УДК 004.272 + 004.89

ПОСТРОЕНИЕ РАЦИОНАЛЬНЫХ ПЛАНОВ ВЫПОЛНЕНИЯ ПАРАЛЛЕЛЬНЫХ ЧАСТЕЙ ПРОГРАММ И ПРИМЕНЕНИЕ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ РЕШЕНИЯ ПОДОБНЫХ ЗАДАЧ

В.М.Баканов (*vbakanov@hse.ru*)
Национальный Исследовательский Университет
"Высшая Школа Экономики" (НИУ ВШЭ), Москва

Аннотация: Рассматриваются подходы к проблеме и программная система для разработки алгоритмов построения рациональных планов выполнения параллельных частей программ на основе их графовых представлений при учете ограничений реальных многопроцессорных вычислительных систем. Для достижения гибкости разработки алгоритмов эквивалентных преобразований информационных графов используется встроенный скриптовый язык программирования Lua. Показывается целесообразность применения методов искусственного интеллекта при решении целевой задачи.

Ключевые слова: Информационный граф алгоритма, Анализ тонкой информационной структуры алгоритма, Скрытый параллелизм, Ярусно-параллельная форма информационного графа, Эквивалентные преобразования представления информационного графа, Встроенный скриптовый язык программирования Lua, Стратегия построения рационального плана выполнения параллельной программы, Методы искусственного интеллекта.

Введение

В настоящее время из путей повышения производительности вычислительных систем доступен метод параллелизации обработки данных; путь повышения тактовой частоты ограничивается

фундаментальными законами (закон Рэлея, 1871) и технологией производства процессоров.

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, которые могут быть выполнены параллельно; основное требование к таким блокам параллелизма состоит в независимости (ортогональности) их по данным.

Обзор исследований в данной области

Анализ алгоритмов 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 нашей стране разработкой методов анализа структуры алгоритмов на основе графовых моделей занимались авторы [Воеводин и др., 2002].

Ситуация усложняется необходимостью рационального использования ресурсов (число и возможности параллельных вычислителей) конкретной многопроцессорной вычислительной системы (МВС), которые всегда ограничены; именно поэтому задача планирования использования ресурсов столь важна. Формально это задача распознавания и оптимизации, причем конкретных критериев оптимизации может быть много (в зависимости от поставленной исследователем цели).

При представлении программы информационным графом (вычислительная модель “операторы - операнды”) вершины графа соответствуют преобразователям информации (операторам), а дуги - информационным связям операторов (данным). В данном случае термин "операторы" в известной степени условен, ибо под "операторами" можно понимать отдельные последовательности вычислений любого (в принципе) размера (в зависимости от соглашений, принятых при декомпозиции исходной программы). Свойства информационного графа алгоритма (ИГА): ацикличность, направленность, детерминированность. Информационные графы (наряду с иными формами графов) широко применяются для анализа программ [AlgoWiki, 2016].

Удобным и часто используемым методом формального выявления параллелизма является представление ИГА в ярусно-параллельной форме

(ЯПФ); при этом операторы, могущиеся выполняться независимо друг от друга (фактически параллельно), располагаются на одном ярусе.

Глубина ЯПФ (определяемая через число ярусов) определяет общее время выполнения алгоритма, ширина ЯПФ – максимальное число задействованных отдельных (параллельно работающих) вычислителей (напр., отдельных ядер процессора) данной МВС. На рис. 1 приведёна ЯПФ графа простой процедуры вычисления вещественных корней полного квадратного уравнения (*каноническая* ЯПФ, вследствие простоты алгоритма соответствие вершин графа вычислительным операциям не приводится).

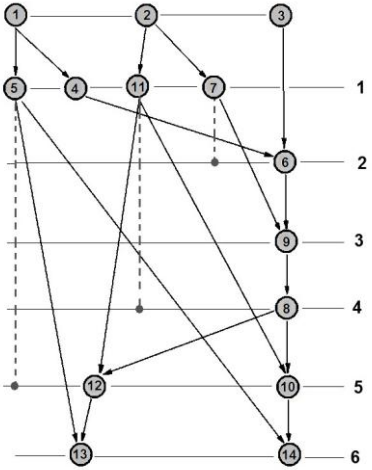

Рис.1. Ярусно-параллельная форма процедуры вычисления вещественных корней полного квадратного уравнеия (вершины 1,2,3 - входные данные, 13 и 14 - вычисление выходных данных), цифры крайние справа - номера ярусов ЯПФ (пунктиром показан диапазон возможного расположения операторов по ярусам)

Формально каноническая ЯПФ уже представляет собой некоторый план выполнения отдельных частей программы (номера ярусов задают последовательность выполнения параллельных групп операторов на данном ярусе), однако подобное расписание реализуемо лишь на гипотетической МВС с бесконечно большим числом параллельных вычислителей (концепция *неограниченного параллелизма*). Реальные ЯПФ

информационных графов обладают большой неравномерностью в распределении числа операторов по ярусам (выпуклый пример - применяемый для ассоциативных операций граф сдваивания), при этом использование ресурсов МВС нерационально - с точки зрения выполнения каждой программы максимум эффективности использования ресурсов обычно достигается при равномерном (или близком к равномерному) распределении операторов по ярусам.

Применяемые при исследовании методы

Однако в любом ЯПФ имеется *вариативность* в расположении вершин графа (операторов) между ярусами (величина перемещения операторов между ярусами ограничивается соблюдением информационных зависимостей операторов в графе; для ЯПФ графа на рис. 1 оператор 5 может быть перемещен "вниз" на любой ярус с 2 по 5-й, оператор 11 - на любой ярус с 2 по 4-й, оператор 7 - на ярус 2). Эта особенность ЯПФ даёт возможность оптимизации ЯПФ (в смысле, напр., достижения наибольшей равномерности распределения операторов по ярусам); в случае рис. 1 возможна "разгрузка" яруса 1 от операторов 14 и 2, в результате приведенная программа может быть выполнена на 2 (вместо 4-х) параллельно работающих вычислителях за то же время.

Данная задача близка к проблеме разбиения графов и согласно [Гери и др., 1982] относится к классу *NP*-полных. Для ИГА большого размера трудоёмкость оптимизации путем перебора запредельно велика (задача перестановок с учетом влияния изменения конфигурации ЯПФ после каждой), существующие алгоритмы разбиения графов являются эвристическими. С практической точки зрения необходимо иметь гибкий аппарат для решения задачи планирования с разной степенью оптимизации (вследствие *NP*-полноты задачи понимая эту процедуру как последовательное повышение качества эвристик в сторону улучшение критерия оптимизации) - от "быстрых" методов (основанных, напр., на использовании ЯПФ) до более трудоемких (методы "ветвей и границ", эволюционные и др.) для получения лучших решений; использование подходов искусственного интеллекта (ИИ) в данном случае оправдано и должно привести к улучшению результатов.

Т.о. решаются (*последовательно* выполняемые) задачи:

а) Выявление в заданном алгоритме собственно наличие параллелизма (для выполнения этого подходит, напр., метод представления ИГА в ЯПФ).

б) Планирование параллельного выполнения алгоритма с учётом реалий (ограничений) конкретной многопроцессорной вычислительной системы (МВС).

Ограничения МВС - число параллельно работающих вычислителей, объемы их памяти и др., выполнение программы с учётом этого достигается эквивалентными (не изменяющими информационных зависимостей в графе) преобразованиями ЯПФ-разбиений ИГА.

Будем использовать следующие параметры ЯПФ графа:

- H - глубина ЯПФ графа, определяется количеством ярусов
- W_i – ширина i-того яруса (количество операторов на нём)
- W – ширина ЯПФ графа, определяется максимумом операторов по всем ярусам; $W = max(W_i)$
- $\bar{W}$ – средняя ширина ЯПФ; $\bar{W} = \frac{1}{N}\sum_{i=1}^{i=N} W_i$, N - число ярусов ЯПФ
- k - степень (коэффициент) неравномерности распределения операторов по ярусам ЯПФ графа; $k = max(W_i)/min(W_i)$

Типовые постановки оптимизационной задачи (обычно реализовываются комбинации):

а) Максимально полная загрузка неизвестного заранее (но не меньшего $\bar{W}$) числа параллельно работающих вычислительных узлов имеющейся МВС при минимальном времени выполнения задачи: $k \to 1{,}0$ при $H = H_0 = const$ (где H_0 - длина критического пути - минимальная из всех возможных высот у ЯПФ данного графа).

б) Наиболее полное использование ресурсов (вычислительных узлов) имеющейся МВС при допущении увеличения времени выполнения (отход от канонической ЯПФ): $max(W_i) \leq W_0$ при $H \geq H_0$ (здесь W_0 - число доступных параллельно работающих вычислителей в заданной МВС); однако желательно $H - H_0 \to min$ (минимизация времени выполнения).

Для реализации стратегий построения планов выполнения параллельных частей программ использован встроенный скриптовый язык программирования (ВСЯП) Lua [Иерузалимски, 2014] главным образом вследствие его компактности (возрастание размера исполняемого файла родительского приложения не более чем на ≈100 килобайт) и простоты обучаемости (в основном Lua следует синтаксису языка C); широкие возможности (поддерживается объектно-ориентированная модель программирования и др.) и полная бесплатность исходных текстов только

подтверждают правильность выбора. В целом возможностей Lua вполне достаточно для реализации наиболее трудоемких методов оптимизации задачи планирования. Клиентская часть системы (предварительное сообщение [Баканов, 2016]) написана на языке программирования C/C++ и является 32-битным GUI-приложением Windows, в связи с предполагаемым большим числом экспериментов рассматривается интегрирование ее в инфраструктуру BOINC (*Berkeley Open Infrastructure for Network Computing*) и расположение на сервере ВУЗ'а.

Система API-вызовов разработанной системы позволяет реализовывать любой из перечисленных критериев оптимизации (а также их комбинации), т.к. выбор критерия осуществляет собственно пользователь (на основе задач, им поставленных). Программный интерфейс рассматриваемой системы включает три типа API-вызовов (всего около 30 штук, причём каждый является Lua-"оберткой" соответствующей C-функции):

- Информационные (служат для получения информации о ИГА, его ЯПФ и параметрах вычислительного поля; эти данные являются *фактами* для последующей выработки стратегий разработки расписания с использованием методов ИИ).
- Акционные (после применения *правил* служат для реализации конкретных стратегий решения задачи построения расписания).
- Вспомогательные (вывод рассчитанных данных в текстовом и графическом виде, работа с файловой системой и т.п.).

Примерами информационных вызовов являются - получить характер распределение ширин ярусов ЯПФ, диапазон ярусов допустимого расположения каждого оператора, число и параметры параллельных вычислителей и др., акционных – создать ЯПФ по указанному ИГА, переместить заданный оператор на конкретный ярус ЯПФ (с учетом ненарушения информационных зависимостей), создать новый пустой ярус, уничтожить пустой ярус и др. В процессе выполнения акционных функций собственно ИГА не изменяется (т.е. программа остаётся той же самой) - меняется лишь ЯПФ-представление графа; выполнение API-функций подробно протоколируется в файле с уникальным именем. Данные расчетов выдаются как в текстовом, так и в графическом (ленточная диаграмма распределения операторов по ярусам ЯПФ) виде и могут быть сохранены в распространенных форматах.

В случае применение данной системы в качестве компоненты распараллеливающего компилятора ИГА априорно существует

(компилятор при работе производит анализ программы на информационные зависимости - т.е. фактически строит ИГА). Автор для получения корректного ИГА ряд лет использует программный симулятор вычислителя потоковой (DATA-FLOW) архитектуры [Баканов, 2015], в котором программа отлаживается и в дальнейшем экспортируется в файловый формат списка смежных вершин. Особенностью этого симулятора является возможность моделирования асинхронного режима выполнения операторов (для каждого задается время исполнения в условных единицах).

Одна из наиболее сложных задач – определение наиболее эффективных стратегий балансировки ЯПФ информационного графа алгоритма (здесь под балансировкой понимается равномерность распределения операторов по ярусам ЯПФ). Описанные на ВСЯП стратегии могут реализовывать любые алгоритмические методы оптимизационных вычислений на графах (совсем необязательно использующие возможности ЯПФ).

Задача имеет прямое отношение к проблеме эффективного распараллеливания алгоритмов и, в частности, к вопросу наиболее эффективного использования ресурсов многопроцессорных вычислительных систем (МВС). На первой стадии из всех реалий конкретной МВС учитывается только число параллельных вычислителей; разрабатываются методы учета гетерогенности (по параметрам объема памяти, возможностей вычислений с различными типами данных, векторных вычислений и т.п.) вычислительного поля с возможностью выбора выполнения определенного оператора на соответствующем по параметрам вычислителе (для реализации этого режима понадобилась разработка около 20 дополнительных API-вызовов).

В качестве *параметра эффективности* собственно стратегии преобразования ЯПФ предлагается использовать число перестановок (аналогично задачам сортировки) операторов с яруса на ярус для получения заданного результата (минимизируется), штрафной функцией может быть превышение высоты ЯПФ над величиной критического пути.

Для каждого расписания выполнения операторов становятся известными затраты времени на передачи данных между отдельными вычислителями (разумным дополнительным критерием оптимизации является требование минимизации времени пересылок данных - при этом необходима информация о зависимости времени обмена от размеров сообщения для коммуникационной среды данной МВС).

Результаты, их обсуждение и перспективы

Эффективность некоторых разработанных (и реализованных на языке Lua) стратегий проверялась на распространенных алгоритмах, неравномерность распределения числа операторов по ярусам ЯПФ оценивалась неравномерностью распределения операторов по ярусам k, дополнительно определялась величина среднеквадратичного отклонением ширин ярусов $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{i=N}(W_i - \overline{W})^2}$. Эти (не являющимися излишне изощренными) стратегии в самом деле позволяют достичь требуемого результата (наблюдается как снижение неравномерности ширин ЯПФ, так и среднеквадратичного отклонения ширин ярусов ЯПФ), однако с разной эффективность для различных ИГА. В целом наблюдается повышение эффективности стратегии при увеличении сложности (и, соответственно, вариативности) ИГА. Особенна интересна и практически важна задача априорного (до проведения собственно преобразований ЯПФ) предсказания уровня эффективности стратегий (задача распознавания). Большего эффекта можно добиться путем совершенствования стратегий и разумного их совместного применения.

Для получения наилучших результатов при планировании выполнения параллельных частей программ планируется использовать подходы ИИ. В самом деле, данная задача является эвристической и сводится к выбору наиболее эффективных стратегией решения (критерии оптимизации предложены ранее). Для каждого типа алгоритмов на основе фактов (полученных с помощью информационных функций API) требуется (с использованием эмпирических правил) выбрать определенную стратегию решения заданной проблемы (общая схема рис. 2).

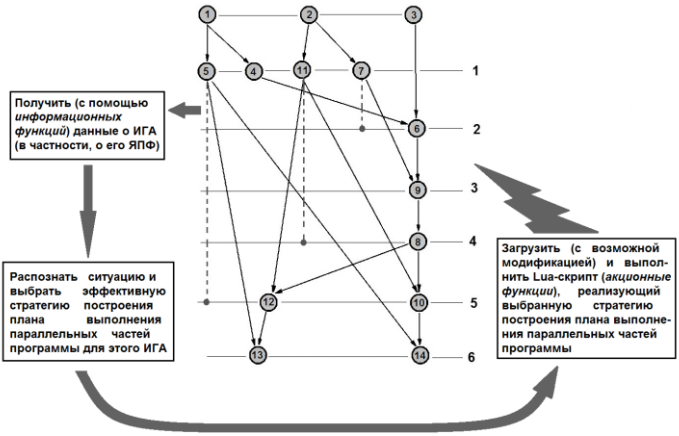

Рис.2. Этапы работы с программной системой разработки алгоритмов для составления плана выполнения частей параллельных программ с использованием методов искуственного интеллекта

Заключение

Применение встроенного скриптового языка программирования для моделирования методов оптимального планирования задач для параллельных вычислительных систем позволило добиться гибкости и эффективности в реализации поставленной цели, что было бы сомнительным при использовании иных подходов. Использование методов искусственного интеллекта призвано автоматизировать данный процесс и привести к получению лучших результатов. Результаты исследований предназначены для анализа алгоритмов (в плане выявления потенциала распараллеливания и эффективности выполнения), при разработке распараллеливающих компиляторов и в процессе обучения специалистов в области технологий параллельного программирования.

Список литературы

[AlgoWiki, 2016] AlgoWiki. Открытая энциклопедия свойств алгоритмов. [Электронный ресурс]. Дата обновления: 03.09.2015. URL: http://algowiki-project.org (дата обращения: 09.05.2016).

[Баканов, 2015] Баканов В.М. Управление динамикой вычислений в процессорах потоковой архитектуры для различных типов алгоритмов. // Журнал "Программная инженерия", 2015, № 9, с. 20-24.

[Баканов, 2016] Баканов В.М. Программный инструментарий анализа информационной структуры алгоритмов по их информационным графам. // Параллельные вычислительные технологии (ПаВТ'2016): труды международной научной конференции (г.Архангельск, 28.03–01.04.2016). —Челябинск: Издательский центр ЮУрГУ, 2016, с. 432-441.

[Воеводин и др., 2002] Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. — 608 с.

[Гери и др., 1982] Гери М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. — 416 с.

[Иерузалимски 2014] Иерузалимски Роберту. Программирование на языке Lua. — М.: ДМК Пресс, 2014. — 382 с.

CONSTRUCTION OF RATIONAL IMPLEMENTATION PLANS OF PARALLEL PARTS OF THE PROGRAM USING ARTIFICIAL INTELLIGENCE TO SOLVE SUCH PROBLEMS

V.M.Bakanov (e881e@mail.ru, vbakanov@hse.ru)
National Research University "Higher School of Economics" (HSE), Moscow

101000, Moscow, Myasnitskaya str., 20. 8-495-771-3232, 8-495-531-0031

Abstract: The approach to the problem and to develop a software system based on the construction of the rational parts of parallel programming plans of their algorithms of graph representations taking into account the limitations of the real multiprocessor systems. To achieve flexibility in the development of algorithms for equivalent transformation of information using graphs embedded scripting Lua programming language. We show the feasibility of artificial intelligence methods in the target.

Keywords: Information graph algorithm analysis algorithm thin information structure, parallelism, parallel form information graph, Equivalent transformations submission information graph, Built-in scripting Lua programming language, build management strategy implementation plan of the parallel program, methods of artificial intelligence.

==========

Тематика доклада: прикладные интеллектуальные системы

Баканов Валерий Михайлович, д-р техн. наук, профессор Национального Исследовательского Университета "Высшая Школа Экономики" (НИУ ВШЭ), член РАИИ, http://www.hse.ru/staff/vbakanov#teaching
E-Mail: e881e@mail.ru, vbakanov@hse.ru

Дом. адрес: 117461, Москва, Херсонская ул., д.12/5, кв.533.
Тел. дом. 8-499-122-1328.