

FIREBIRD 🐦 CONF 2025

Кирилл Куликов

**Firebird в квантовых вычислениях:
как СУБД может стать мостом между
классическими и quantum-системами**

КТО Я? КАК ОКАЗАЛСЯ ЗДЕСЬ?

Студент 2 курса направления 'Компьютерные системы и комплексы'.
Начинающий специалист, энтузиаст квантовых вычислений. Причина моего присутствия - желание опробовать себя спикером и показать на обзор общества свой доклад.

- Студент-практикант
- Дилетант-энтузиаст
- Просто молодой парень

ИСТОРИЯ ИДЕИ И ЕЕ РОЖДЕНИЕ

Учебный проект и первая мысль. Работая над своим проектом 'Плантации бананов в Индии', я столкнулся с парадоксом:

→ Простые операции ≠ Быстрые операции

Оказалось, что SQL — это язык, где: `SELECT * FROM users` отрабатывает мгновенно, а `SELECT users.name, orders.total FROM users JOIN orders ON...` может превратиться в многочасовой кошмар

```
SELECT e.First_name, e.Last_name, e.Position, e.Start_date
FROM Employee e
JOIN Plantation p ON e.ID_plantation = p.ID_plantation
WHERE p.Name = 'Plantation Name';
```

← Казалось бы простой запрос, но при 1М+ записей - тормозит.

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ? ПРИНЦИП РАБОТЫ И ПРОБЛЕМЫ.

Что такое СУБД?

→ “Умный склад информации”

Принцип работы:

- Хранит данные в таблице
- Позволяет искать, сортировать, изменять данные с помощью языка SQL
- Данные хранятся на диске → загружаются в оперативную память → обрабатываются процессором.

Проблемы классических СУБД:

JOIN-операции тормозят на больших объемах

Чем больше данных, тем медленнее поиск

СУТЬ КВАНТОВЫХ ВЫЧИСЛЕНИЙ. БУСТ ЗА СЧЕТ ПРИНЦИПОВ КВАНТОВОЙ МЕХАНИКИ

Квантовые вычисления используют принципы квантовой механики, для обработки информации. В отличие от классических битов, которые могут быть только 0 или 1, квантовые биты (кубиты) могут находиться в суперпозиции обоих состояний одновременно.

Преимущество квантовых компьютеров – их способность выполнять операции над всеми возможными состояниями одновременно. Экспоненциальный прирост вычислительной мощности для конкретных алгоритмов, недоступный для классических систем. Квантовые компьютеры не заменят классические, а будут дополнять их, решая специфические, особо сложные задачи!

Суперпозиция

Кубит может быть в нескольких состояниях одновременно, что позволяет обрабатывать несколько вариантов сразу.

Запутанность

Состояния двух или более кубитов становятся взаимозависимыми, даже если они физически разделены.

Квантовый параллелизм

Способность выполнять операции над всеми возможными входными данными одновременно.

АЛГОРИТМ ГРОВЕРА: КВАНТОВЫЙ ПОИСК В БД

Представьте огромную базу данных, где каждый элемент уникален, и нет никакого индекса. Классический компьютер должен был бы перебрать каждый элемент по очереди, пока не найдет нужный. Квантовый алгоритм Гровера, используя суперпозицию, позволяет «проверить» все элементы одновременно, значительно сокращая время поиска.

1 Быстрый поиск

$O(\sqrt{N})$ операций вместо $O(N)$ для неструктурированных данных.

2 Квадратичное ускорение

Значительный выигрыш по сравнению с классическими методами.

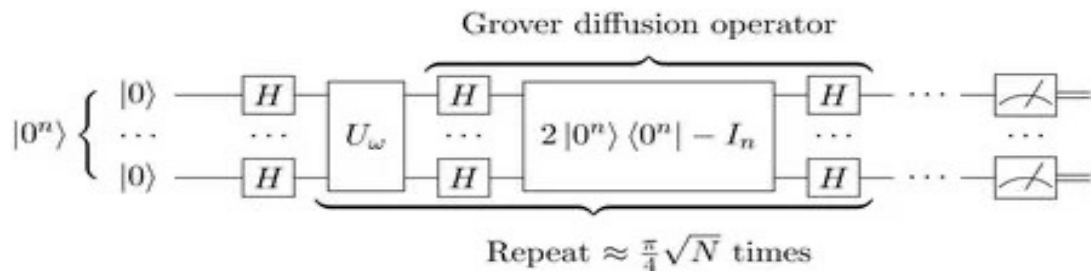
3 Применимость

Идеален для поиска в больших, неупорядоченных наборах данных.

КАК РАБОТАЕТ АЛГОРИТМ ГРОВЕРА?

Алгоритм Гровера использует амплитудное усиление — квантовый процесс, который увеличивает вероятность обнаружения целевого элемента за счёт "поворота" состояния системы в нужном направлении.

Алгоритм Гровера итеративно применяет квантовый оператор (набор вентилей), постепенно усиливая вероятность обнаружения искомого элемента при измерении.



Инициализация

Создание суперпозиции всех возможных состояний.

Амплитудное усиление

Применение оператора Гровера для увеличения вероятности целевого состояния.

Итерации

Повторение процесса для достижения максимальной вероятности.

Измерение

Измерение кубитов, что приводит к коллапсу суперпозиции и получению результата.

ПОИСК И JOIN БЕЗ ИНДЕКСОВ: НЕДОСТАТКИ И ПРОБЛЕМА

Недостатки поиска

Отсутствие индексов → полный перебор таблицы

Высокая стоимость I/O → Количество операций ввода-вывода

Замедление при росте данных → Объем данных

JOIN без индексов: когда это становится проблемой?

Nested loops join → Производительность катастрофически падает

Hash join → Если одна из таблиц мала, чтобы поместиться в память

Sort-merge join → Сначала сортируются, потом сливаются

КЕЙС-ПРИМЕР

Индексы увеличивают объем данных в 2-3 раза

```
SELECT u.user_id, e.action_time, e.event_data
FROM users u
JOIN events e ON u.user_id = e.user_id -- JOIN без индекса!
WHERE u.country_code = 'US'
AND e.action_time BETWEEN '2024-01-01' AND '2024-12-31';
```

Как теоретически применить алгоритм Гровера?

```
SELECT user_id, action_time, event_data
FROM events
WHERE action_time BETWEEN '2024-01-01' AND '2024-12-31';
```

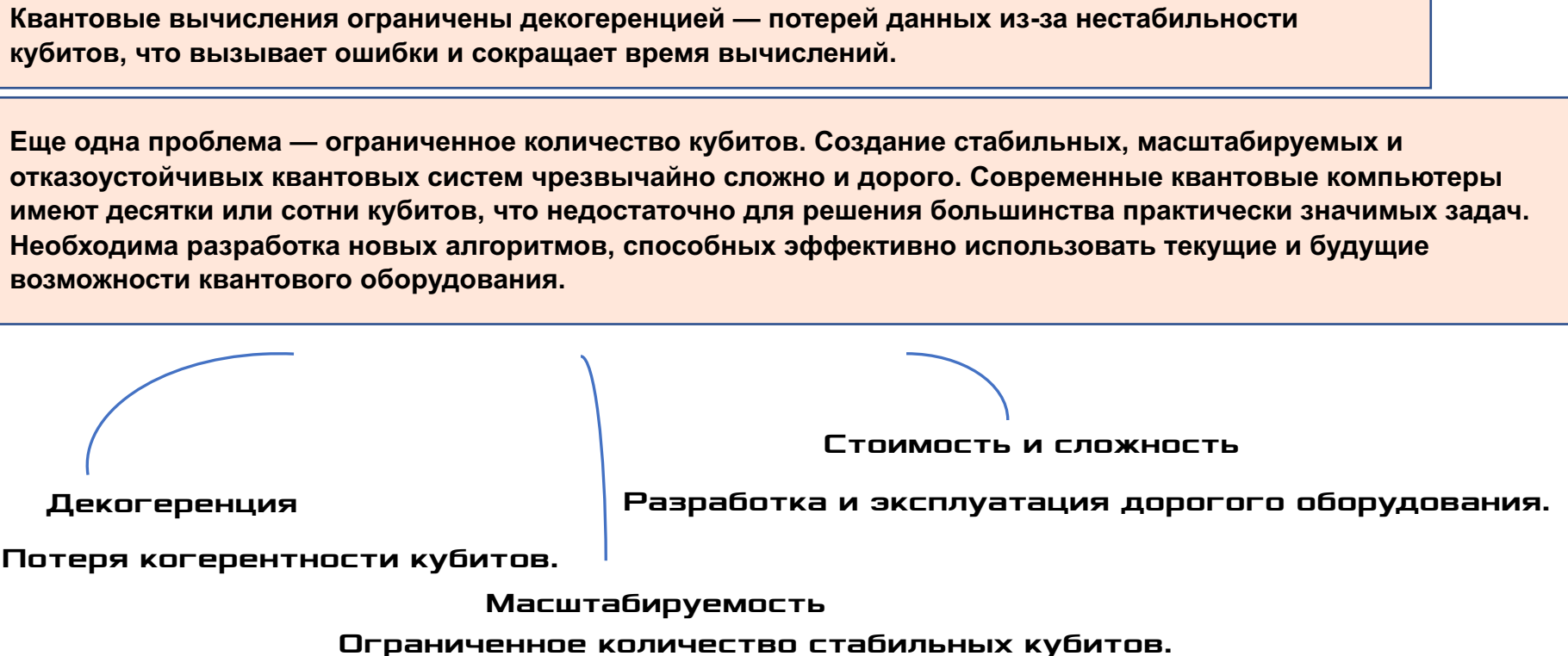
Классический поиск: $10 \text{ млн} \times 1 \text{ млн} = 10^{13}$ операций.

Гровер: $\sqrt{10^6} = 1000$ операций для каждого user_id (итого $\sim 10 \text{ млн} \times 1000 = 10^{10}$ — но с квантовым параллелизмом).

ПРОБЛЕМЫ И ВЫЗОВЫ КВАНТОВЫХ ВЫЧИСЛЕНИЙ

Квантовые вычисления ограничены декогеренцией — потерей данных из-за нестабильности кубитов, что вызывает ошибки и сокращает время вычислений.

Еще одна проблема — ограниченное количество кубитов. Создание стабильных, масштабируемых и отказоустойчивых квантовых систем чрезвычайно сложно и дорого. Современные квантовые компьютеры имеют десятки или сотни кубитов, что недостаточно для решения большинства практически значимых задач. Необходима разработка новых алгоритмов, способных эффективно использовать текущие и будущие возможности квантового оборудования.



Декогеренция

Потеря когерентности кубитов.

Масштабируемость

Ограниченное количество стабильных кубитов.

Стоимость и сложность

Разработка и эксплуатация дорогого оборудования.

ОГРАНИЧЕНИЯ ПАМЯТИ И СРАВНЕНИЯ С КЛАССИЧЕСКИМИ АЛГОРИТМАМИ

Квантовые компьютеры дополняют классические, но не заменяют их: ограниченная стабильность кубитов делает непрактичным хранение больших данных, где СУБД вроде Firebird остаются незаменимыми.

В то время как классические алгоритмы могут быть неэффективны для задач с экспоненциальной сложностью, квантовые алгоритмы не могут эффективно обрабатывать большие объемы данных, которые не укладываются в ограниченное количество кубитов.

50-100

Кубитов

Текущее количество
стабильных кубитов.

1000K

Кубитов

Необходимое количество для
полноценных вычислений.

10^{-6}

Сек

Время жизни кубитов
(микросекунды).

ТРАДИЦИОННАЯ И КВАНТОВАЯ СУБД

Основное отличие между традиционными и квантовыми вычислениями заключается в базовом элементе хранения информации.

Квантовые компьютеры обеспечивают экспоненциальный прирост производительности для специфических задач: алгоритм Шора факторизует числа за полиномиальное время, недостижимое классическим системам, благодаря параллельным квантовым операциям в суперпозиции.

Традиционные вычисления

- Биты (0 или 1)
- Последовательные операции
- Ограничены барьерами транзисторов

Квантовые вычисления

- Кубиты (суперпозиция 0 и 1)
- Параллельные вычисления благодаря суперпозиции и запутанности
- Потенциал для решения задач, недоступных для классических компьютеров

FIREBIRD КАК ИНТЕГРАЦИОННЫЙ СЛОЙ

Firebird – мост между классическими и квантовыми вычислениями: сочетает надежность СУБД с мощностью квантовых технологий для задач, где индексы неэффективны (графовые связи, неструктурированные данные).

Гибридная архитектура объединяет преимущества Firebird (надёжное хранение и транзакции) с квантовыми вычислениями, создавая основу для прорывных решений в анализе данных и оптимизации систем.

Хранение данных

Firebird как надежное хранилище для больших объемов информации.

Подготовка запроса

Идентификация задач, требующих квантовых вычислений.

Квантовая обработка

Передача данных в квантовый процессор для вычислений.

Возврат результата

Запись результатов квантовых вычислений обратно в Firebird.

29 МАЯ | МОСКВА

FIREBIRD  CONF

2025