Земляков Д. А.
АО “МЦСТ”, г. Москва

ПРОГРАММНАЯ КОНВЕЙЕРИЗАЦИЯ ЦИКЛОВ В ОПТИМИЗИРУЮЩЕМ КОМПИЛЯТОРЕ ДЛЯ АРХИТЕКТУРЫ SPARC И “ЭЛЬБРУС”

Как правило, значительная часть времени исполнения программ приходится на циклы, поэтому задача распараллеливания циклов важна для обеспечения максимальной производительности. В случае сильной зависимости по данным особенно возникает необходимость в программной конвейеризации. Проблема приобретает наибольшую актуальность применительно к архитектурам со статическим планированием, рассчитанным на достижение высших показателей производительности, к которым относятся серии отечественных микропроцессоров архитектуры SPARC и «Эльбрус», предназначенных для разработки крупномасштабных информационно-вычислительных систем стратегического назначения.

Большинство алгоритмов программной конвейеризации могут быть отнесены либо к алгоритмам модульного планирования (modulo scheduling), либо к алгоритмам по распознаванию устойчивого состояния (kernel recognition). Однако, наибольший интерес представляет идея, лежащая в основе алгоритма Enhanced pipeline scheduling, не относящегося к этим классам. Данный алгоритм не требует проведения сложных итеративных попыток планирования, либо поиска ядра в раскрученном коде. На каждом этапе сохраняется корректное состояние цикла, в отличие от полного перестроения в других алгоритмах. Конвейеризация происходит во время переноса операций по обратной дуге.

Две основные фазы сменяют друг друга, пока не будут рассмотрены все межитерационные зависимости цикла. В первой проводится поиск операций рабочего множества, таких операций критического пути, которые не имеют предшественников по зависимостям в теле цикла без учёта обратных дуг. Во второй фазе операции рабочего множества перемещаются на итерацию вверх. Исходные операции одновременно копируются в пролог цикла и переносятся по обратной дуге вниз цикла, как операции с последующей итерации. Важным моментом является разрыв существующих и возникающих антизависимостей. Конвейеризация происходит при переносе операции по обратной дуге, когда та копируется вниз цикла, как операция последующей итерации. Пролог и эпилог формируются автоматически. Эффективность зависит от доступности ресурсов и зависимостей по данным между операциями.

На основе данного алгоритма была реализована и внедрена в оптимизирующий компилятор для архитектуры SPARC оптимизация программной конвейеризации циклов. Были сформулированы и протестированы условия применимости оптимизации, легшие в основу анализа контекста. При невозможности статически определить количество итераций была использована динамическая проверка, что позволило расширить границы применимости. Ресурсное планирование учитывает особенности планирования операций для данной архитектуры. Для избежания выполнения неэффективных переносов операций алгоритм был дополнен анализом эффективности, просчитывающим наперед потенциальный результат. Оптимизация была адаптирована для архитектуры «Эльбрус», для чего помимо архитектурных особенностей были учтены особенности компилятора и некоторые специфичные оптимизации.

Были проведены замеры времени исполнения задач из пакетов SPEC CPU на машинах с микропроцессорами МЦСТ R1000, совместимыми с архитектурой SPARC V9, а также машинах с архитектурой “Эльбрус”. Замеры эффективности показали прирост производительности до 11%.

Литература:

1. Альфред Ахо, Моника Лам, Рави Сети, Джеффри Ульман Компиляторы: принципы, технологии и инструментарий = Compilers: Principles, Techniques, and Tools. — 2-е издание. - М.: «Вильямс», 2008. - 1184 с. - 1500 экз. - ISBN 978-5-8459-1349-4

2. Steven S. Muchnick Advanced Compiler Design Implementation. - 5-е издание. - San Francisco: Morgan Kaufmann Publishers, 1997. - 856 с. - ISBN 978-1-5586-0320-2

3. 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

4. V. H. Allan, R. B. Jones, R. M. Lee, and S. J. Allan. Software pipelining. ACM Comput. Surv., 27(3), 1995.