

<https://doi.org/10.32362/2500-316X-2019-7-5-7-19>



УДК 004.4'422

Система автоматического распараллеливания линейных программ для машин с общей и распределенной памятью

Ш.Г. Магомедов[@],
А.С. Лебедев

МИРЭА – Российский технологический университет, Москва 119454, Россия

[@]Автор для переписки, e-mail: msgg@list.ru

Эффективное программирование параллельных архитектур всегда было сложной задачей и особенно усложняется при их современном разнообразии. Задача автоматического распараллеливания программного кода была сформулирована с момента появления первых параллельных отечественных вычислителей (например, ПС2000). К настоящему времени разработаны языки и технологии программирования, которые упрощают работу программиста (Т-Система, MC#, Erlang, Go, OpenCL), но не делают распараллеливание автоматическим. Сложившаяся ситуация требует разработки и развития эффективных инструментов программирования вычислительных систем. Такие инструменты должны поддерживать разработку параллельных программ для систем с общей и распределенной памятью. В работе рассматривается задача автоматического распараллеливания линейных программ для таких систем. Обсуждаются разработанные методы вычисления пространственно-временных преобразований, оптимизирующих локальность программы. Рассматривается реализация методов на языке Haskell в рамках source-to-source транслятора, осуществляющего автоматическое распараллеливание. Осуществляется сравнение быстродействия параллельных программ lu, atax, syr2k, полученных с помощью разработанной системы и современного инструмента Pluto. Эксперименты проводились на двух машинах архитектуры x86_64, объединенных сетью InfiniBand. В качестве технологий распараллеливания использовались OpenMP и MPI. Быстродействие результирующей параллельной программы свидетельствует о практической применимости разработанной системы распараллеливания линейных программ.

Ключевые слова: автоматическое распараллеливание, линейные программы, модель многогранников, оптимизация локальности, линейное целочисленное программирование.

Для цитирования: Магомедов Ш.Г., Лебедев А.С. Система автоматического распараллеливания линейных программ для машин с общей и распределенной памятью // Российский технологический журнал. 2019. Т. 7. № 5. С. 7–19. <https://doi.org/10.32362/2500-316X-2019-7-5-7-19>

A tool for automatic parallelization of affine programs for systems with shared and distributed memory

Shamil G. Magomedov[@],
Artem S. Lebedev

MIREA – Russian Technological University, Moscow 119454, Russia

[@]Corresponding author, e-mail: msgg@list.ru

Effective programming of parallel architectures has always been a challenge, and it is especially complicated with their modern diversity. The task of automatic parallelization of program code was formulated from the moment of the appearance of the first parallel computers made in Russia (for example, PS2000). To date, programming languages and technologies have been developed that simplify the work of a programmer (T-System, MC#, Erlang, Go, OpenCL), but do not make parallelization automatic. The current situation requires the development of effective programming tools for parallel computing systems. Such tools should support the development of parallel programs for systems with shared and distributed memory. The paper deals with the problem of automatic parallelization of affine programs for such systems. Methods for calculating space-time mappings that optimize the locality of the program are discussed. The implementation of developed methods is done in Haskell within the source-to-source translator performing automatic parallelization. A comparison of the performance of parallel variants of lu, atax, syr2k programs obtained using the developed tool and the modern Pluto tool is made. The experiments were performed on two x86_64 machines connected by the InfiniBand network. OpenMP and MPI were used as parallelization technologies. The performance of the resulting parallel program indicates the practical applicability of the developed tool for affine programs parallelization.

Keywords: automatic parallelization, affine programs, polyhedral model, locality optimization, integer linear programming.

For citation: Magomedov Sh.G., Lebedev A.S. A tool for automatic parallelization of affine programs for systems with shared and distributed memory. *Rossiiskii tekhnologicheskii zhurnal* = Russian Technological Journal. 2019;7(5):7-19 (in Russ.). <https://doi.org/10.32362/2500-316X-2019-7-5-7-19>

Введение

Вычислительно емкие научные и инженерные приложения в большинстве своем тратят значительную часть процессорного времени на исполнение вложенностей циклов, часто удовлетворяющих свойствам линейных программ. Для анализа и преобразования таких программ существует модель многогранников [1], позволяющая применить техники распараллеливания и локализации данных с целью повышения быстродействия критических вычислительно емких участков.

Среди первых проектов, в которых были реализованы методы модели многогранников, выделяют LooPo [2] и PIPS [3]. Дальнейшее развитие методы модели многогранников нашли в современном source-to-source трансляторе Pluto [4, 5] и работах, фокусирующихся на агрегировании итераций в блоки для дополнительного улучшения временной и пространственной локальности вычислений [6, 7]. В работе [8] авторы применили методы машинного обучения для выбора оптимизирующих преобразований в модели многогранников, сделав шаг в сторону от распространенной практики применения

классических методов оптимизации. Практическая ценность методов модели многогранников для систем трансляции предметно-ориентированных языков (DSL) была показана в проекте Pencil [9], при этом остается актуальным традиционный вариант использования распараллеливающих систем: код, написанный на языке C или FORTRAN, оптимизируется для параллельного выполнения и компилируется для целевой архитектуры, о чем свидетельствуют направления развития open-source [4, 10, 11] и коммерческих компиляторов [12, 13].

Распараллеливание программ в модели многогранников осуществляется в пять этапов:

1. Анализируются зависимости по данным: потоковые зависимости (чтение после записи), антизависимости (запись после чтения), зависимости через выход (запись после записи).
2. Вычисляется многомерное расписание программы, обладающее параллелизмом и оптимизирующее временную локальность данных при вычислениях.
3. Вычисляется пространственное размещение, обладающее параллелизмом и оптимизирующее пространственную локальность данных при вычислениях.
4. Выполняется агрегирование итераций в блоки для дополнительного улучшения временной и пространственной локальности вычислений.
5. Генерируется параллельная программа согласно подходу [14] в соответствии с пространственно-временными преобразованиями, вычисленными на этапах 2–4.

Целью настоящей работы является практическая реализация ранее разработанных методов вычисления пространственно-временных преобразований [15–17] в рамках source-to-source транслятора линейных программ, написанных на языке C. Аналогом для сравнения служит инструмент Pluto – развитый source-to-source транслятор, реализующий методы [4, 5]. Вначале приводится понятийный аппарат модели многогранников, затем особенности реализации разработанной системы, после чего проводится сравнение системы с инструментом Pluto на примере трех алгоритмов линейной алгебры: LU-разложение квадратной матрицы, матричные произведения $atax$ и $syr2k$.

Понятийный аппарат модели многогранников

Модель многогранников – модель последовательных и параллельных вычислений, разработанная для анализа и преобразования линейных программ. Поиск и применение оптимизирующих преобразований осуществляется в рамках математической модели («полиэдрального представления») без непосредственных манипуляций с исходным текстовым представлением программ. Класс линейных программ описывается следующими ограничениями:

- в качестве исполнительных операторов используются только операторы присваивания, правая часть которых – арифметическое выражение;
- для описания повторяющихся операций используются только циклы с известным числом повторений (циклы DO языка FORTRAN или эквивалентные циклы for языка C); допускаются вложенности циклов произвольной структуры (плотные и неплотные); границы изменения счетчиков циклов задаются аффинными функциями от индексов циклов, обрамляющих данный цикл, и внешних переменных программы;
- допускается использование операторов ветвления, в условиях которых используются только аффинные функции от индексов циклов, обрамляющих оператор, и внешних переменных программы; не допускаются побочные выходы из циклов;