

Слайд 1

Доброе утро, уважаемая комиссия. Вашему вниманию представляется курсовая работа на тему “Метод ускорения запросов в базе данных”. Руководитель проекта: Строганов Юрий Владимирович.

Слайд 2

Читаются цель и задачи

Слайд 3

В 2021 году компанией Гугл был представлен язык программирования, который основывается на рассмотрении способов организации хранилища данных с помощью математического аппарата, который представлен на слайде. Какие можно выделить преимущества такого подхода? Первое, данные могут храниться в виде предикатов. Второе, сохраняются операции теории множеств. И последнее, лингвистически запросы становятся проще для чтения и записи.

Слайд 4

Прежде, чем приступить работе, необходимо ограничить SQL-грамматику, рассмотрев только операторы SELECT, FROM, фильтрации WHERE и, например, LIKE, AND/OR. Также будет рассмотрен оператор связывания таблиц JOIN.

Слайд 5

Для того, чтобы понять, а что вообще можно сделать, необходимо разобраться с тем, как работает планировщик. В нем можно выделить независимые узлы, работу которых можно организовать параллельно. Дополнительно на основе идеи использования предикатов следует сочетать с одним из логических языков программирования.

Слайд 6

Среди параллелизмов выделяют следующие: независимый AND и OR. AND-параллелизм использует конъюнкцию предикатов, при этом результат будет собран из всех частей запроса. В основе OR-параллелизма лежит дизъюнкция предикатов, результат – первый выполнившийся предикат.

Слайд 7

Выбор СУБД играет немаловажную роль в проекте. Необходима СУБД, находящаяся в открытом доступе, ведь многие коммерческие компании ограничивают сейчас доступ к своим продуктам. Для развития первоначальной задумки необходима поддержка СУБД реляционной алгебры и SQL-запросов. Выбор пал на СУБД Postgres, так как она является стабильной и хорошо структурированной по сравнению с остальными.

Слайд 8

Реализации Prolog также должны быть открытыми и поддерживать следующие критерии (написаны на слайде, запомнить). Интересные результаты были продемонстрированы версией SICStus-Prolog, но к сожалению, студенческую лицензию за 3 месяца они так и не прислали.

Слайд 9

На слайде представлены технические характеристики устройства, на котором были проведены эксперименты.

Слайд 10

Первый эксперимент – определение зависимости времени выполнения SELECT запроса с фильтрацией от числа столбцов при постоянном числе записей. Обратите внимание на интересные результаты выполнения реализаций Prolog. Они выполнялись быстрее PostgreSQL без кэша, а также взяли преимущество на числе столбцов от 4 до 7.

Слайд 11

Во втором эксперименте определялась зависимость выполнения JOIN-запроса с фильтрацией по первой таблице от числа столбцов при постоянном числе записей. В этом случае Prolog реализации просели сильно настолько, что производительность различается в 10000 раз. При этом всего 10^5 записей.

Слайд 12

Последний эксперимент заключался в определении зависимости выполнения SELECT-запроса с фильтрацией, используя распараллеливание в виде потоков. Используя 2-3 потока удалось получить преимущество в производительности при 4, 5 и 9 столбцах при числе записей 10^7 .

Слайд 13

В ходе проекта была выполнена поставленная цель (на слайде читать) и задачи (на слайде).

Слайд 14

В качестве направлений развития данной области можно выделить следующие (слайд и краткие пояснения).