

Баканов В.М.

Программный инструментарий анализа тонкой информационной структуры программ по их информационным графам с целью выявления скрытого параллелизма и построения рационального плана выполнения параллельных частей программ

*Научно-Исследовательский Университет
Высшая Школа Экономики (НИУ ВШЭ),
Россия, Москва*

Предлагается программный инструментарий для выработки эффективных стратегий преобразования представлений информационных графов алгоритмов с целью выявления скрытого параллелизма и определения рационального плана (расписания) выполнения параллельных частей программ при учете ограничений реальных многопроцессорных вычислительных систем. Для достижения гибкости разработки сценариев преобразований представлений графа используется встроенный скриптовый язык Lua.

Ключевые слова: информационный граф алгоритма, тонкая информационная структура программы, скрытый параллелизм, ярусно-параллельная форма информационного графа, встроенный скриптовый язык программирования Lua, рациональная стратегия построения плана выполнения параллельной программы.

В настоящее время из путей повышения производительности вычислительных систем доступен метод параллелизации обработки данных; иные пути ограничены современным уровнем технологий и физическими законами.

Однако параллелизация (фактически одновременное = параллельное выполнение различных частей одной программы на независимо работающих вычислительных блоках) требует от алгоритмистов и программистов дополнительных усилий по выявлению в алгоритме участков, могущих быть выполненными параллельно; основное требование к таким блокам (*зернам* или *гранулам*) параллелизма состоит в независимости (ортогональности) их по данным.

Анализ программ (обычно именуемый исследованием их тонкой информационной структуры, [1]) по такому критерию непрост, поэтому говорят о скрытом (не фиксируемом при поверхностном рассмотрении) параллелизме; мощным инструментом при таком анализе является представление программ графовыми моделями. В России разработкой методов анализа структуры алгоритмов на основе графовых моделей занимались авторы работы [1]. В рамках улучшения подготовки инженерных кадров в Российской системе образования по приоритетным направлениям развития научно-технологического комплекса в НИУ ВШЭ по направлению обучения "Программная инженерия" используется разработанный программный инструментарий для анализа тонкой

информационной структуры программ по их информационным графам (ИГ) с целью выявления скрытого параллелизма и построения рационального плана выполнения параллельных частей программ.

ИГ алгоритма описывает исключительно информационные зависимости в алгоритме. Представление ИГ в Ярусно-Параллельной Форме (ЯПФ, *SPF – Stacked Parallel Form*) - одно из наиболее мощных средств выявления скрытого параллелизма. Высота ЯПФ (определяемая числом его ярусов) даёт общее время выполнения алгоритма, ширина ЯПФ – число задействованных отдельных (параллельно работающих) вычислителей многопроцессорной вычислительной системы (МВС). При этом необходимо рационально использовать ресурсы конкретной МВС. Эти ресурсы (число параллельно работающих вычислителей, объемы памяти на каждом и др.) всегда ограничены, поэтому задача планирования использования ресурсов очень важна. Формально это задача распознавания и оптимизации, причем конкретные критерии оптимизации определяются поставленной исследователем целью.

Реальные ЯПФ информационных графов обладают большой неравномерностью распределения числа операторов по ярусам, при этом использование ресурсов МВС очень нерационально - максимум эффективности использования ресурсов достигается при равномерном (или близком к равномерному) распределении операторов по ярусам.

При этом практически в любом ЯПФ имеется вариабельность в смысле перемещения вершин графа (операторов) между ярусами (величина этого перемещения ограничивается информационными зависимостями в ИГ); эта особенность дает возможность оптимизации ЯПФ (напр., в смысле балансировки - достижения наибольшей равномерности распределения операторов по ярусам). Достижение поставленной цели возможно с помощью различных стратегий перестановки операторов по ярусам; выбор оптимальной (рациональной) стратегии для ЯПФ различной сложности – интересная научно-практическая задача, имеющая прямое отношение к проблеме эффективного распараллеливания алгоритмов.

По данным [1] поставленная задача выработки рационального расписания выполнения параллельных частей программ относится к классу *NP*-полных (комбинаторная задача с ограничениями). С практической точки зрения необходимо иметь мощный и гибкий аппарат для нахождения приемлемых решений этой задачи (исходя из сказанного приемлемы эвристические методы).

Описываемый программный инструментариий написан на языке программирования C++ и является 32-битным приложением Windows [2]. Сценарий модификации ЯПФ реализуется на встроенном скриптовом языке Lua [3], причем исключительно применение Lua для реализации стратегий целенаправленного изменения ЯПФ ИГ позволило добиться большой гибкости и

эффективности в реализации поставленной цели. Версия системы доступна для свободного использования (адрес <http://vbakanov.ru/spf@home/>).

Оценка эффективности работы описываемого программного комплекса производилась на основе часто встречающихся в задачах программирования алгоритмов и разработанных стратегий преобразования их ЯПФ. Напр., стратегия БУЛЬДОЗЕР основана на перемещении операторов с наиболее нагруженных ярусов на недогруженные (так нож бульдозера срезает холмы и сдвигает их материал во впадины); критерий останова алгоритма – перебор всех операторов, могущих быть перенесенными “с холмов во впадины”), стратегия ДИХОТОМИЯ предполагает перенос половины находящихся на наиболее нагруженных ярусах операторов на вновь образованные пустые ярусы, созданные под данным; критерий останова – “ужатие” ЯПФ до ширины, не превышающей заданную. Критериями качества преобразования являлась неравномерность загрузки ярусов операторами (как в виде простого коэффициента неравномерности, так и в виде среднего квадратичного отклонения числа операторов на ярусах), критерием трудоемкости преобразования ЯПФ считалось число переносов операторов с яруса на ярус.

Анализ показывает, что описанные (не излишне изощренные) стратегии в самом деле позволяют достичь требуемого результата, однако с разной эффективностью для различных ИГ. Особенно интересна и практически важна задача априорного (до проведения собственно преобразований ЯПФ) предсказания уровня их эффективности (задача распознавания).

Данная разработка применяется в учебном процессе НИУ ВШЭ как в лекционных занятиях при изучении методов выявления скрытого параллелизма в произвольных алгоритмах в целях приобретения инженерных навыков разработки параллельных программ, так и при проведении научно-исследовательского семинара студентов по данной и близким тематикам.

Список литературы:

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2004. - 608 с.
2. Баканов В.М. Программа выявления скрытого параллелизма в информационном графе алгоритма и построения расписания выполнения параллельных частей алгоритма. // Свидетельство о государственной регистрации программ для ЭВМ № 2015617043. Федеральная служба РФ по интеллектуальной собственности (Роспатент), дата поступления 13.V.2015 г., дата регистрации 29.VI.2015 г.
3. Иерусалимски Роберту. Программирование на языке Lua. — М.: ДМК Пресс, 2014. - 382 с.

Bakanov V.M.

Software tool analysis of the fine structure information programs by their information graphs to identify parallelism and building management implementation plan parallel program parts

National Research University "Higher School of Economics" (HSE)

It offers software tools for the development of effective strategies to transform ideas information graphs algorithms to identify parallelism and determining management plan (schedule) of parallel parts of the program taking into account the limitations of the real multiprocessor systems. To achieve flexibility in the development of scenarios graph representations of transformations using embedded scripting language Lua.

Keywords: information graph algorithm, the thin structure of the program information, hidden parallelism, parallel-stacked form of the information graph, embedded scripting language Lua programming, rational strategy of building plan of the parallel program.