

ОТЗЫВ ОФИЦИАЛЬНОГО ОППОНЕНТА

на диссертационную работу Четвериной Ольги Александровны

“Повышение качества компиляции кода в режиме по умолчанию”

представленную к защите на соискание ученой степени кандидата физико-математических наук по специальности 05.13.11 - “Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей”

Актуальность темы

Производительность программ в значительной степени определяется качеством планирования их исполняемого кода, за которое отвечает оптимизирующий компилятор, работающий в соответствии с особенностями целевой архитектуры. Для поддержки этого оптимального планирования на конкретной архитектуре компилятор последовательно применяет ряд преобразований к полученному коду. Полученный в результате преобразований код должен быть семантически эквивалентен исходному, но работать более эффективно. Естественно, что качество таких преобразований (повышение эффективности) нужно как-то оценивать. Для части преобразований качественный результат их применения можно определить однозначно по анализируемому коду. Однако в большинстве случаев оптимальный способ применения преобразований либо зависит от недоступных на этапе компиляции характеристик исполнения кода, таких как профиль исполнения или особенности обрабатываемых данных, либо не может быть точно вычислен за приемлемое время.

Необходимость повышения скорости исполнения кода в режиме по умолчанию (без доступа к профилю) с одновременным учетом времени компиляции и определяет актуальность диссертационной работы.

Структура работы

Диссертация состоит из четырех глав, введения и заключения. Объем диссертации составляет 125 страниц, включая 6 таблиц и 18 рисунков. Список литературы содержит 73 наименования.

Во введении сформулированы цели и задачи работы, обосновывается актуальность выбранной темы, описывается теоретическая и практическая значимость результатов, сформулированы основные положения, выносимые на защиту:

- разработка методов частичной раскрутки коротких путей и рекуррентностей циклов, которые позволяют лучше спланировать сложные циклы с разновесными ветвлениями;
- неагрессивные методы оптимизации работы с памятью;
- числовая многокритериальная оценка качества компиляции, учитывающая по-процедурный характер оптимизации;
- формулировка задачи минимизирующей классификации, соответствующей обучению с построенной оценкой качества компиляции и алгоритмы ее решения

В первой главе рассматриваются способы настройки оптимизации кода приложений в современных компиляторах (gcc, lcc) и сформулирована задача по-процедурной направленной оптимизации. Автором описаны основные режимы компиляции, к которым относятся:

- уровни оптимизации,

- режим межмодульной оптимизации,
- двухфазный режим со сбором профиля исполнения
- установка общих правил для кода программы и ее исполнения.

Отмечается, что сложность настройки качественного применения последовательности преобразований затрудняет (или делает невозможным) настройку теоретически наиболее эффективной компиляции за ограниченное время. Это происходит в силу двойственности (совместного влияния) большинства преобразований с точки зрения использования ресурсов, таких как планируемое количество тактов исполнения, нагрузка на кэш-память, требуемое количество регистров и т.д. Соответственно, это приводит к тому, что для настройки работы компилятора используются эвристические методы, а для более качественной оптимизации конкретного приложения может быть использована дополнительная настройка набора преобразований.

Приведен анализ такого метода автоматической настройки последовательности преобразований как итеративные решатели, работа которых заключается в многократной компиляции кода с различными наборами настроек и сравнении результатов либо по временам исполнения на вычислительной машине, либо по оценочным временам исполнения, которые вычисляются по спланированному коду. Отмечено, что, несмотря на возможные преимущества, все решатели такого типа требуют значительного времени компиляции, а также тренировочного исполнения приложений на представительных данных. Результатом является практическая неприменимость для больших приложений.

В качестве альтернативного метода настройки компиляции рассматривается подход на основе использования машинного обучения для выбора оптимальных настроек компиляции по вычислимым характеристикам компилируемого кода. Подобный подход, используемый в системе *Milepost GCC* для настройки компиляции приложений, подробно проанализирован в диссертационной работе. Отмечено, что хотя проблема разработки набора опций или оптимизационных последовательностей для решения задачи построения итерационных решателей уже в значительной мере исследована в научной литературе, вопрос поиска эффективных оптимизаций для однофазного режима компиляции остается открытым. Решению этой проблемы и посвящена глава 2.

Во второй главе рассматривается вопрос поиска и настройки методов оптимизации, применимых в режиме по умолчанию. Одним из результатов диссертации явилось выявление основных существующих высокоэффективных методов оптимизации, которые по разным причинам не работают или малоэффективны в случае отсутствия информации о профиле исполнения задач. На основании этого анализа в работе предложены альтернативные методы оптимизации и настройки для использования по умолчанию.

В частности, важные отличия работы оптимизаций при отсутствии профильной информации выявились на задачах с большим числом ветвлений, значительная часть которых редко исполняется. Из описанных проблем:

- различным образом осуществляется подстановка процедур;
- не удается повысить качество конвейеризации циклов с несбалансированными альтернативами классическими методами;
- возникают сложности с планированием одновременного исполнения операций соседних веток;
- возникает большее число блокировок конвейера при обращениях к памяти.

С целью увеличения параллельности исполнения сложных циклов (циклов с внутренними циклами, вызовами или большим количеством узлов) с несбалансированными ветвлениями в режиме без профиля или с небольшим отличием значения счетчиков участков, автором разработан метод частичной раскрутки путей исполнения (*short path unroll*).

Вторая часть предложенных в диссертации преобразований и их настроек при компиляции без профиля касается уменьшения количества блокировок по чтениям. Для использования по умолчанию автором предложены неагрессивные методы оптимизации работы с памятью.

Экспериментальные результаты применения предложенных методов показали значительное ускорение ряда приложений и небольшое ускорение исполнения в среднем в режиме однофазной компиляции.

В третьей главе подробно рассмотрен вопрос построения метрики (числовой характеристики), позволяющей оценить качество обучения компилятора (по-процедурное качество компиляции с точки зрения сразу нескольких критериев для пакета задач). Чтобы получить единую числовую оценку производительности компиляторов и процессоров на пакете приложений обычно используется среднее геометрическое значение времен исполнения приложений. В общем случае, одновременно получить минимум времени компиляции и времени исполнения невозможно, что приводит к необходимости поиска минимумов по Парето. В работе доказаны утверждения, которые позволяют однозначным образом скаляризовать целевую функцию задачи оптимизации процесса компиляции по временам исполнения и временам компиляции процедур. Также в работе доказано, что можно однозначно определить вид функционала качества компиляции.

Автором приведен пример использования построенного функционала качества на компиляторе под архитектуру Эльбрус. В точке его минимума было достигнуто ускорение исполнения на 12,36% и уменьшение времени компиляции в 1,9 раз в среднем на пакетах приложений *spec2000 benchmark*.

В четвертой главе сформулирована и решена задача машинного обучения, соответствующая минимизации построенного функционала качества компиляции. Поставленная задача наиболее близка к задаче классификации, в которой для некоторого множества объектов определено множество допустимых ответов, и есть целевая функция, значения которой известны только на конечном подмножестве исходного множества объектов. При этом требуется найти алгоритм, приближающий целевую функцию на всем исходном множестве.

Для измерения ошибки классификации используется эмпирический риск, который представляет собой средневзвешенную по объектам ошибку алгоритма. В диссертационной работе показано, что задача минимизации функционала качества существенно отличается от задачи классификации, поскольку: не определены, и в общем случае не определяемы, верные классы для отдельных объектов; для неаддитивного по объектам функционала качества, каковым является функционал качества компиляции, нельзя вычислить ошибку алгоритма для отдельного объекта. Для преодоления этого, в работе предложен отдельный класс задач машинного обучения.

Экспериментальные результаты применения компиляции с автоматическим выбором оптимизационной линейки с построением 10 областей для классов, что соответствует максимум 140 сравнениям или базовым правилам, показаны в работе на пакетах приложений *spec2000*. Для задач с целыми вычислениями (пакет CINT) было получено среднее ускорение времени исполнения на 2,9% и времени компиляции на 26,7%. Задачи с плавающими вычислениями (пакет CFP), в среднем ускорились на 8,2% по исполнению, а среднее время компиляции сократилось на 6,7%. В качестве тестовых приложений, не входящих в обучающую выборку, в работе использовался пакет *spec95*. На нем было получено ускорение производительности на 35% и компиляции на 94% от ускорения, полученного на обучающем пакете

Также в работе приводятся данные о сравнении минимума и результата машинного обучения. При осуществлении 7-ми шагов классификации было достигнуто 88% возможного ускорения приложений в среднем.

В заключении приводятся основные результаты диссертационной работы и направления дальнейших исследований. Основные результаты диссертационной работы:

1. Предложены преобразования частичной раскрутки критических путей и длин рекуррентностей цикла, позволяющие лучше спланировать сложные циклы с равновесными ветвлениями, и доказаны утверждения об оценке их эффективности.
2. Предложены неагрессивные методы оптимизации работы с памятью, применимые в режиме по умолчанию: метод увеличения дистанции от чтений с ограничением планируемого времени исполнения цикла и метод построения подкачки структур в циклах.
3. Предложен механизм автоматического выбора оптимизирующей последовательности преобразований для каждой процедуры на раннем этапе компиляции.
4. Предложена числовая многокритериальная оценка качества компиляции, учитывающая по-процедурный характер оптимизации, и показано ее преимущество по сравнению с вероятностной оценкой при решении задач машинного обучения компиляторов.
5. Сформулирована и доказана теорема о представлении всех функций двух переменных, обладающих свойством сохранения порядка при растяжениях. Также доказано утверждение о представлении функционала качества компиляции, учитывающего время исполнения и время компиляции.
6. Сформулирована задача минимизирующей классификации, соответствующая задаче машинного обучения компилятора с построенным функционалом качества. Автором предложены методы решения задачи минимизирующей классификации и доказаны утверждения об их сходимости.

В работе описывается использование представленных методов при разработке оптимизирующего компилятора с языков C, C++ и Фортран для микропроцессоров «Эльбрус». Суммарное ускорение от совокупности примененных методов в базовом режиме компиляции составило в среднем около 6,1% для задач с целыми вычислениями и около 15% для задач с плавающими вычислениями. При этом время, требующееся для компиляции приложений с целыми вычислениями, сократилось больше, чем на 20%, а с плавающими - больше, чем на 6%.

Работа прошла научную и практическую апробацию, по ней опубликовано 10 печатных работ, и получено свидетельство о государственной регистрации программы для ЭВМ. Предложенные методы встроены в практические инструменты.

По диссертации можно отметить следующие недостатки:

- Не совсем четко описан выбор признаков для модели обучения. Указано, что это вычисляемые характеристики процедур по их промежуточному представлению после межпроцедурной оптимизации. Указаны признаки, коррелирующие с результатами компиляции (общее количество операций в процедуре, количество различных операций в процедуре, оценочное время исполнения процедуры и т.д.), но не сказано о независимости признаков (например, точно ли не коррелируют общее количество операций в процедуре и количество различных операций в процедуре?), нужно ли какое-либо уменьшение пространства признаков (факторный анализ, PCA) и т.д.

- В работе утверждается о семантической эквивалентности кода в результате преобразований, но формально это никак не доказывается

Указанные недостатки не снижают общей научной и практической ценности результатов данной работы.

Автореферат полно и правильно отражает содержание диссертационной работы.

Диссертационная работа является завершенным научным исследованием и имеет практическую значимость. Она соответствует паспорту специальности 05.13.11 – “Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей” и отвечает всем требованиям, предъявляемым ВАК РФ к кандидатским диссертациям, а ее автор, Четверина Ольга Александровна, заслуживает присуждения ей ученой степени кандидата физико-математических наук по специальности 05.13.11 – “Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей”.

Официальный оппонент

Кандидат физико-математических наук, старший научный сотрудник
лаборатории Открытых информационных технологий факультета ВМК
МГУ имени М.В. Ломоносова, Российская федерация, 119991, Москва, ГСП-1,
Ленинские горы, д. 1, стр. 52, факультет ВМК

24.04.2019г.

Д.Е. Намиот