УДК 004.4'422

Автоматическое распараллеливание рекурсивных процедур

М. А. Горелов

Московский физико-технический институт (государственный университет)

ЗАО «МЦСТ»

Для увеличения производительности современных вычислительных комплексов большинство производителей перешли на создание многоядерных архитектур. Однако большинство существующих приложений написаны для последовательного исполнения на одноядерных машинах. Таким образом, задача автоматического распараллеливания на уровне потоков приобретает все большую актуальность.

Большинство оптимизирующих компиляторов направлены на анализ и распараллеливание циклов и не умеют выявлять параллелизм на уровне вызовов процедур. Однако, существует класс рекурсивных алгоритмов, в процессе работы которых задача разбивается на множество меньших аналогичных подзадач, решается каждая из них, после чего результаты объединяются, и вычисляется решение исходной задачи. Подобные алгоритмы широко распространены и часто применяются для решения таких задач, как, например, сортировка, операции над матрицами и многих других. Подобные задачи имеют большой потенциал для автоматического распараллеливания, так как все подзадачи независимы и, следовательно, могут исполняться параллельно разными потоками.

Целью данной работы стало исследование возможности реализации автоматического распараллеливания рекурсивных вызовов в оптимизирующем компиляторе «Эльбрус». Было решено ограничиться случаем только рекурсий, так как в этом случае промежуточное представление вызываемой функции совпадает с представлением вызывающей функции, что значительно упрощает анализ.

Основная задача анализа диапазонов обращения процедур в память, представленного в работе, — определить независимость различных рекурсивных вызовов в процедуре на основании отсутствия между ними зависимости по данным или управлению. Для этого для каждой процедуры строятся два множества:

- Def — множество адресов, по которым процедура записывает данные;
- Use — множество адресов, по которым процедура читает данные.

Сначала они находятся для каждого узла управляющего графа процедуры, после чего результат обобщается на всю процедуру. Для массива эти множества представляются в виде диапазонов, где каждый диапазон представляет собой элемент вида

[l, u], где l и u — соответственно, его нижняя и верхняя границы. Границы диапазона задаются с помощью PS-форм, а также могут иметь неопределенное значение.

К диапазонам применимы все стандартные операции над множествами, такие как объединение множеств, пересечение и разность. В результате применения этих операций может получиться, что результирующее множество непредставимо в виде одного диапазона. Тогда используется список диапазонов. Для списков диапазонов также применимы все вышеперечисленные операции.

Когда известно, к каким переменным обращается процедура, чтобы найти диапазоны для рекурсивного вызова, нужно подставить его параметры в уже найденные def и use для процедуры. После подстановки параметров можно исследовать вызовы на независимость.

Исследование показало возможность автоматического распараллеливания рекурсивных процедур средствами оптимизирующего компилятора «Эльбрус». Анализ диапазонов обращения процедур в память был успешно реализован и протестирован на задаче быстрой сортировки.

Литература

1.1 *M. Gupta, S. Mukhopadhyay, N. Sinha.* Automatic parallelization of recursive procedures.

1.2 *R. Rugina, M. Rinard.* Automatic parallelization of divide and conquer algorithms