УДК 004.4'422

Преобразование компилятором массива структур в несколько массивов

В.Е. Шампаров[1]***, А.Л. Маркин***[1]

[1]АО «МЦСТ»

Одной из задач оптимизирующей компиляции является улучшение работы с кэшем процессора. Кэш хранит часто используемые данные из оперативной памяти в т. н. кэш-линиях определённого размера.

Многие языки программирования предоставляют возможность группировать различные данные в массивы и структуры. Но существуют задачи, в которых использование указанных возможностей затрудняет оптимизацию обращений к кэш-памяти. Одним из подобных случаев является хранение массива структур в структуре. Если в таком случае программа в цикле обращается только к одному полю каждого элемента массива, то в кэш загружаются не только используемые данные, но и не нужные для цикла поля структур — элементов массива. Из-за этого возрастает количество кэш-промахов. Для исправления указанной ситуации требуется увеличение локальности данных, которое достигается с помощью декомпозиции — преобразования массива структур в несколько массивов простых типов, соответствующих полям структуры элемента массива.

В компиляторах GCC версии 4.3 [2] и Open64 [3] была реализована оптимизация Structure Peeling, декомпозирующая массив структур. В рамках данной работы на основе алгоритма Structure Peeling для Open64 создан алгоритм выявления и декомпозиции массива структур, лежащего в структуре. Он выполнен в виде оптимизации SA2AIS в составе компилятора LCC для микропроцессоров семейства «Эльбрус».

Алгоритм предполагает два основных этапа:

1. Анализ применимости — обнаружение структур, которые можно декомпозировать, и участков кода с доступом к полям этих структур.
2. Применение оптимизации — создание новой иерархии полей структур и соответственное изменение всех обращений.

Этап применения оптимизации состоит из следующих стадий:

1. Создание новой иерархии полей структур.
2. Изменение типов переменных компилируемой программы согласно новой иерархии.
3. Изменение обнаруженных на этапе анализа обращений к полям элементов массивов.
4. Изменение всех прочих обращений к изменённым структурам.

В ходе разработки оптимизации проведено исследование нескольких задач из пакета тестирования SPEC CPU2006 [4]. Контекст для применения оптимизации обнаружен только в задаче 462.libquantum. При применении реализованной оптимизации к данной задаче удалось получить ускорение выполнения задачи на 52,5%. Количество блокировок конвейера процессора из-за неготовности данных уменьшилось на 71,1%, а количество исполненных операций почти не изменилось.

Литература

1. *Cooper K.D., Torczon L.* Engineering a Compiler. 2nd edition. – USA: Elsevier, Inc, 2013 – С. 6 – 824 с.
2. *Golovanevsky O., Zaks A.* Struct-reorg: current status and future perspectives // Proceedings of the GCC Developers' Summit. 2007. С. 47-56.
3. *Chakrabarti G., Chow F.* Structure Layout Optimizations in the Open64 Compiler: Design, Implementation and Measurements // 2008 International Symposium on Code Generation and Optimization (CGO). 2008.
4. http://www.spec.org/ [Электронный ресурс]