УДК 004.4'24

ПОДХОД К АВТОМАТИЧЕСКОМУ РАСПАРАЛЛЕЛИВАНИЮ ПОСЛЕДОВАТЕЛЬНЫХ АЛГОРИТМОВ ДЛЯ ЗАДАННЫХ КОНФИГУРАЦИЙ ГЕТЕРОГЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

А. Г. Волошко[1], А. Н. Ивутин[2], А. С. Новиков[3]

Тульский государственный университет, пр. Ленина, 92, 300012, г. Тула, Россия,
[1]atroshina@mail.ru, [2]alexey.ivutin@gmail.com, [3]alsnovikov@yandex.ru

Рассматриваются методы распараллеливания программного кода для гетерогенных систем на основе построения модели изначальной последовательной программы и ее модификации путем эквивалентных преобразований. В статье уделено внимание вопросам оптимального распределения инструкций по исполнителям.

Ключевые слова: распараллеливание; гетерогенная система; оптимизация; семантические связи; сети Петри.

APPROACH TO AUTOMATIC PARALLELIZATION OF SEQUENTIAL ALGORITHMS FOR GIVEN CONFIGURATIONS OF HETEROGENEOUS COMPUTING SYSTEMS

A. G. Voloshko[1], A. N. Ivutin[2], A. S. Novikov[3]

Tula State University, ave. Lenin, 92, 300012, Tula, Russia,
[1]atroshina@mail.ru, [2]alexey.ivutin@gmail.com, [3]alsnovikov@yandex.ru

The paper considers methods of paralleling program code for heterogeneous systems based on constructing a model of the original sequential program and its modification by equivalent transformations. The article pays attention to the issues of optimal distribution of instructions to executors.

Keywords: parallelization; heterogeneous system; optimization; semantic relations; Petri nets.

Современное состояние развития компьютерных систем можно охарактеризовать двумя словами: параллельность и гетерогенность. Последнее время задачи повышения производительности ЭВМ решаются за счет использования многопроцессорных и многоядерных технологий даже для персонального использования. Даже мобильные устройства давно [1] являются многоядерными. А повышенные требования к энергопотреблению устройств привело к появлению гетерогенных систем с разными ядрами в рамках одного процессора [2]. Решаемые на компьютерах задачи становятся все сложнее и требуют объединения их в большие гетерогенные сети, а также использование всех доступных вычислителей, в том числе и графических процессоров. Однако, создание параллельных программ с учетом гетерогенности и особенностей архитектуры вычислительной системы до сих пор остается сложной задачей, требующей высокой квалификации программиста не только в связи с необходимостью использования различных технологий, но и сложностью поиска участков кода, пригодных для распараллеливания.

В настоящее время существуют решения, позволяющие автоматизировать разработку параллельных программ для гетерогенных систем, например, Федорова А. [3] предлагает алгоритмы работы планировщика для распределения задач по асимметричным ядрам процессора, в работе [4] описан единый интерфейс для работы с технологиями OpenMP, CilkPlus, Intel TBB, PPL, BoostThreads, который используется для автоматического распараллеливания последовательного кода для указанных технологий, в [5] описывается реализация, которая отображает последовательную и неаннотированную С-программу в конвейерную реализацию, работающую на наборе FPGA, каждая из которых имеет несколько внешних блоков памяти и многие другие. Однако, не представлен общий подход к распараллеливанию, объединяющий технологии для разных типов гетерогенных систем.

Авторами данной статьи предлагается подход на основе моделирования программного кода в виде сети Петри с дополнительными семантическими связями (СПДСС) [6], что позволяет автоматически определять участки кода, доступные для распараллеливания и на основе эквивалентных преобразований формировать модель параллельной программы. Принципиально подход можно описать следующим образом:

1. Моделирование последовательного программного кода. На данном этапе определяется текущая последовательность выполнения действий и определяются все информационные (семантические) зависимости между операциями.
2. Анализ и построение параллельной модели программы выполняется на основе определенных на предыдущем этапе семантических связей между отдельными участками кода и строится минимально последовательная СПДСС, располагающая независимые друг от друга по семантике команды в параллельных секциях по управлению. При этом полагается, что доступно бесконечное количество исполнителей. На этом этапе также не учитывается архитектура гетерогенной системы, в том числе и возможность в принципе выполнять те или иные инструкции на заданном оборудовании.
3. Оптимизация параллельной программы под заданную гетерогенную систему. На данном этапе учитываются особенность архитектуры гетерогенной системы, определяются применяемые для распараллеливания технологии, производится выбор критерия оптимальности (время, энергоэффективность, баланс загрузки процессоров или решение многокритериальной задачи), а также формулируются дополнительные ограничения для оптимизации кода. Затем производится выбор метода оптимизации. Дело в том, что данная зада-

ча, относясь к NP-трудным, для точного решения требует полного перебора вариантов распределения операций по исполнителям, однако, зачастую с использованием ряда эвристик можно сократить область перебора в несколько раз, либо достаточно хорошее решение за приемлемое время. Результатом этого этапа является оптимальная или квазиоптимальная (в случае использование эвристических алгоритмов) модель параллельной программы для заданной гетерогенной системы.

4. Модернизация программного кода выполняется на основе сравнения первоначальной и оптимизированной модели. Добавляются соответствующие команды использования соответствующих технологий для формирования параллельных областей и синхронизации.

Следует отметить, что все эти этапы могут быть формализованы и автоматизированы. Для этого требуется разработка системы, состоящей из следующих модулей:

- средство анализа и трансляции программы в низкоуровневое представление, позволяющее наилучшим образом анализировать программы на параллельность и избавляющее от привязки к конкретному языку программирования. В качестве такого средства можно использовать систему LLVM;
- модуль формирования модели последовательного программного кода на основе анализа его низкоуровневого представления – создает СПДСС;
- модуль формирования минимально последовательной СПДСС реализует второй этап из представленных выше;
- модуль настройки параметров оптимизации – позволяет задать дополнительные ограничения, определить критерии оптимизации;
- модуль анализа целевой архитектуры гетерогенной системы – позволяет определить возможность и целесообразность использования отдельных исполнительных устройств для распараллеливания. Так, например, для нециклических параллельных областей не имеет смысл использовать графические процессоры и т. д.;
- модуль формирования оптимальной СПДСС – проводит оптимизацию с учетом заданных ограничений. При этом, для упрощения работы предложено использовать базу данных архитектур и применимых технологий распараллеливания, где хранятся параметры для компонентов вычислительной системы, их характеристики, возможные для применения технологии в зависимости от компонентов и их связей в системе и накладные расходы на распараллеливание для указанных типов архитектур. Кроме того, на основании нечеткого логического вывода производится выбор (если предварительно он не был осуществлен в модуле настройки параметров оптимизации) метода поиска оптимального решения. Результатом работы этого модуля является оптимизированная СПДСС;
- модуль обратного преобразования модели в программный код путем формирования дополнительных конструкций и команд, необходимых для распараллеливания в соответствии с используемой для данной архитектуры технологией;
- средство генерации исполняемой программы (также возможно использовать встроенные возможности системы LLVM).

Применение такого подхода позволяет автоматически анализировать и распараллеливать программы определенного класса (с циклами с известным числом повторений или с наличием семантически независимых задач в программе) для гетерогенных систем.

Библиографический список

1. Макаров Е. LG Optimus 2X: первый в мире двухъядерный смартфон с Tegra 2. URL: https://mobiltelefon.ru/post_1292483372.html (дата обращения: 20.09.2022).
2. Apple A11 Bionic. URL: https://en.wikichip.org/wiki/apple/ax/a11 (дата обращения: 20.09.2022).
3. Fedorova A. [et al.]. Maximizing power efficiency with asymmetric multicore systems // Communications of the ACM. 2009. Т. 52. №. 12. С. 48–57.
4. Venkat A. [et al.]. Automating wavefront parallelization for sparse matrix computations // Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE Press, 2016. С. 41.
5. Ziegler H. et al. Coarse-grain pipelining on multiple FPGA architectures // Proceedings. 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines. IEEE, 2002. С. 77–86.
6. Ivutin A. N., Troshina A. G. Semantic Petri-Markov nets for automotive algorithms transformations // 2018 28th International Conference Radioelektronika (RADIOELEKTRONIKA). IEEE, 2018. С. 1–6.

Исследование выполнено при финансовой поддержке РФФИ и Правительства Тульской области в рамках научного проекта 19-41-710003 р_а.