

Методы ускорения перебора соединений таблиц в планировании SQL запросов

НГУ, ММФ, 22126, Пьянзин Богдан

27 февраля 2025 г.

Структура курсовой работы

1 Введение

- Актуальность темы.
- Цель исследования.
- Задачи работы.
- Краткое описание рассматриваемых вопросов.

2 Основная часть

2.1 Теоретические основы соединений таблиц в SQL запросах

- Определение и классификация соединений и планирование запроса.
- Задача выбора порядка JOIN-ов.
- Факторы, влияющие на производительность (размеры таблиц, статистики, индексы, типы данных и т.д.).
- Влияние порядка JOIN на производительность СУБД.

2.2 Методы ускорения перебора соединений таблиц

- Классические подходы перебора соединений таблиц.
- Реализация в PostgreSQL.
- Современные подходы. %TODO

2.3 Сравнительный анализ методов ускорения %TODO

- Преимущества и недостатки каждого метода %TODO

3 Заключение %TODO

- Итоги работы.
- Рекомендации по применению методов.
- Перспективы исследований.

4 Список литературы

Введение

Актуальность темы

Современные СУБД работают с большим объёмом информации и транзакций, используя для ввода запросов язык SQL. В процессе трансляции запрос превращается сначала в логическое представление в виде дерева, затем с помощью оптимизатора СУБД в физический план исполнения. Производительность СУБД напрямую зависит от качества сгенерированного физического плана. Для создания "хорошего" плана, нужно решить одну из NP-полных задач перебора соединений таблиц, так как она требует сложных вычислений и значительных затрат ресурсов.

Сложность задачи обусловлена тем, что количество возможных соединений (бинарных) с n таблицами равно количеству бинарных деревьев с n листьями - Числа Каталана, которые имеют экспоненциальную скорость роста.

В классических подходах используется динамическое программирование (DP) и эвристический поиск для решения этой проблемы. Однако с ростом количества таблиц, потребление памяти (для хранения промежуточных результатов) и времени планирования становятся слишком велики. Эвристические методы не гарантируют оптимальность решения, так как находят локальное оптимальное решение, но не гарантируют глобальное оптимальное решение.

Современные подходы предлагают использовать машинное обучение на основе статистик, которые могут динамически меняться в процессе выполнения запроса. И создавать на их основе оптимальные планы.

Таким образом, выбор неэффективного плана запроса может привести к значительному замедлению работы системы, а процесс поиска хорошего решения является нетривиальной задачей. В свою очередь оптимальный план позволяет эффективно исполнить запрос, с приемлемым потреблением CPU, памяти, I/O, сетевых ресурсов (особенно актуально для распределённых СУБД).