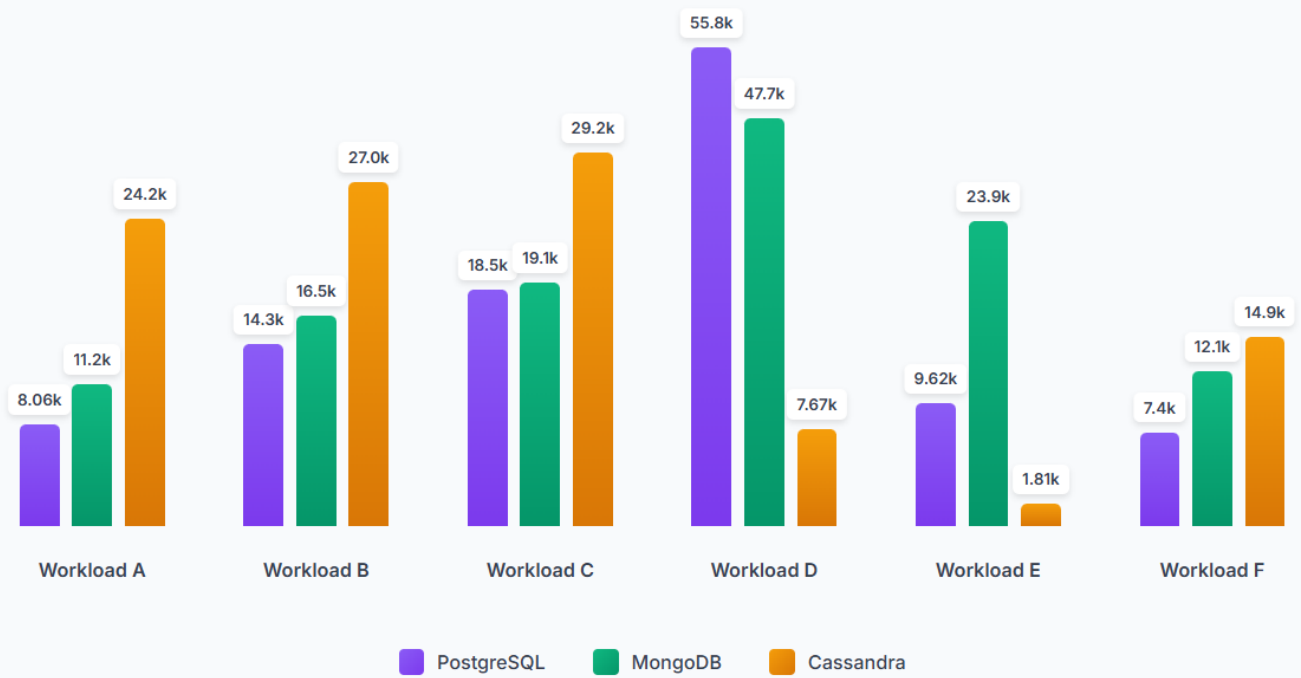
- Исключительная производительность Workload D
- Полноценная ACID-совместимость
- Нелинейная масштабируемость с пиком при 128 потоков

WORKLOAD D	WORKLOAD E
55.8k ops/sec	9.62k ops/sec

ОСНОВНЫЕ РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ

СТОЛБЧАТАЯ ДИАГРАММА | РАДАРНЫЕ ДИАГРАММЫ



ЛИДЕРЫ ПО WORKLOAD'АМ

A, B, C	WORKLOAD D
Cassandra	PostgreSQL
WORKLOAD E	WORKLOAD F
MongoDB	Cassandra

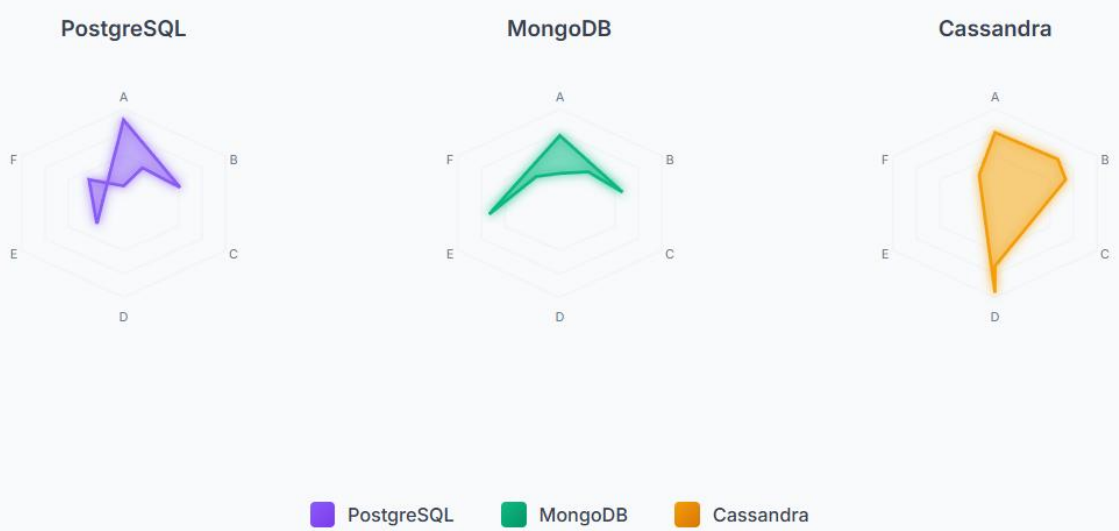
КЛЮЧЕВЫЕ ВЫВОДЫ

- → **Cassandra** лидирует в стандартных CRUD-операциях (A, B, C)
- → **PostgreSQL** доминирует в Workload D (чтение последних записей)
- → **MongoDB** эффективен для операций сканирования (Workload E)
- → Максимальная разница в производительности достигает **30x**

ОСНОВНЫЕ РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ

СТОЛБЧАТАЯ ДИАГРАММА | **РАДАРНЫЕ ДИАГРАММЫ**



ЛИДЕРЫ ПО WORKLOAD'АМ

A, B, C **Cassandra**	WORKLOAD D **PostgreSQL**
WORKLOAD E **MongoDB**	WORKLOAD F **Cassandra**

КЛЮЧЕВЫЕ ВЫВОДЫ

→ **Cassandra** лидирует в стандартных CRUD-операциях (A, B, C)

→ **PostgreSQL** доминирует в Workload D (чтение последних записей)

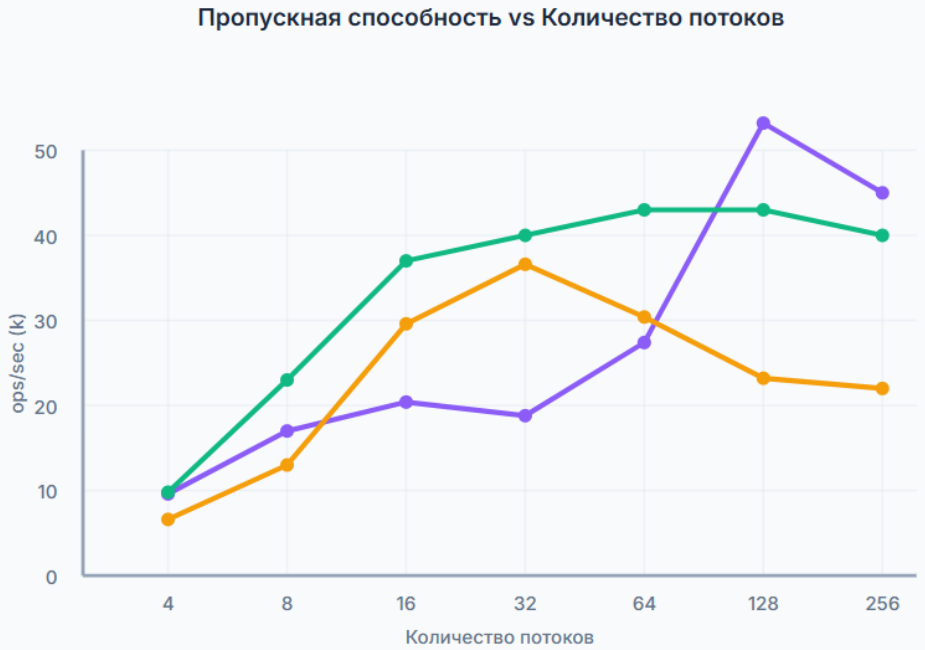
→ **MongoDB** эффективен для операций сканирования (Workload E)

→ Максимальная разница в производительности достигает **30x**

МАСШТАБИРУЕМОСТЬ И ЗАДЕРЖКИ

ПОВЕДЕНИЕ ПРИ УВЕЛИЧЕНИИ ПАРАЛЛЕЛИЗМА

POSTGRESQL MONGODB CASSANDRA



ПИКОВЫЕ ЗНАЧЕНИЯ

POSTGRESQL	32.8k @ 128 потоков
MONGODB	27.0k @ 64-128 потоков
CASSANDRA	23.2k @ 32 потока

ХАРАКТЕРИСТИКИ МАСШТАБИРУЕМОСТИ

→ **MongoDB:** плавный рост до 27.0k ops/sec при 64-128 потоках

→ **Cassandra:** пик 23.2k ops/sec при 32 потоках, затем снижение

→ **PostgreSQL:** нелинейное поведение с максимумом 32.8k ops/sec при 128 потоков

ПРАКТИЧЕСКАЯ ЗНАЧИМОСТЬ

→ Не существует **универсального решения** для всех типов нагрузок

→ Выбор СУБД должен основываться на **конкретных требованиях** приложения

→ Важность **предварительного тестирования** на реальных данных

ЗАКЛЮЧЕНИЕ

ВКЛАД ИССЛЕДОВАНИЯ И ПЕРСПЕКТИВЫ

НАУЧНЫЙ ВКЛАД

- **Комплексное сравнение** трех различных архитектурных подходов к СУБД
- Использование **реального большого датасета** (~12 ГБ) вместо синтетических данных
- **Стандартизированная методология** тестирования с обеспечением воспроизводимости
- Детальный анализ поведения при **различных уровнях параллелизма**

189
ПРОВЕДЕНО ТЕСТОВ

3 СУБД	**6** WORKLOADS
7 УРОВНЕЙ ПОТОКОВ	**12+** ГБ ДАННЫХ

ПРАКТИЧЕСКАЯ ЦЕННОСТЬ

- **Обоснованные рекомендации** по выбору СУБД для конкретных сценариев
- Выявление **оптимальных конфигураций** для каждой системы
- Понимание **ограничений и особенностей** масштабирования

30x
МАКСИМАЛЬНАЯ РАЗНИЦА ПРОИЗВОДИТЕЛЬНОСТИ

Cassandra CRUD ЛИДЕР	**PostgreSQL** WORKLOAD D
MongoDB СКАНИРОВАНИЕ	**100%** ВОСПРОИЗВОДИМОСТЬ

НАПРАВЛЕНИЯ ДАЛЬНЕЙШИХ ИССЛЕДОВАНИЙ

- → Тестирование в **распределенных конфигурациях**
- → Исследование влияния **различных типов данных**
- → Анализ поведения при **отказах и восстановлении**
- → Оценка **энергоэффективности** различных СУБД

∞
ПОТЕНЦИАЛ РАЗВИТИЯ

Кластеры РАСПРЕДЕЛЕННОСТЬ	**Fault Tolerance** ОТКАЗОУСТОЙЧИВОСТЬ
Green IT ЭКОЛОГИЧНОСТЬ	**Big Data** МАСШТАБ ДАННЫХ

ЗАКЛЮЧЕНИЕ

ВКЛАД ИССЛЕДОВАНИЯ И ПЕРСПЕКТИВЫ

НАУЧНЫЙ ВКЛАД

- ✓ **Комплексное сравнение** трех различных архитектурных подходов к СУБД
- ✓ Использование **реального большого датасета** (~12 ГБ) вместо синтетических данных
- ✓ **Стандартизированная методология** тестирования с обеспечением воспроизводимости
- ✓ Детальный анализ поведения при **различных уровнях параллелизма**

189
ПРОВЕДЕНО ТЕСТОВ

3 СУБД	**6** WORKLOADS
7 УРОВНЕЙ ПОТОКОВ	**~12** ГБ ДАННЫХ

ПРАКТИЧЕСКАЯ ЦЕННОСТЬ

- ✓ **Обоснованные рекомендации** по выбору СУБД для конкретных сценариев
- ✓ Выявление **оптимальных конфигураций** для каждой системы
- ✓ Понимание **ограничений и особенностей** масштабирования

30x
МАКСИМАЛЬНАЯ РАЗНИЦА ПРОИЗВОДИТЕЛЬНОСТИ

Cassandra CRUD ЛИДЕР	**PostgreSQL** WORKLOAD D
MongoDB СКАНИРОВАНИЕ	**100%** ВОСПРОИЗВОДИМОСТЬ

НАПРАВЛЕНИЯ ДАЛЬНЕЙШИХ ИССЛЕДОВАНИЙ

- → Тестирование в **распределенных конфигурациях**
- → Исследование влияния **различных типов данных**
- → Анализ поведения при **отказах и восстановлении**
- → Оценка **энергоэффективности** различных СУБД

∞
ПОТЕНЦИАЛ РАЗВИТИЯ

Кластеры РАСПРЕДЕЛЕННОСТЬ	**Fault Tolerance** ОТКАЗОУСТОЙЧИВОСТЬ
Green IT ЭКОЛОГИЧНОСТЬ	**Big Data** МАСШТАБ ДАННЫХ

Спасибо!

Контакты