

Отзыв официального оппонента

Новикова Сергея Викторовича на диссертационную работу
Гимпельсона Вадима Дмитриевича
«Сокращение длины критических путей при динамической трансляции
двоичных кодов»,
представленную к защите на соискание учёной степени кандидата физико-
математических наук по специальности 05.13.11 – «Математическое и
программное обеспечение вычислительных машин, комплексов и
компьютерных сетей»

Решение проблемы повышения производительности является одной из основных целей при разработке аппаратных комплексов и программных средств. Создание программ, максимально полно и эффективно использующих возможности вычислительной системы, требует не только высокой квалификации разработчиков, но уже невозможно без применения компиляторных систем, способных в автоматическом режиме создавать высокопроизводительный двоичный код. Ключевой задачей компилятора при оптимизации кода является уменьшение длины критического пути. Любые новые подходы, позволяющие улучшать данную характеристику результирующего кода, вызывают большой интерес среди разработчиков компиляторных систем. Именно поэтому методы сокращения длины критических путей, представленные в данной работе, имеют несомненную **актуальность**.

Оптимизационные задачи, как правило, имеют высокую вычислительную сложность и занимают много процессорного времени. Поэтому ресурсоёмкие алгоритмы, позволяющие наиболее полно оптимизировать программный код, обычно применяются при статической компиляции. В условиях динамической трансляции, когда скорость работы алгоритма становится зачастую даже более важной характеристикой, чем качество генерируемого кода, возникают жёсткие временные ограничения на возможности реализации методов оптимизации. Разработчикам динамических трансляторов приходится отказываться от эффективных методов уменьшения критических путей в коде, сильно упрощая алгоритмы. Помимо этого, при трансляции кода из бинарного представления возникает набор дополнительных проблем, как, например, задача восстановления точного контекста. Это порождает новые контексты применения оптимизаций, отличные от классических примеров.

Научная новизна результатов, представленных в работе, заключается

в разработке быстрого алгоритма сокращения длины критического пути в ациклических областях и разработке быстрого алгоритма конвейеризации циклов. Разработанные алгоритмы имеют строгое формальное обоснование, а реализация этих алгоритмов в двоичном трансляторе для архитектуры «Эльбрус» показывает их высокую эффективность.

Диссертационная работа состоит из введения, четырёх глав, заключения, двух приложений и списка литературы из 113 наименований.

Во введении приводится обоснование актуальности работы, формулируются цели и задачи, приводятся основные научные результаты, обосновываются теоретическая, практическая ценности и приводятся данные об апробации работы.

В первой главе делается общий обзор технологии двоичной трансляции, используемой для обеспечения совместимости программного обеспечения, разработанного для отличной от используемой микропроцессорной архитектуры. Целевой выбрана архитектура EPIC. Дается её описание, характеристики, описываются особенности двоичного транслятора для этой архитектуры как целевой платформы. Показывается, как при трансляции двоичного кода возникают зависимости между инструкциями, которые можно оптимизировать и, таким образом, ускорить результирующий код. Для наглядного объяснения в качестве псевдокода используется внутреннее представление двоичного транслятора. В результате формулируется задача разработки набора быстрых алгоритмов для сокращения длины критических путей для двоичного транслятора с архитектуры x86 в архитектуру «Эльбрус»:

- быстрые алгоритмы минимизации высоты графа зависимостей без построения новых операций;
- быстрые алгоритмы минимизации высоты графа зависимостей с построением новых операций;
- алгоритм конвейеризации циклов, интегрированный с методами разрыва зависимостей.

Во второй главе предложены методы разрыва зависимостей без построения новых операций в ациклических областях. Первый метод – алгоритм переименования регистров, базирующийся на счётчиках использования регистров на компонентах графа потока данных. Второй метод – алгоритм применения спекулятивности. Третий метод – использование частичных предикатов. В конце главы приводятся

экспериментальные результаты, которые показывают, что алгоритм построения частичных предикатов был реализован в двоичном трансляторе для микропроцессора «Эльбрус» и дал 1-2% прироста производительности на результирующем коде. Замедление скорости трансляции при этом составило 0,4%.

В третьей главе рассмотрены методы разрыва зависимостей в ациклических областях с построением новых операций. Предложен новый алгоритм минимизации высоты графа зависимостей, который использует технику разрыва антивисимостей, зависимостей по выработке результата, предикатных зависимостей, зависимостей между обращениями в память. Разрыв зависимостей позволяет сократить длину критического пути, однако для осуществления каждого рассматриваемого автором разрыва необходимо построить новую операцию. В силу этого простой алгоритм, который разрывает все зависимости, не эффективен, так как создаётся большое количество новых операций. Автором предложен новый алгоритм, который с одной стороны позволяет достичь минимальной высоты графа зависимостей, а с другой пытается не строить лишних операций. Такой подход позволяет получить скорость работы результирующего кода близкую к оптимальной. Автор приводит формальное обоснование доказывая, что приведённый алгоритм минимизации высоты графа зависимостей преобразует произвольный граф зависимостей в граф зависимостей с минимально возможной высотой для этих преобразований. В конце главы приводятся результаты замеров производительности двоичного транслятора для микропроцессора «Эльбрус» на пакете бенчмарков. С применением данного алгоритма целочисленные задачи пакета SPEC CPU2000 показывают прирост производительности на 12%, вещественные задачи на 22%. Наиболее часто исполняющиеся участки кода ОС Windows 2000 и пользовательских приложений увеличили скорости работы результирующего кода на 23%. Общее время работы алгоритма составило 3.8% от времени трансляции.

В четвертой главе автором предлагается быстрый алгоритм конвейеризации циклов. Необходимо отметить, что это первый опубликованный случай эффективного использования такой сложной оптимизации, как конвейеризация циклов, в динамическом двоичном трансляторе. Основной подход данного алгоритма базируется на идее переноса на каждом шаге через начало цикла нескольких операций. Решение, какие операции переносить, а какие нет, основывается на разметке времён раннего и позднего планирования на расширенном графе зависимостей. При этом переносятся только те операции, которые действительно нужно

переносить, чтобы улучшить качество результирующего кода, а те, перенос которых не даёт никакого положительного эффекта, отбраковываются. Это существенно повышает эффективность оптимизации. Для реализации данного подхода в работе разработан алгоритм разметки времён раннего и позднего планирования на расширенном графе зависимостей. Доказана корректность и оптимальность предложенного алгоритма, а также производится оценка его алгоритмической сложности. Также в четвёртой главе приводится описание эвристики, с помощью которой можно произвести оценку минимального возможного размера цикла, при котором алгоритм конвейеризации должен прекращать свою работу. Описанный алгоритм конвейеризации циклов был реализован в двоичном трансляторе для микропроцессора «Эльбрус». На целочисленных и вещественных задачах из пакета SPEC CPU2000 применение алгоритма дало 5% и 28% увеличения скорости работы результирующего кода соответственно. На горячих участках ОС Windows 2000 и пользовательских приложений увеличение скорости работы результирующего кода составило 17%. Время работы алгоритма составило 3.5% и 10% от общего времени трансляции на целочисленных и вещественных задачах соответственно.

По диссертационной работе имеются следующие замечания:

- Применение предложенных методов вполне актуально для систем статической компиляции для языков высокого уровня. Безусловным плюсом был бы дополнительный анализ на данную тему.
- Основной акцент в диссертации сделан на анализе применения методов для архитектуры EPIC, в то время как результаты для суперскалярных процессоров, представляющие несомненный интерес, отсутствуют.
- Пакет SPEC CPU2000, используемый для замеров производительности, является устаревшим индустриальным стандартом и имеет более современную версию SPEC CPU2006.

Приведённые замечания не влияют на общую положительную оценку работы.

Диссертационная работа Гимпельсона В.Д. является законченным научным исследованием. **Практическая значимость** диссертации заключается в том, что все предложенные в работе методы и новые алгоритмы были реализованы в двоичном трансляторе для микропроцессора «Эльбрус» и демонстрируют существенное повышение производительности двоичного транслятора. Для предложенных методов и алгоритмов сформулированы и доказаны теоремы, показывающие корректность и

оптимальность предложенных алгоритмов, что подтверждает достоверность и обоснованность результатов работы. Основные результаты работы были опубликованы, и прошли апробацию на научных конференциях и семинарах.

Автореферат полно и правильно отражает содержание диссертационной работы.

Диссертационная работа Гимпельсона Вадима Дмитриевича соответствует всем требованиям ВАК РФ, предъявляемым к диссертациям на соискание учёной степени кандидата физико-математических наук, а её автор, Гимпельсон Вадим Дмитриевич, заслуживает присуждения ему учёной степени кандидата физико-математических наук по специальности 05.13.11 – «математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей».

Кандидат технических наук, руководитель группы
по разработке бинарной трансляции компании Intel

С.В. Новиков

Подпись кандидата технических наук Нови
С.В. заверяю, Финансовый Директор R&D
компании Intel

«21» марта 2018 г.