УДК 519.857

ПАРАЛЕЛЛИЗАЦИЯ УМНОЖЕНИЯ МАТРИЦ

А.В. Малышев
Технический директор ООО «Дартел»

Аннотация. В статье описывается поставленный эксперимент по определению целесообразности пересмотра классических алгоритмов умножения матриц с целью более эффективного использования многопроцессорных систем.
Ключевые понятия: линейная алгебра, матрицы, параллельные вычисления

Машинный анализ и расчет математических моделей, отражающих устройства электронной техники, в настоящее время часто осуществляется при помощи формирования и решения системы дифференциальных уравнений. При этом решение уравнений, как правило, производится с использованием численных методов, для которых значительная часть алгоритмов была разработана несколько десятилетий назад. Годы практического использования позволили отшлифовать алгоритмы и сделать их практически оптимальными для использования. Но научно-технический прогресс, благодаря которому многоядерные вычислительные системы становятся доступными для использования в качестве настольных ЭВМ, заставляет пересмотреть алгоритмы и организовать вычисления таким образом, чтобы максимально использовать параллелизацию для достижения максимальной производительности.

В данной статье рассматривается конкретная задача, возникающая при математическом моделировании устройств преобразовательной техники. Как известно, в соответствии с методом переменных состояния, расчетная система уравнений в канонической форме имеет вид [1]:

$$\begin{aligned} \dot{x} &= A \cdot x + B \cdot v \\ \dot{y} &= C \cdot x + D \cdot v \end{aligned} \qquad (1),$$

где x – вектор переменных состояния,

y – вектор источников тока и напряжения.

Фактически, машинный анализ электронной схемы сводится к построению и численному решению уравнения (1). Многие методы численного решения дифференциальных уравнений основаны на семействе уравнений Рунге [2], аппроксимирующих искомую функцию кусочно-линейной функцией, наклон которой к оси абсцисс определяется при помощи вычисления правых частей уравнений в различных точках в окрестности уже известной. На основании вычислений определяется коэффициент наклона прямой, наиболее близко аппроксимирующей искомую функцию и приближенное значение искомой функции на следующем шаге вычислений.

Практические исследования, проведенные автором при разработке системы моделирования устройств преобразовательной техники, показывают что основные затраты вычислительных мощностей при расчете процессов, происходящих в устройствах преобразовательной техники, приходятся на операции матричного умножения. Ранг матриц, участвующих в уравнении (1) для промышленных моделей может измеряться десятками и даже сотнями. Таким образом, перемножение матриц такого ранга поглощает до 80% времени исполнения программы.

Недостатком классических алгоритмов работы с матрицами Штрассена, Винограда, Копперсмита-Винограда [3], как, впрочем, и многих других классических алгоритмов, является их ориентированность на традиционную неймановскую [4] архитектуру компьютера. В то же время, архитектура современных процессоров, серийно выпускающихся и используемых для построения бытовых и научных ПК давно отошла от традиционной и в настоящий момент ее можно отнести к гарвардской архитектуре [4]. Таким образом, представляется возможным, и подтверждается экспериментально улучшение производительности классических алгоритмов, в частности, алгоритмов умножения матриц при реорганизации их для использования параллельных вычислений.

Необходимо отметить, что идея распараллеливания умножения матриц достаточно подробно изучена [5,6]. Тем не менее, простейший эксперимент показывает, что практически все доступные автору программные пакеты для моделирования электронных устройств реализуют последовательный алгоритм умножения матриц, реализующийся тремя вложенными циклами. Чтобы убедиться в этом, достаточно запустить достаточно сложную задачу по умножению и отследить загрузку процессоров многопроцессорной системы в ходе работы. Как правило, характер загрузки процессоров имеет следующий вид:

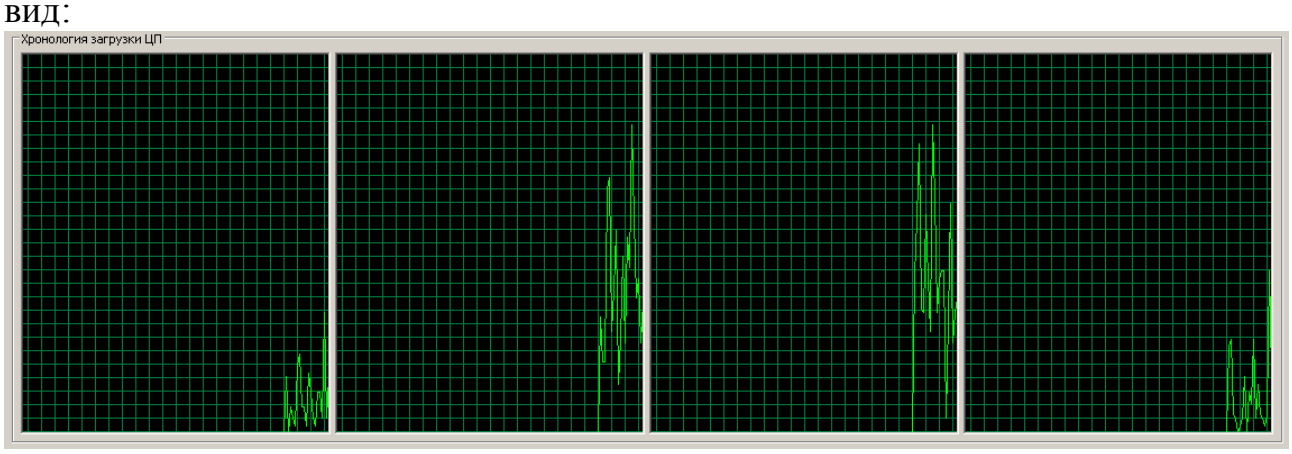

Рис. 1. Загрузка процессора при умножении матриц в Matlab 6.5

Здесь видно, что в каждый момент времени задача занимает только один процессор. Благодаря переключению контекста, необходимому для функционирования многопользовательской операционной системы MS Windows, задача может переходить с процессора на процессор. Тем не менее,

общая загрузка системы при выполнении умножения не превышает 25%, что позволяет сделать вывод о неэффективности использования имеющихся ресурсов. Данный рисунок является фрагментом снимка экрана, точнее, менеджера задач в ходе работы классического последовательного алгоритма умножения матриц.

Как правило, параллелизация матричного умножения организуется на основе разбиения задачи умножения на ряд подзадач по ленточному алгоритму, алгоритму Кэннона или Фокса [7]. В данной работе, экспериментальные измерения проводились на основе реализации ленточного алгоритма.

Алгоритм представляет собой итерационную процедуру, количество итераций которой совпадает с числом подзадач. На каждой итерации алгоритма каждая подзадача содержит по одной строке матрицы A и одному столбцу матрицы B. При выполнении итерации проводится скалярное умножение содержащихся в подзадачах строк и столбцов, что приводит к получению соответствующих элементов результирующей матрицы C. По завершении вычислений в конце каждой итерации столбцы матрицы B должны быть переданы между подзадачами с тем, чтобы в каждой подзадаче оказались новые столбцы матрицы B и могли быть вычислены новые элементы матрицы C. При этом данная передача столбцов между подзадачами должна быть организована таким образом, чтобы после завершения итераций алгоритма в каждой подзадаче последовательно оказались все столбцы матрицы B.

С использованием разработанной автором реализации ленточного алгоритма были проведены эксперименты в целях изучения целесообразности модификации разрабатываемого программного комплекса для моделирования устройств преобразовательной техники. В каждом эксперименте производилось умножение матриц размером A[SIZE,2*SIZE] и B[2*SIZE,SIZE]. Для повышения времени работы алгоритма и получения более наглядных результатов, умножение производилось TIMES раз. Для экспериментов была использована рабочая станция на основе Intel Core Quad Q6600 2,4 ГГц, содержащая 4 ядра под управлением ОС Windows XP.

Результаты эксперимента сведены в нижеследующую таблицу:

SIZE	TIMES	Количество элементарных действий	Количество создаваемых потоков	Ленточный алгоритм, µs на элем. операцию	Последовательный алгоритм, µs на элем. операцию	Прирост быстродействия %
1000	10	80000000000	40	0,625352	2,122715	239,443138
500	50	50000000000	200	0,384833	1,478816	284,274354
100	10000	80000000000	40000	0,155283	0,4754	206,151833
50	50000	50000000000	200000	0,423823	0,487002	14,9070135
30	80000	17280000000	320000	1,801389	0,519154	-71,1803439
10	80000	640000000	320000	46,33724	0,731175	-98,4220576

По данным эксперимента можно составить график, показанный на рис.2.

На данном графике особый интерес представляют две точки. Во-первых, резкий излом графика при размере матриц порядка 20 000 элементов. Объясняется это тем, что матрицы меньшего ранга полностью умещаются в кэше современных процессоров, что приводит к существенному уменьшению затрат времени на элементарную операцию за счет ускорения доступа к

элементам матрицы. Как только матрицы перестают умещаться в кэш, эффективность работы с ними резко падает. Это происходит как раз на уровне 20 000 элементов.

Не менее интересна точка пересечения графиков. Это момент, когда накладные расходы на организацию параллельных вычислений превосходят выигрыш от более эффективного использования процессоров. В данной реализации параллельность реализовывалась посредством запуска 4 различных потоков для вычисления. К сожалению, задача запуска отдельного потока является достаточно ресурсоемкой сама по себе, и при размерах матриц порядка 50x100, т.е. 5 000 элементов оказывается, что быстрее использовать последовательный метод. Существуют различные методики, которые позволяют снизить этот порог, но избавиться от него не удается.

Подводя итог проведенным исследованиям, можно сделать следующие

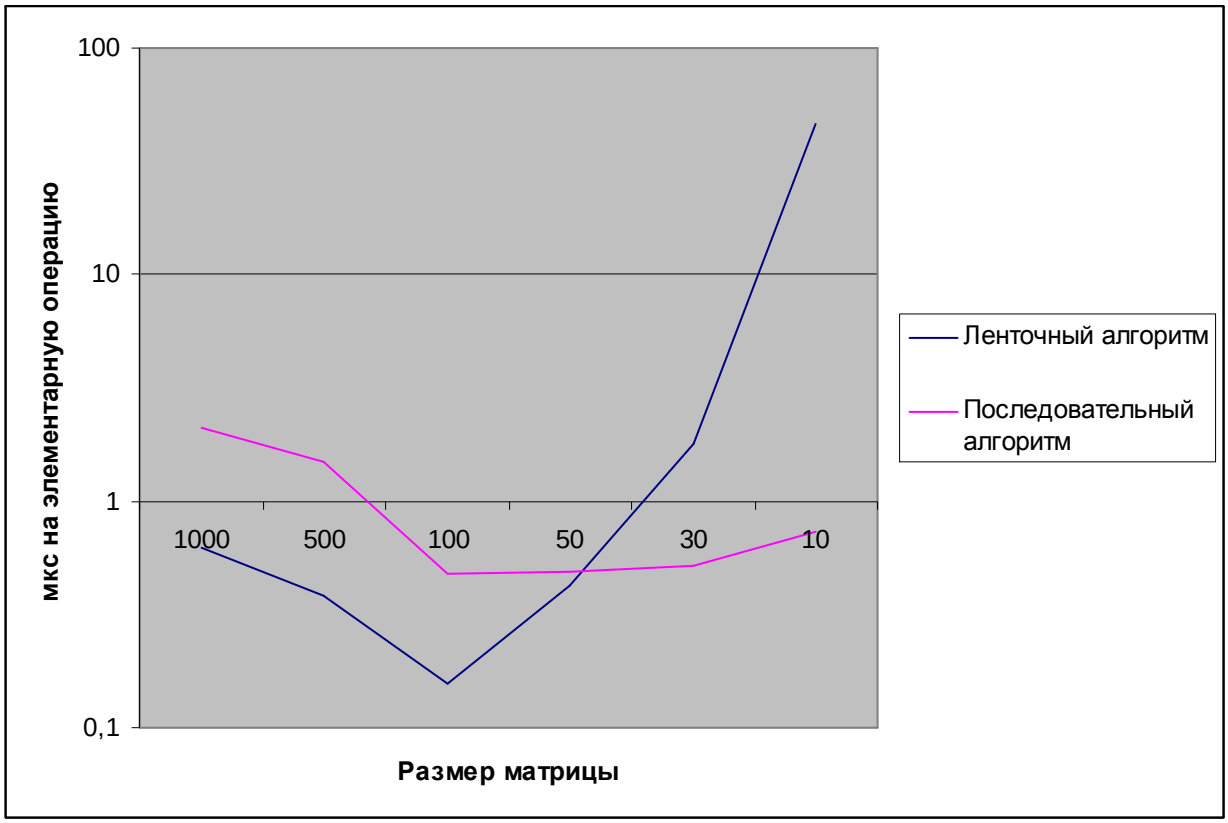

Рис.2. Эффективность умножения матриц

выводы. Во-первых, при разработке приложений, ориентированных на высокую производительность, всегда требуется учитывать архитектуру современных ЭВМ, и осуществлять корректировку классических алгоритмов, если это оправдано. Во-вторых, благодаря специфике современных процессоров, очень многообещающим представляется применение алгоритма Кэннона, поскольку при использовании этого алгоритма матрица разбивается на блоки, каждый из которых может полностью содержаться в кэш-памяти процессора. Тем не менее, подбор оптимальной величины разбиения представляет достаточно сложную задачу, общее решение которой до сих пор не существует. В-третьих,

для матриц достаточно малого размера (менее тысячи элементов) последовательные алгоритмы еще долгое время будут оптимальными из-за крайне высоких затрат на организацию параллельных вычислений и коммуникаций между вычислителями.

Список использованной литературы

1. Чуа, Л.О. Машинный анализ электронных схем. Алгоритмы и вычислительные методы / Л.О. Чуа, Пен-Мин Лин – М.: Наука,1980
2. Бахвалов, Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков – М.: Наука, 1987
3. Самарский, А.А. Численные методы / А.А. Самарский, А.В. Гулин – М.:Наука, 1989
4. Таненбаум, Э. Архитектура компьютера / Э. Таненбаум – СПб.: Питер, 2002
5. Воеводин, В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин – СПб.: БХВ-Петербург, 2002
6. Гергель, В.П. Основы параллельных вычислений для многопроцессорных вычислительных систем / В.П. Гергель, Р.Г. Стронгин – Н. Новгород: Изд-во ННГУ, 2001
7. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р.М. Ривест: – МЦНТО, 1999

Сведения об авторе

Малышев Андрей Владимирович, технический директор ООО «Дартел»,
тел. 30-79-00, email amalyshev@dartel.ru