

ОТЗЫВ ОФИЦИАЛЬНОГО ОППОНЕНТА

на диссертационную работу Буцацкого Рубена Артуровича
«Метод динамической компиляции SQL-запросов для реляционных
СУБД»,

представленную к защите на соискание учёной степени кандидата технических наук по специальности 2.3.5 – «математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»

Актуальность темы

Снижение стоимости памяти на несколько порядков и быстрый рост ее объемов за последние десятилетия позволили размещать значительную часть реляционных систем управления базами данных (СУБД) или даже всю СУБД целиком в оперативной памяти. Вследствие этого наибольший вес при работе с СУБД приобрели обработчики запросов, выполняемые процессором с помощью интерпретации. Если запросы достаточно сложные и выполняются долго, то накладные расходы на интерпретацию плана выполнения могут составлять значительную часть времени выполнения запроса. Для дальнейшего сокращения времени выполнения больших аналитических систем требуется ускорение выполнения запросов, которого можно добиться с применением динамических компиляторных технологий.

Диссертационная работа Р. А. Буцацкого посвящена *актуальной* задаче – разработке и реализации метода динамической компиляции запросов с целью ускорения их выполнения в СУБД.

Структура работы

Диссертационная работа содержит 168 страниц и состоит из введения, четырёх глав, заключения, списка литературы из 80 наименований, и двух приложений.

Во введении обосновывается актуальность темы исследования, ставятся цели и задачи, выделяется научная новизна и основные положения, выносимые на защиту.

В первой главе дается глубокий и хорошо структурированный обзор предметной области по применению моделей выполнения запросов и методов динамической компиляции в современных СУБД. Описывается четыре модели выполнения запроса в СУБД: модель итераторов (Volcano или pull, кортеж за

раз); материализующая модель (оператор за раз, колонка за раз); векторизующая модель (векторная, блочная); модель явных циклов (push, кортеж за раз). Указываются недостатки каждой из моделей и применимость к ним методов динамической компиляции. Далее описываются существующие методы динамической компиляции в СУБД, которые разделяются на два класса: компиляция выражений и компиляция запросов. Динамическая компиляция выражений, реализованная в распределенных системах, позволяет получить сравнительно небольшое ускорение, до 30% на тестовом наборе TPC-H. Динамическая компиляция запросов во многих современных коммерческих СУБД реализуется на основе модели явных циклов и позволяет получить ускорение в несколько раз. Однако в большинстве СУБД используется модель итераторов (Volcano-модель), что усложняет применимость современных методов динамической компиляции запросов. Решение этой проблемы является предметом диссертационной работы, рассматриваемой в последующих главах.

Во второй главе описывается метод динамической компиляции запросов с трансформацией операторов плана запроса из Volcano-модели в модель явных циклов, что является важнейшим результатом работы. Приводится процесс генерации кода запроса в модели явных циклов с описанием разработанного интерфейса оператора, необходимого для трансформации и выполнения плана запроса в обратном порядке. Описывается используемый при компиляции важный механизм поддержки прерывания в модели явных циклов для досрочного завершения обработки запроса. Разработанный автором процесс декомпозиции алгоритмов операторов Volcano-модели на функции разработанного интерфейса в модели явных циклов позволяет трансформировать *на лету* операторы из одной модели в другую с использованием динамического компилятора. Этот процесс представлен псевдокодом их реализации для всех основных типов операторов СУБД. Описывается метод динамической компиляции выражений с применением оптимизации по открытой вставке предкомпилированных функций СУБД. Безусловно важными являются разработанные автором эвристики, определяющие стратегии выбора способа выполнения запросов, основывающиеся на оценке затрат на интерпретацию, на динамическую компиляцию, на кэширование и повторное использование скомпилированного кода запроса, который избавляет от расходов на компиляцию и оптимизацию.

В третьей главе приводится описание программной реализации динамического компилятора запросов для открытой СУБД PostgreSQL на

основе разработанных в диссертации методов. Представлена архитектура реализованного динамического компилятора на базе системы LLVM. Приводится описание с указанием особенностей выбранной СУБД PostgreSQL и компиляторной инфраструктуры LLVM. Описана реализация функций-генераторов интерфейса в модели явных циклов для основных типов операторов СУБД PostgreSQL. Важным также является предложенный автором метод кэширования кодов запросов, использующих специально формируемый массив параметров запроса, для возможности повторного применения скомпилированного кода.

В четвёртой главе приведены результаты апробации реализованного динамического компилятора запросов на промышленных тестовых наборах TPC-H и TPC-DS. Приведенные результаты наглядно демонстрируют высокую эффективность разработанных в работе методов динамической компиляции запросов, обеспечивающих ускорение в несколько раз по сравнению с интерпретатором PostgreSQL.

В заключении перечислены основные результаты диссертационной работы, а также обсуждаются возможные направления дальнейших исследований.

В приложениях приводятся подробные результаты тестирования производительности реализованного динамического компилятора запросов для СУБД PostgreSQL на тестовых наборах TPC-H и TPC-DS на платформах x86-64 и ARM, копия свидетельства о регистрации программы для ЭВМ.

Научная новизна и практическая значимость

В работе получены следующие новые научные результаты:

1. Метод динамической компиляции запросов с трансформацией на лету операторов плана запроса из модели Volcano в модель явных циклов, который может быть применён к СУБД с моделью Volcano для реализации динамического компилятора.

2. Метод динамической компиляции выражений в SQL-запросах с применением открытой вставки предварительно скомпилированных встроенных функций СУБД.

3. Эвристики стратегии выполнения запроса на основе оценок затрат на интерпретацию и динамическую компиляцию, устанавливающие стратегии выполнения для запросов разной сложности. Метод кэширования кода, сгенерированного динамическим компилятором, для одинаковых SQL-запросов.

4. Программное расширение к СУБД PostgreSQL, реализующее метод динамической компиляции SQL-запросов.

Результаты диссертационной работы опубликованы в семи научных статьях в рецензируемых журналах, рекомендованных ВАК, две из которых индексируются Scopus и WoS.

Замечания и недостатки по представленной диссертационной работе:

1. В разделе 1.2.1 автор пишет, что в СУБД PostgreSQL, начиная с версии 11, есть поддержка JIT-компиляции выражений. Однако в работе нет сравнительного тестирования разработанного динамического компилятора с существующим JIT-компилятором выражений.

2. Выбранная автором компиляторная инфраструктура LLVM хорошо подходит для реализации статических компиляторов, но, хотя она имеет в своем составе специальные модули поддержки динамической компиляции, является довольно тяжелой. В работе нет оценок влияния выбранной инфраструктуры на время динамической компиляции запросов.

3. Определенный интерес представляет распределение времени работы динамического компилятора на различные этапы: генерация представления при переходе от Volcano-модели к модели явных циклов; оптимизации представления; генерация кода целевой архитектуры. Такой анализ, важный с точки зрения ускорения работы динамического компилятора, отсутствует в работе.

4. В главе 4 автором не обоснован выбор среднего арифметического в качестве среднего значения времени выполнения всех запросов из тестовых наборов, хотя в них имеется значительный разброс – от нескольких процентов до нескольких раз.

5. Для СУБД масштаба 75 оригинальные (NUMERIC, CHAR(1)) типы СУБД по сравнению с нативными (DOUBLE, ENUM) на тестовом наборе TPC-H приводят к более быстрому выполнению для платформы x86-64 и к более медленному (что естественно) – для ARM. Эти странные различия не нашли объяснения в работе.

6. Автором в недостаточной мере раскрыта проблема возможности отладки и поддержки сгенерированного динамическим компилятором кода.

Перечисленные недостатки не влияют на общую положительную оценку работы.

Диссертационная работа Р. А. Бучацкого является законченным научным исследованием. Основное содержание диссертации отражено в опубликованных статьях и обсуждено на научных конференциях. Практическая ценность подтверждается результатами применения разработанных методов.

Автореферат полно и правильно отражает содержание диссертационной работы.

Выводы

Диссертационная работа соответствует всем требованиям ВАК РФ, предъявляемым к диссертациям на соискание учёной степени кандидата технических наук, а её автор, Бучацкий Рубен Артурович, заслуживает присуждения ему учёной степени кандидата технических наук по специальности 2.3.5 – «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей».

Официальный оппонент

кандидат технических наук, начальник отделения «Системы программирования» Публичного акционерного общества «Институт электронных управляющих машин им. И. С. Брука», Российская федерация, 119334, Москва, ул. Вавилова, д. 24,

В.Ю. Волконский

21 ноября 2022 г.