

Баканов Валерий Михайлович

д-р техн. наук, профессор

ФГАОУ ВО «Национальный исследовательский университет

«Высшая школа экономики»

г. Москва

DOI 10.31483/r-110815

**ВЫЧИСЛИТЕЛЬНЫЙ ПРАКТИКУМ ДЛЯ ПОСТРОЕНИЯ
РАЦИОНАЛЬНЫХ ПЛАНОВ (РАСПИСАНИЙ)
ВЫПОЛНЕНИЯ ПРОГРАММ НА ЗАДАННОМ НАБОРЕ
ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЕЙ**

Аннотация: в статье представлен авторский программный исследовательский комплекс ПРАКТИКУМ DF-SPF для анализа произвольных алгоритмов на формальное наличие естественного потенциала параллелизма и дальнейшего его, параллелизма, рационального использования в вычислительных практиках. Для целенаправленного преобразования специальных сечений информационного графа алгоритма используются эвристические методы, реализованные в форме API-вызовов на встроенном скриптовом языке Lua. Основными целевыми критериями являются достижение максимальной плотности кода, максимизация скорости выполнения программы и минимизация вычислительной трудоёмкости получения данного плана параллельного выполнения.

Ключевые слова: анализ информационной структуры программ, информационный граф алгоритма, ярусно-параллельная форма графа, эквивалентные преобразования графа, рациональные параметры плана параллельного выполнения программы.

Параллелизация обработки данных в настоящее время применяется для сокращения времени вычислений путём одновременной обработки данных по частям на множестве различных вычислительных устройств с последующим объединением полученных результатов и является основным средством реализации суперкомпьютерных вычислительных технологий [1]. Существенно при этом,

что параллельный вариант программы может быть выполнен большим числом отличающихся вариантов. Каждый вариант характеризуется своим планом (расписанием), задающим последовательность выполнения каждого блока (гранулы) команд программы для конкретного заданного поля параллельных вычислителей. При этом каждый план выполнения параллельной программы обладает рядом параметров, характеризующих качество выполнения общей программы (время выполнения, степень использования имеющегося оборудования, требования к исполняющему оборудованию и др.).

Таким образом, правомерно говорить о задаче многомерной оптимизации, целью которой является построение максимально эффективных (по выбранным критериям) планов выполнения параллельных программ. Решение подобных задач особенно актуально сейчас, когда разрабатываются серии отечественных процессоров, работа которых неэффективна без разработки качественного инструментального программного обеспечения.

Особый интерес представляет серия процессоров архитектуры VLIW (*Very Long Instruction Word*, сверхдлинная машинная команда). Для концепции VLIW размер гранулы параллелизма равен одной машинной команде, что соответствует представлению ILP (*Instruction-Level Parallelism*, параллелизм уровня машинных команд). В такой системе отсутствует аппаратное распараллеливание, а набор команд подготавливается соответствующими блоками компилятора или интерпретатора – стиль EPIC (*Explicitly Parallel Instruction Computing*, набор инструкций с явным параллелизмом). В рамках импортозамещения в России развивается серия микропроцессоров ЭЛЬБРУС, реализующих VLIW и EPIC-идеологию [2].

Известны целый ряд подходов к решению задачи разработки планов (расписаний) параллельного выполнения программ [3], однако основной проблемой их реализации является (в общем случае) присущая данной задаче *NP*-сложность построения [4].

Автором был разработан программный исследовательский инструмент ПРАКТИКУМ DF-SPF, предназначенной для как для формального нахождения

(скрытого, внутреннего для алгоритмов) параллелизма, так и для построения рациональных (разумных, стремящихся к оптимальным) планов (расписаний) выполнения программ, базирующихся на этих алгоритмах; программный комплекс описан в книге [5]. Общая схема и схема взаимодействия данных в разработанной системе приведена на рис. 1.

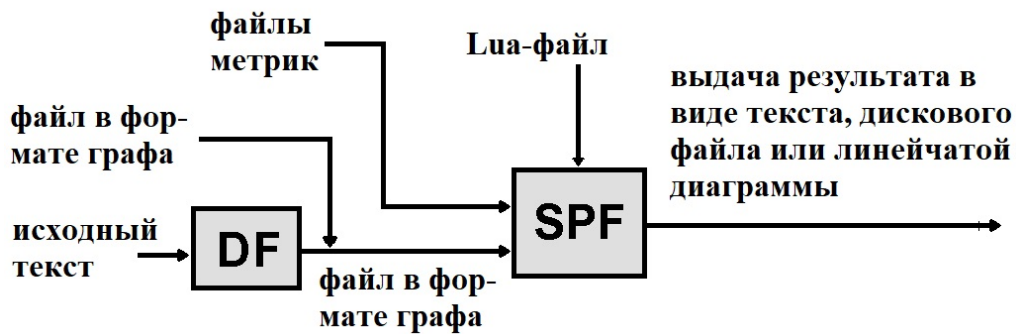


Рис. 1. Структурная схема и схема информационного взаимодействия компонентов исследовательского инструмента ПРАКТИКУМ DF-SPF

На вход программной системы поступает описание анализируемой программы в императивном виде в форме ассемблероподобного языка или формального его описания в виде ориентированного ациклического Информационного Графа Алгоритма (далее ИГА) – зависимость вида «операторы → операнды»; при этом вершины графа ассоциируются с операторами (группами операторов) программы, а дуги – с линиями передачи данных.

Исходной информацией для анализа является программа с отсутствием явных указаний на последовательности выполнения операторов, соответствующая архитектурной модели SMP (*Symmetric Multi Processing*, симметричная мультипроцессорность). Фактически имитационная модель DF представляет собой симулятор граф-машины с сохранением принципа единократного присваивания и возможностью контроля интенсивностью вычислений путём управления дисциплиной выборки готовых к выполнению команд на поле параллельных вычислителей.

Выявление и анализ внутреннего логического параллелизма в алгоритмах реализовано с использованием агентной модели (модуль DF) и универсального

анализатора/преобразователя алгоритмов в форме информационных графов (модуль SPF). Неотъемлемой частью функционала модуля SPF является блок построения специальных сечений ИГА в виде его Ярусно-Параллельной Формы (далее ЯПФ), хотя реализованный языковыми средствами API (*Application Programming Interface*, интерфейс программирования приложения) встроенного языка потенциально предоставляет возможность осуществить любой произвольный метод анализа и преобразований алгоритмов. Оба упомянутых модуля разработаны с использованием языка C/C++ в стиле GUI для модели Win'32, являются полностью Open Source и оба могут быть выгружены для свободного использования (формат инсталляционных файлов: http://vbakanov.ru/dataflow/content/install_df.exe, http://vbakanov.ru/spf@home/content/install_spf.exe).

Вычислительные эксперименты проводятся над набором оформленных в виде библиотеки программ, реализующих наиболее часто применяющиеся алгоритмы (напр., линейной алгебры; библиотека может неограниченно расширяться усилиями участников исследований). Условность выполнения операторов реализуется *предикатным методом*, что позволяет избежать мультивариантности ЯПФ; программные циклы перед выполнением разворачиваются («*unrolling*») с использованием макросов.

Для получения рациональных планов параллельного выполнения служат целенаправленные эквивалентные (не нарушающие информационных связей в алгоритме) преобразования ЯПФ, описываемые с использованием скриптового языка Lua [6]. Основой создания таких сценариев является эвристический подход совместно с итерационным методом движения в сторону повышения качества разрабатываемых планов параллельного выполнения программ. Рассматривая сущность «операторы» в программном модуле SPF как группы команд любого размера, этот модуль может быть использован и при составлении планов параллельного выполнения программ с гранулами параллелизма значительно большего размера – напр., при проведении занятий эта возможность использовалась автором при планировании вычислений на кафедральном инструментальном вычислительном кластере архитектуры MPP (*Massive Parallel Processing*,

массово-параллельная обработка) при наличии крупномасштабного параллелизма уровня данных.

Автором данный исследовательский комплекс успешно ряд лет используется при проведении работ (в частности, научно-исследовательского характера) самого различного уровня – от формального выявления параллелизма в произвольных алгоритмах (определения его основных параметров – напр., зависимости от размерности обрабатываемых данных) до решения сложных оптимизационных задач по целенаправленной модификации сечений графов алгоритмов под конкретные параллельные вычислительные структуры (включая гетерогенные).

Особенно творчески эффективной для учащихся оказалась многовариантность применения API-вызовов Lua при реализации конкретных эвристических методов целенаправленного преобразования алгоритмов. В ходе выполнения работ учащиеся осваивают основные принципы генерации и оптимизации методов проектирования рациональных (разумных) планов параллельного выполнения алгоритмов (и реализованных на их основе программ). Это особенно важно для будущих разработчиков трансляторов и виртуальных машин, исследователей свойств алгоритмов (в направлении нахождения рационального использования потенциала скрытого их параллелизма). На практике информационный граф алгоритма без труда восстанавливается на основе анализа причинно-следственных связей в программе после завершения работы компилятора; при этом логично библиотечные процедуры обрабатывать (компоновать) отдельно.

Исследовательский ПРАКТИКУМ DF-SPF и разработанные с его помощью методики (приёмы выявления скрытого параллелизма и его параметров в произвольных алгоритмах, способы построения рациональных планов выполнения параллельных программ на заданном поле вычислителей) ряд лет применяются при обучении студентов в университетах России и дали возможность повысить компетенции учащихся в области теории и практики параллельной обработки данных.

Список литературы

1. Воеводин В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин. – СПб.: БХВ-Петербург, 2004. – 608 с.
2. Архитектура микропроцессоров ЭЛЬБРУС [Электронный ресурс]. – Режим доступа: http://www.elbrus.ru/elbrus_arch (дата обращения: 31.03.2024).
3. Федотов И.Е. Параллельное программирование. Модели и приёмы / И.Е. Федотов. – М.: Солон-Пресс, 2018. – 390 с.
4. Гэри М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. – М.: Мир, 1982. – 416 с.
5. Баканов В.М. Практический анализ алгоритмов и эффективность параллельных вычислений / В.М. Баканов. – М.: Пробел-2000, 2023. – 198 с. [Электронный ресурс]. – Режим доступа: <https://clck.ru/39sdgk> (дата обращения: 31.03.2024).
6. Иерузалымски Роберту Программирование на языке Lua / Роберту Иерузалымски. – М.: ДМК Пресс, 2014. – 382 с.