

Автоматическая векторизация вызовов трансцендентных функций*Д.А. Земляков*

АО «МЦСТ»

ПАО «ИНЭУМ им. И.С. Брука»

Как правило, значительная часть времени исполнения программ приходится на циклы, поэтому задача оптимизации циклов важна для обеспечения максимальной производительности [1]. Наличие в теле цикла вызовов трансцендентных функций может значительно ослабить эффект от его векторизации и распараллеливания. В этом случае особенно возникает необходимость в автоматической векторизации операций вызовов. Проблема приобретает наибольшую актуальность применительно к архитектурам со статическим планированием, рассчитанным на достижение высших показателей производительности, к которым относятся серии отечественных микропроцессоров архитектуры «Эльбрус», предназначенных для разработки крупномасштабных информационно-вычислительных систем стратегического назначения [2].

Методы автоматической векторизации исследуются достаточно давно, и как показывает практика, их использование позволяет значительно увеличить производительность процессоров. Большинство алгоритмов ориентировано на использование коротких векторных инструкций, на которые можно заменить группы изоморфных скалярных операций после «раскрутки» цикла. Однако при наличии в цикле вызовов трансцендентных функций наибольший интерес представляет алгоритм, не относящийся к этому классу. В данном алгоритме используются значительно более быстрые векторные трансцендентные функции, использующие архитектурные особенности конкретного микропроцессора. В частности, для микропроцессоров архитектуры «Эльбрус» подобные функции реализованы в библиотеке EML (Elbrus Math Library) [2].

Для векторизации рассматриваются только удовлетворяющие требованиям анализа алгоритма операции самых вложенных циклов с подходящей структурой. В случае известного числа итераций цикла необходимо, чтобы оно превышало некоторое пороговое значение, после которого векторизация становится эффективной. В случае неизвестного числа итераций строится две версии цикла, к одной из которых применяется оптимизация, а выбор используемой версии происходит при исполнении программы. В рабочем цикле перед операцией вызова трансцендентной функции строятся операции записи ее аргументов во временный входной массив. Вместо чтений результата исходной скалярной операции вызова строятся операции чтения результата векторной функции, взятые из временного выходного массива. В новом пустом эпилоге исходного цикла строится вызов необходимой векторной функции, аргументами которой становятся указанные временные входной и выходной массивы. Оригинальная скалярная операция вызова удаляется, строится копия исходного цикла после вызова векторной функции. В оригинале и копии цикла устраняются избыточные операции.

В случае циклов с большим или неизвестным числом итераций может потребоваться разрезание исходного цикла. Для эффективной работы с данными они должны размещаться в стеке, входной и выходной массивы векторной функции должны помещаться в кэш L1. Исходя из этих требований размером массива была выбрана константа 1000. Также при разрезании исходного цикла внутренний цикл не должен отличаться от исходного с т.з. базовой индуктивности. В результате преобразования строится внешний охватывающий цикл с шагом счетчика равным выбранной константе. Счетчик внешнего цикла одновременно служит начальным значением для счетчика модифицированного внутреннего цикла и его копии, полученной в результате векторизации. Число итераций внутренних циклов на каждом шаге, кроме последнего, равно выбранной константе. На последнем шаге идет работа с оставшимися данными.

Автоматическая векторизация вызовов трансцендентных функций на основе данного алгоритма была реализована и внедрена в оптимизирующий компилятор для архитектуры «Эльбрус». Были сформулированы и протестированы условия применимости оптимизации, легшие в основу анализа контекста. Динамическая проверка количества итераций расширила границы применимости алгоритма. Для избежания выполнения действий, портящих контекст

более эффективным методам оптимизации, например, вынесению инвариантных операций, алгоритм был дополнен анализом эффективности.

Были проведены замеры времени исполнения задач из пакетов SPEC CPU2006 [3] на машинах с микропроцессорами с архитектурой “Эльбрус”, показавшие прирост производительности до 17%.

Литература

1. *Muchnick S.* Advanced Compiler Design Implementation. 5-е издание. — San Francisco: Morgan Kaufmann Publishers, 1997. 856 с. ISBN 978-1-5586-0320-2
2. *Ким А.К., Перекатов В.И., Ермаков С.Г.* Микропроцессоры и вычислительные комплексы семейства Эльбрус. — СПб.: Питер, 2013. 272 с. ISBN 978-5-459-01697-0
3. Standard Performance Evaluation Corporation. [Электронный ресурс] <http://www.spec.org/>