ОТОБРАЖЕНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ АЛГОРИТМА ЦИКЛИЧЕСКОЙ ПРОГОНКИ НА МОДЕЛЬ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Л.В. Логанова

Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет) (СГАУ), Самара, Россия

Предложенная методика отображения математической модели алгоритма циклической прогонки на модель вычислительной системы позволяет выполнить анализ эффективности согласования выбранного алгоритма с гетерогенной вычислительной средой и предсказать эффективность вычислений по алгоритму для решения совокупности СЛАУ.

Ключевые слова: метод циклической прогонки, математическая модель, алгоритм, вычислительная система.

Введение

Моделирование физических процессов с помощью методов конечных разностей и конечных элементов зачастую сводится к необходимости решения системы линейных алгебраических уравнений (СЛАУ) с матрицей ленточного вида. Причем круговые, сферические и цилиндрические физические подобласти характеризуются матрицами, содержащими, кроме ленты на центральных диагоналях, ненулевые элементы в верхнем правом и левом нижнем углах. Переход к многомерным задачам сопряжен с заметным ростом вычислительной сложности. Именно этот фактор определяет растущую потребность в использовании высокопроизводительных вычислительных систем и, как следствие, в синтезе алгоритмов, учитывающих такую архитектуру.

Для создания эффективных параллельных алгоритмов решения СЛАУ с ленточными матрицами традиционно используются следующие прямые методы: прогонки (Томаса), циклической редукции и декомпозиции области. Прогонка в отличие от циклической редукции характеризуется меньшим объёмом коммуникаций, в отличие от декомпозиции области более низкой вычислительной сложностью[1,2]. Это и определило выбор автора в пользу применения метода Томаса для решения СЛАУ с матрицей ленточного вида. При этом переход к рассмотрению многомерных сеточных областей и, как следствие, к совокупности СЛАУ, позволяет снять ограничение на количество процессоров, используемых для решения ленточных СЛАУ. Однако обычная прогонка не применима для решения сеточных уравнений неявных разностных схем с периодическими граничными условиями. Поэтому в данной работе рассматривается метод циклической прогонки[3].

Зачастую, при выборе численного метода в тени остаются вопросы, связанные с архитектурой аппаратных средств и программным обеспечением, используемых для его реализации. Приступая к синтезу параллельных алгоритмов, необходимо выделить те общие ограничения, продиктованные особенностями выбранной архитектуры вычислительной системы (ВС), которые могут потребовать изменений структуры алгоритма для его эффективной реализации[4]. Соответствующее направление исследований, получившее название «отображение проблем вычислительной математики на архитектуру вычислительных систем»[5] было сформировано еще в 70-е годы прошлого столетия академиком Г.И. Марчуком.

В настоящее время с активным внедрением в повседневную практику GPGPU (GPU общего назначения) интерес к данной тематике многократно возрос.

В данной работе предложена методика отображения математической модели алгоритма циклической прогонки на модель вычислительной системы. При этом появляются возможности проведения анализа эффективности согласования указанного алгоритма с гетерогенной вычислительной средой и предсказания результатов вычислений по алгоритму для решения совокупности СЛАУ на ВС, содержащей графические вычислительные устройства. Авторский подход к построению математических моделей алгоритма циклической прогонки и ВС, включает развитие подходов, представленных в работах [6 -10].

Математическая модель алгоритма, основанного на методе циклической прогонки

Рассмотрим модель алгоритма циклической прогонки, в основе которой лежит графовое представление G(V, E) (рисунок 1), где V – множество вершин, соответствующих блоку операций, E – множество дуг, построенных с учетом информационных зависимостей между вершинами [6-8]. При решении одной СЛАУ он может иметь только последовательную реализацию на ВС.

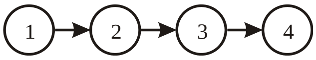

Рис. 1. Макрограф алгоритма циклической прогонки (1 - определение прогоночных коэффициентов, 2 – вычисление значений промежуточных сеточных функций, 3,4 – поиск значений искомой сеточной функции)

Для представления модели алгоритма решения совокупности СЛАУ, следует обратить внимание на наличие в алгоритме вложенных циклических конструкций (например, нахождение прогоночных коэффициентов для нескольких СЛАУ). В этом случае модель следует рассматривать как совокупность подграфов нахождения прогоночных коэффициентов, значений промежуточных и искомых сеточных функций.

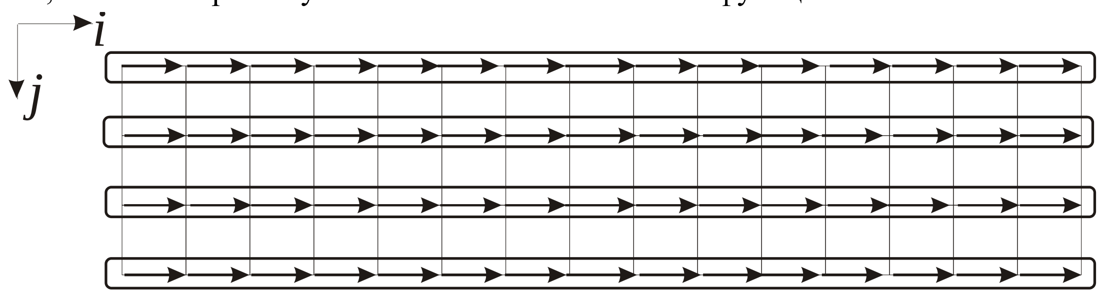

Рис.2. Граф нахождения прогоночных коэффициентов α,β,γ

Аналогично строятся графы нахождения значений промежуточных и искомых сеточных функций.

Отображение модели алгоритма на модель вычислительной системы

Модель ВС, включающая GPU также имеет графовое представление S=(C, L), где C – множество процессоров, L – множество ребер, связывающих вычислительные устройствах[9,10].

Отображение графа алгоритма G(V, E) на S=(C, L), обозначим как $\varphi : V \to C$. Представим его матрицей X={xij, где xij = 1, если φ(vi) =pj, и xij = 0 иначе, vi ∈Vk, Vk∈V, pj ∈C} и введём следующие функции:

ρ(vi) – число операций по алгоритму, соответствующих i этапу вычислений по алгоритму; ρ(pj) – производительность вычислителя; bw – ширина шины памяти; β - частота шины памяти; δ - полоса пропускания; δ(eij) – количество пересылок; tl(lij) - латентность.

Оценка времени выполнения вычислительных операций с учетом синхронизации составит:

$$T^{app} = \max_{V_k} \sum_{\mathrm{v}_i \in V_k} \tau_m \cdot x_m \cdot \frac{\rho(\mathrm{v}_i)}{\rho(p_j)},\ где\ V_k \in V\ ,$$
$$p_j \in C,\ \tau_m = \begin{cases} 0, \tau(\mathrm{v}_i) \neq \tau(p_j) \\ 1, \tau(\mathrm{v}_i) = \tau(p_j) \end{cases} \qquad (1)$$

Время накладных расходов

$$T_{lGPU} = \max_{E_i}\Big(\sum_{e \in Ei} (t_l(l_{ij})\delta(e_{ij}) + \tau_i \cdot x_i \frac{\delta(e_{ij})}{b_w \cdot \beta})\Big) \qquad (2)$$

$$T_{lMEM} = \max_{V_k} \frac{\sum_{\mathrm{v}_i \in V_k} \delta(\mathrm{v}_i)}{b_w \cdot \beta} \qquad (3)$$

$$T^o = T_{lGPU} + T_{lMEM} \qquad (4)$$

Следует обратить внимание, что один и тот же объём данных может считываться и записываться за различное количество обращений, которое зависит от организации структур хранения.

$$T = T^{app} + T^o \qquad (5)$$

При выполнении вычислений в гетерогенной среде необходимо дополнительно произвести оценку времени выполнения вычислений на CPU.

Реализация алгоритма циклической прогонки на одном GPU

При реализации алгоритма на GPU каждый процессор решает одну и ту же задачу, только над разными данными, т.о. несколько СЛАУ решаются одновременно[11]. Вычисления по алгоритму организованы с учетом модели алгоритма и архитектуры ВС [12]. Вследствие чего каждая нить вызываемого единственного ядра выполняет решение одной СЛАУ в соответствии с формулами метода циклической прогонки[3,11].

Произведём оценку времени выполнения вычислений по алгоритму с помощью модели, предложенной выше. Размеры исследуемой сеточной области (MxN) выберем с учётом ограничений на память GPU и обеспечения полной или близкой к полной загрузке каждой нити.

Для подтверждения адекватности отображения модели алгоритма на модель ВС были проведены вычислительные эксперименты. При этом рассматривалась вычислительная система с видеокартой GeForce GTX 470 , CPU Intel Core 2 Duo E8500 3.16 ГГц, операционной системой Microsoft Windows XP с установленным драйвером NVIDIA CUDA Version 2.3. Обмен данными между потоками CPU реализован средствами OpenMP, между памятью GPU и CPU осуществляется соответствующими функциями CUDA Runtime API [13 - 15] через PCI-Express x16, скорость передачи которого 4 Гб/с. Предложенные алгоритмы реализованы на языке C с расширениями средствами программного пакета Microsoft Visual Studio 2008.

При работе с глобальной памятью, обладающей высокой латентностью, для повышения скорости доступа к ней использовалась возможность объединения нескольких запросов, для чего элементы векторов, необходимых каждой нити в некоторый момент, располагались последовательно.

Результаты сравнения, полученные экспериментальным путём и на модели, представлены на рисунке 3. Разница значений на графиках объясняется, возможно, тем, что заявленные Nvidia характеристики графического устройства на практике не всегда достижимы.

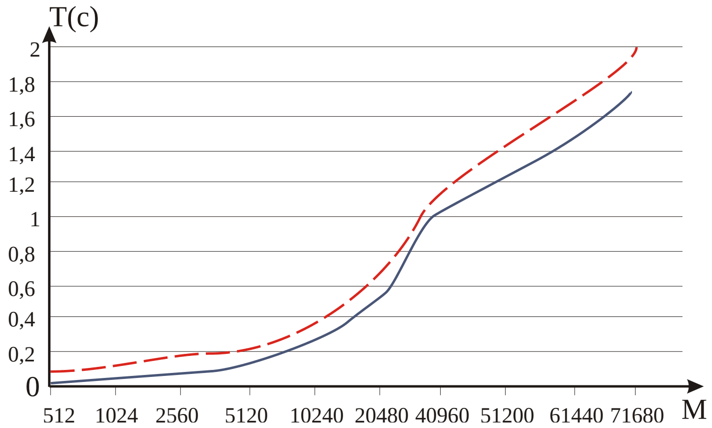

Рис. 3. Зависимость времени выполнения вычислений по алгоритму на GPU от размеров сеточной области (сплошная линия – на модели, пунктирная – экспериментально)

Заключение

Таким образом, в работе была предложена методика, которая позволяет выполнить отображение математической модели алгоритма циклической прогонки на вычислительную систему и предсказать эффективность вычислений по алгоритму для выбранной архитектуры ВС (СКО= 0,19). В отличие от [6 -8,16] предложенная модель алгоритма позволяет рассматривать двумерные сеточные области, а модель вычислительной системы в отличие от [9,10,16] содержит GPU. Представляется целесообразным проведение исследований на вычислительной системе, содержащей несколько вычислительных узлов, каждый из которых включает графическое вычислительное устройство.

Литература

1. Hockney R.W., Jesshope C. R. Parallel Computers: Architecture, Programming and Algorithms. Bristol: Adam Hilger,1981, 392 p.
2. Головашкин, Д. Л. Параллельные алгоритмы решения сеточных уравнений трёхдиагонального вида, основанного на методе встречных прогонок // Математическое моделирование. – 2005. – Т.17, № 11 – С. 118–128.

3. Самарский, А. А. Методы решения сеточных уравнений / А .А. Самарский, Е. С. Николаев. – М.:Наука,1978. – 561с.
4. Воеводин, В. В. Отображение проблем вычислительной математики на архитектуру вычислительных систем // Вычислительные методы и программирование. – 2000. – Т.1. – С. 37–44.
5. Марчук, Г. И. Проблемы вычислительной техники и фундаментальные исследования / Г. И.Марчук, В. Е.Котов // Автоматика и вычислительная техника.- 1979. – Т.2. – С. 3–14.
6. Воеводин, В. В. Математические модели и методы в параллельных процессах. – М.: Наука. Гл. ред. физ.-мат. Лит, 1986. – 296 с.
7. Воеводин, В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. – С.Пб.: БХВ-Петербург, 2002. – 608 с.
8. Гергель, В. П. Теория и практика параллельных вычислений: учебное пособие /В. П. Гергель. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. – 423 с.
9. Гергель, В. П. Теоретические основы экспериментального исследования алгоритмов планирования задач для вычислительного кластера с помощью симулятора / В. П. Гергель, П. Н. Полежаев // Вестник ОГУ. – 2010. – Т.9 (115). – С.115–121.
10. Хорошевский, В. Г. Архитектура вычислительных систем: учебное пособие / В. Г. Хорошевский – 2-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. – 520 с.
11. Головашкин, Д. Л. Решение сеточных уравнений неявных разностных схем с циклическими краевыми условиями на двумерных сеточных областях с использованием нескольких графических вычислительных устройств / Д. Л. Головашкин, Л.В.Логанова // Компьютерная оптика. – 2012. – Т.36, №4 – С.534–540.
12. NVIDIA CUDA computer unified device architecture, programming guide. – NVIDIA Corporation. – 2009. – Version 2.0
13. Антонов, А. С. Параллельное программирование с использованием технологии OpenMP: учебное пособие. – М.: Изд-во МГУ, 2009. – 77 с.
14. Боресков, А. В. Основы работы с технологией CUDA / А. В. Боресков, А. А. Харламов. – М.: ДМК Пресс, 2010. – 232 с.
15. NVIDIA CUDA Reference Manual. – NVIDIA Corporation. – 2010. – Version 3.0.
16. Логанова, Л. В. Выбор математической модели параллельного алгоритма с учётом особенностей архитектуры суперкомпьютера «Сергей Королёв» // Перспективные информационные технологии (ПИТ-2014): Труды Международной научно-технической конференции. – Самара, 2014. – С.335–339.