

Интернет-журнал «Наукоедение» ISSN 2223-5167 <http://naukovedenie.ru/>

Том 8, №2 (2016) <http://naukovedenie.ru/index.php?p=vol8-2>

URL статьи: <http://naukovedenie.ru/PDF/62TVN216.pdf>

DOI: 10.15862/62TVN216 (<http://dx.doi.org/10.15862/62TVN216>)

Статья опубликована 21.04.2016.

Ссылка для цитирования этой статьи:

Уральский Н.Б., Сизов В.А., Капустин Н.К. Применение модифицированного генетического алгоритма для распараллеливания задачи умножения матриц большой размерности в гетерогенных системах обработки данных // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 8, №2 (2016) <http://naukovedenie.ru/PDF/62TVN216.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/62TVN216

УДК 51

Уральский Николай Борисович

ФГБОУ ВПО «Российский государственный социальный университет», Россия, Москва
Аспирант
E-mail: nik-ural@yandex.ru

Сизов Валерий Александрович

ФГБОУ ВПО «Российский государственный социальный университет», Россия, Москва
Профессор
Доктор технических наук
E-mail: sizovva@gmail.com

Капустин Николай Клементьевич

ФГБОУ ВПО «Московский авиационный институт (национальный исследовательский университет)», Россия, Москва
Доцент
E-mail: vopros150@yandex.ru

Применение модифицированного генетического алгоритма для распараллеливания задачи умножения матриц большой размерности в гетерогенных системах обработки данных

Аннотация. Рассматривается решение классической задачи умножения матриц большой размерности в гетерогенных распределённых системах обработки данных с помощью модифицированного генетического алгоритма. Приводятся постановка и результаты компьютерного эксперимента по сравнительной оценке эффективности этого модифицированного генетического алгоритма с классическим и смешанным генетическими алгоритмами, а также обыкновенным блочным методом в системах с симметричной и гетерогенной конфигурацией.

Матричные операции составляют значительную часть вычислений многих задач в различных областях науки и техники. При обработке матриц большой размерности в многопроцессорных системах для ускорения процесса вычислений требуется распараллеливание исходной задачи. Среди различных матричных операций умножение матриц одна из операций, часто применяемая при исследованиях в области распараллеливания программ в распределённых системах обработки данных. В отличие от существующих блочных методов параллельного умножения, методы, включающие поиск оптимального расписания, позволяют сократить время вычислений в гетерогенных системах.

В процессе эксперимента проведен сравнительный анализ методов, основанных на построении расписаний и блочного метода. Проведенный эксперимент доказывает, что при использовании методов с построением расписаний в гетерогенных системах время вычислительного процесса сокращается.

Ключевые слова: гетерогенные структуры; распараллеливание; генетический алгоритм; умножение матриц; распределённые системы обработки данных

Введение

Матричные вычисления фигурируют в процессах решения различных научных вычислительных задач в таких областях, как вычислительная математика, физика, экономика и др. [3]. Операции над матрицами широко используются при математическом моделировании разнообразных процессов, явлений и систем.

Являясь вычислительно-трудоемкими, матричные вычисления представляют собой классическую область применения параллельных вычислений. С одной стороны, использование высокопроизводительных многопроцессорных систем позволяет существенно повысить сложность решаемых задач. С другой стороны, в силу своей достаточно простой формулировки матричные операции предоставляют, подходящую основу для демонстрации и анализа приемов, и методов параллельного программирования [7].

Специальные подходы к исследованию ряда задач, в частности, например, задач электродинамики на основе решения интегральных уравнений, опираются на работу с большими матрицами, генерация которых является основным вычислительно емким местом. В таких случаях очень часто помогает то, что вычисление отдельных элементов матриц можно производить параллельно [4].

В качестве наглядного примера задачи, в решения которой применяются операции над матрицами, можно привести обычную в физике или в машиностроении задачу затухания, описанную Э. Таненбаумом в работе [11]. Обычно это матрица, содержащая некоторые исходные значения. Эти значения могут представлять собой температуру в разных точках листа металла. Замысел может состоять в определении скорости распространения разогрева от пламени, воздействующего на один из его углов.

Начиная с текущих значений, к матрице для получения ее следующей версии применяется преобразование, соответствующее законам термодинамики, чтобы посмотреть все температурные показания через время ΔT . Затем процесс повторяется снова и снова, представляя температурные значения в виде функции, зависящей от времени нагревания листа. Со временем алгоритм производит серию матриц, каждая из которых соответствует заданной отметке времени.

В том случае если матрица имеет слишком большие размеры (миллион на миллион) для ускорения ее вычисления требуется распараллеливание исходной задачи для решения её на многопроцессорных системах, в которых разные узлы работают над разными частями матрицы, вычисляя новые элементы матрицы на основе прежних [11].

Именно по такой схеме решалась задача дифракции электромагнитного поля на диэлектрическом анизотропном теле произвольной формы. Задача была решена за 26 дней на четырех разных кластерах суперкомпьютерного комплекса НИВЦ МГУ. Использовались только те интервалы времени, когда процессоры кластеров не были заняты пользователями. Весь расчет проведен на фоне штатной работы суперкомпьютерного комплекса только за счет использования периодов незанятости процессоров. Эффективность работы созданной

вычислительной среды составила 98%. Решение этой же задачи в обычном режиме заняло бы четыре года работы одного компьютера [4].

1. Алгоритмы распараллеливания операции умножения матриц в распределённых системах

Умножение матриц одна из операций часто применяемая при исследованиях в области распараллеливания программ для распределённых систем обработки данных. Различные алгоритмы распараллеливания данной операции рассматриваются во многих источниках, например в работах [8, 9] описаны методы, основанные на разделении матриц на строки и столбцы, после такого разбиения, выполняемого на центральном узле, полосы данных рассылаются между остальными узлами для выполнения вычислений.

В работах [7, 8, 9] описаны, так называемые блочные методы параллельного умножения матриц, суть которых сводится к тому, что исходные и результирующая матрицы представляются в виде наборов прямоугольных блоков размера $m \times m$. Тогда операцию матричного умножения матриц A и B в блочном виде можно представить следующим образом:

$$\begin{pmatrix} A_{11} & A_{12} & A_{1k} \\ & \dots & \\ A_{k1} & A_{k2} & A_{kk} \end{pmatrix} \otimes \begin{pmatrix} B_{11} & B_{12} & B_{1k} \\ & \dots & \\ B_{k1} & B_{k2} & B_{kk} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{1k} \\ & \dots & \\ C_{k1} & C_{k2} & C_{kk} \end{pmatrix}$$

где каждый блок C_{ij} матрицы C определяется в соответствии с выражением

$$C_{ij} = \sum_{l=1}^k A_{il} B_{lj}$$

Данный метод подробно рассмотрен в работе [7]. При таком способе организации вычислений пересылка данных оказывается распределенной по времени и это может позволить совместить процессы передачи и обработки данных. Метод является примером широко распространенного способа организации параллельных вычислений, состоящего в распределении между процессорами обрабатываемых данных с учетом близости их расположения в содержательных постановках решаемых задач. Подобная идея, часто называемая в литературе *геометрическим принципом распараллеливания*, широко используется при разработке параллельных методов решения сложных задач, поскольку во многих случаях может приводить к значительному снижению потоков пересылаемых данных за счет локализации на процессорах существенно информационно-зависимых частей алгоритмов (в частности, подобный эффект может быть достигнут при численном решении дифференциальных уравнений в частных производных) [7].

Все вышеописанные методы оптимально работают в т.н. симметричных системах, в которых узлы имеют одинаковую производительность. В задачи центрального узла входит только распределение частей матрицы по узлам.

Но при решении крупномасштабных задач встает вопрос - как организовать вычислительную сеть в рамках одного предприятия, включающую однотипные узлы и коммуникации с одинаковой пропускной способностью. В данном случае более рациональным выглядит применение т.н. гетерогенных систем [13].

Один из типов таких распределённых систем - вычислительные среды, рассмотрен в работе [4]. Данные системы обладают параллельной архитектурой и распределенной памятью, что на первый взгляд делает их похожими на вычислительные кластеры. Но это лишь внешнее сходство. Истинная сущность распределенных вычислительных сред совершенно иная и определяется набором свойств, которого не было ни у одной из ранее существовавших компьютерных систем.

Эти вычислительные системы могут быть построены на базе частных корпоративных сетей, и узлы таких систем представляют машины конечных пользователей.

Суммарная производительность компьютеров сети весьма высока и, следовательно, есть необходимый потенциал для создания инструмента решения больших задач. Среда формируется на основе уже действующих компьютеров сети, новых затрат практически не требуется, что и делает такой подход экономически привлекательным. Более того, развитие и модернизация вычислительных сред происходит автоматически по мере обновления компьютерного парка сети [4].

2. Распараллеливание задачи умножения матриц с применением методов на основе эволюционного подхода

Как видно из предыдущей главы блочные методы имеют хорошую совместимость с SMP системами, в которых все узлы обладают одинаковой производительностью и нет смысла затрачивать время на планирование.

В гетерогенных системах и распределённых вычислительных средах такой подход может оказаться неэффективным из-за больших простоев узлов. В таких системах целесообразно будет использовать алгоритмы распараллеливания, в основе которых присутствует диспетчеризация, и которые позволяют подогнать структуру ПО под конфигурацию системы, либо некоторое её состояние [10, 12].

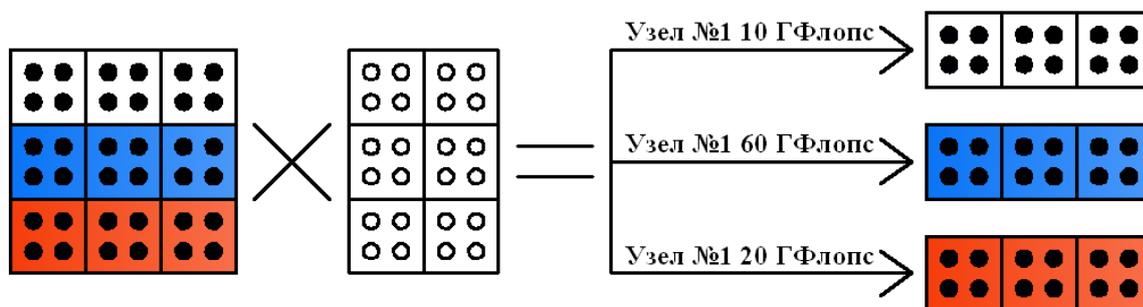


Рисунок 1. Распределение блоков матрицы в системе без учета производительности узлов

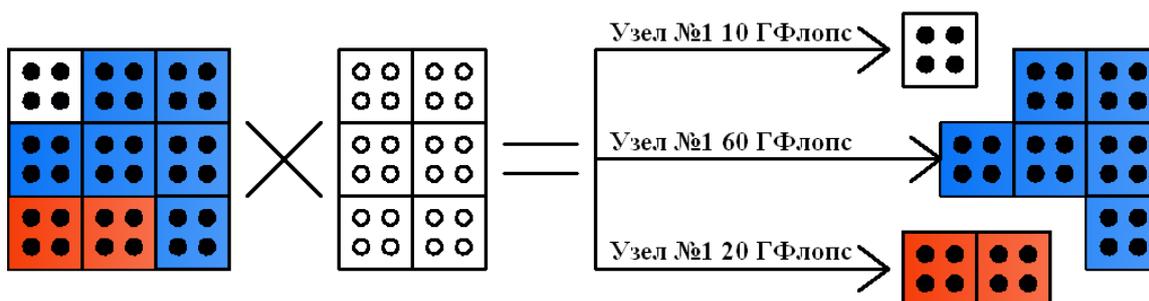


Рисунок 2. Распределение блоков матрицы в системе с учетом производительности узлов

На рисунках 1 и 2 показаны возможные варианты распределения матричных блоков в системах с учетом и без учета производительности узлов.

Существуют различные алгоритмы нахождения оптимальных расписаний. В данной работе проводится исследование методов, основанных на эволюционном подходе, а именно генетических алгоритмов (ГА).

Применение классического генетического алгоритма в части синтеза оптимальных расписаний исполнения параллельных задач обсуждается во многих источниках и его рассмотрение выходит за рамки данной статьи [1, 2, 6].

В работе [12] предложен модифицированный ГА, в котором фитнес-функция назначает неточные оценки расписаниям. Суть метода заключается в том, что при вычислении оценок фитнес-оператор ГА обрабатывает только часть расписания, производя циклическое сравнение уже вычисленного времени выполнения расписания с лучшей оценкой на данном этапе работы фитнес-функции. Недостаток разработанного метода заключается в большой склонности ГА, использующего модифицированную фитнес-функцию, к преждевременной сходимости, для снижения эффекта ранней конвергенции в статье предлагается попеременное включение классического и модифицированного алгоритмов (смешанный ГА).

При использовании алгоритмов распараллеливания на основе построения расписаний с учётом производительности узлов исходная задача умножения матриц разбивается на атомарные процедуры, которые получают на вход блоки данных.

Предполагается, что во время выполнения умножения матриц в системе, один узел параллельно выполняет поиск расписания для следующей операции умножения матриц и далее выполняет генерацию кода и рассылку частей программы по узлам [4, 9].

Разложив операцию умножения матриц на отдельные процедуры можно заметить, что данная задача оптимально согласуется с концепцией операционных модулей предлагаемой в работе [10]. Вся задача разбивается на атомарные процедуры умножения и сложения элементов матрицы, т.е. происходит построение графа канонической формы информационно-зависимых задач (ИЗЗ). На следующем шаге стартует генетический алгоритм, на выходе которого генерируется расписание, т.о. атомарные процедуры объединяются в операционные модули и распределяются по сети.

3. Планирование эксперимента по оценке эффективности различных методов параллельного умножения матриц в распределённых системах и его результаты

3.1. Цель эксперимента

Цель эксперимента, описанного в данной работе, заключалась в сравнении эффективности классического ГА, модифицированного ГА, смешанного ГА и обыкновенного блочного метода в системах с симметричной и гетерогенной конфигурацией на примере параллельного выполнения операции умножения матриц.

3.2. Инструментальные средства

В процессе эксперимента применялся персональный компьютер (ПК) (таблица 1) и программный комплекс, моделирующий работу вычислительной сети.

Таблица 1

Характеристики ПК

Процессор	ОЗУ
Intel Core i7-4770K Haswell (3500MHz, LGA1150, L3 8192Kb)	16 Гб

Разработанный комплекс включает программу, которая выполняется под управлением операционной системы реального времени OCPB QNX 6.5.0. Данная OCPB применялась в целях обеспечения максимальной точности измерения времени выполнения алгоритмов. QNX 6.5.0, основанная на микроядерной архитектуре, характеризуется повышенным быстродействием. Особенности данной архитектуры обеспечивают реакцию системы в течение строго определённого периода времени.

Программный комплекс позволяет задавать различные конфигурации сети и производить сравнение эффективности ГА и блочных методов.

3.3. Результаты эксперимента

В процессе проведения эксперимента в программу вводились исходные данные с различными конфигурациями сети и комплекса ИЗЗ, представляющего операцию умножения двух квадратных матриц размерностью 10000x10000. За атомарную операцию принималась операция умножения строки и столбца размерностью 10 элементов.

В первой части эксперимента, результаты которой представлены в таблице 2 проводилось сравнение вышеописанных методов на симметричной системе.

Таблица 2

Результаты эксперимента для симметричной системы

Название метода	Время поиска расписания	Среднее время выполнения		
		Размерность системы		
		100 узлов	1000 узлов	2500 узлов
Классический ГА	100 мкс	1,2 с	70 мкс	59 мкс
Модифицированный ГА	100 мкс	1,2 с	68 мкс	61 мкс
Смешанный ГА	100 мкс	0,94 с	64 мкс	58 мкс
Блочный метод	0	0,78 с	59 мкс	46 мкс
Классический ГА	50 мкс	1,6 с	90 мкс	68 мкс
Модифицированный ГА	50 мкс	1,2 с	69 мкс	63 мкс
Смешанный ГА	50 мкс	1,1 с	67 мкс	61 мкс
Блочный метод	0	780 мс	59 мкс	46 мкс

Из таблицы 2 видно, что блочный метод в симметричной системе явно преобладает над методами, синтезирующими расписания.

Во второй части эксперимента производилось сравнение в гетерогенной системе.

Таблица 3

Результаты эксперимента для гетерогенной системы

Название метода	Время поиска расписания	Среднее время выполнения		
		Размерность системы		
		100 узлов	1000 узлов	2500 узлов
Классический ГА	100 мкс	1,1 с	92 мкс	83 мкс
Модифицированный ГА	100 мкс	0,94 с	80 мкс	73 мкс
Смешанный ГА	100 мкс	0,82 с	67 мкс	62 мкс
Блочный метод	0	2,4 с	354 мкс	287 мкс

Название метода	Время поиска расписания	Среднее время выполнения		
		Размерность системы		
		100 узлов	1000 узлов	2500 узлов
Классический ГА	50 мкс	1,2 с	96 мкс	85 мкс
Модифицированный ГА	50 мкс	0,98 с	87 мкс	77 мкс
Смешанный ГА	50 мкс	0,87 с	69 мкс	63 мкс
Блочный метод	0	2,4 с	354 мкс	287 мкс

Из второй части эксперимента (таблица 3) видно, что несмотря на временные затраты, связанные с поиском оптимального расписания, эффективность методов, применяющих ГА, выше чем у блочного, что связано с большой неравномерностью вычислительных ресурсов гетерогенной системы.

Заключение

Проведенный эксперимент показал, что при использовании методов, основанных на генерации расписаний, операция умножения матриц в гетерогенной сети производится эффективней, чем с применением блочных методов. К недостаткам ГА можно отнести временные затраты на поиск расписания.

Также следует обратить внимание на проблему выбора размера атомарных процедур. При минимальном размере процедур временные затраты на синтез расписания значительно возрастают, из чего можно сделать вывод, что применение данных алгоритмов возможно в программах, в которых операция умножения выполняется циклически. В таком случае возможно в начале вычислений параллельно запускать поиск расписания для последующей операции. Описанные методы могут также находить применение в параллельных программах сортировки данных.

В системах, имеющих симметричную конфигурацию, более эффективным будет применение блочных методов. К достоинствам, которых следует также отнести относительную простоту реализации.

ЛИТЕРАТУРА

1. Аверченков В.И., Казаков П.В. Эволюционное моделирование и его применение / 2-е изд., – М.: Флинта, 2011.
2. Бабин Д.В., Вороной С.М., «Генетический алгоритм построения расписаний для многопроцессорных вычислительных систем», Донецкий государственный институт искусственного интеллекта, Украина «Искусственный интеллект», 2005.
3. Беллман Р. Введение в теорию матриц / 2-е изд., – М.: Флинта, 2015.
4. Воеводин В. Решение больших задач в распределенных вычислительных средах/ Автомат. и телемех., 2007
5. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления / СПб.: БВХ-Петербург, 2002.
6. Jaspal Singh, Harsharanpal Singh, Efficient Tasks scheduling for heterogeneous multiprocessor using Genetic algorithm with Node duplication”, Indian Journal of Computer Science and Engineering Vol. 2 No. 3 Jun-Jul 2011.
7. Интернет-Университет Суперкомпьютерных Технологий: Теория и практика параллельных вычислений
<http://www.intuit.ru/studies/courses/1156/190/lecture/4952>.
8. Корнеев В.В. Параллельные вычислительные системы. - М.: Нолидж, 1999.
9. Корнеев В.В. Параллельное программирование в MPI. Москва-Ижевск: Институт компьютерных исследований, 2003.
10. Сизов В.А. Проектирование программного и информационного обеспечения комплекса связанных задач в сети ЭВМ / Автоматизированные системы управления, 1995.
11. Таненбаум Э. Современные операционные системы. 4-е изд. - СПб.: Питер, 2015.
12. Уральский Н.Б., Сизов В.А., Капустин Н. К. Оптимизация вычислительного процесса фитнес-функции генетического алгоритма в распределённых системах обработки данных / Интернет-журнал «НАУКОВЕДЕНИЕ» Том 7, №6, 2015 <http://naukovedenie.ru>.
13. Хорошевский В.Г. Распределённые вычислительные системы с программируемой структурой / Вестник СибГУТИ №2, 2010.

Uralskii Nikolai Borisovich

Russian state social university, Russia, Moscow
E-mail: vopros150@yandex.ru

Sizov Valerii Aleksandrovich

Russian state social university, Russia, Moscow
E-mail: nik-ural@yandex.ru

Kapustin Nikolay Klementievich

Moscow aviation institute, Russia, Moscow
E-mail: vopros150@yandex.ru

Application of a modified genetic algorithm parallelizing the matrix multiplication task of high dimensionality in distributed data processing systems

Abstract. The solution of the classical problem of multiplication of matrices of large dimension in the heterogeneous data processing systems using a modified genetic algorithm. The statements and the results of the computer experiment for the comparative evaluation of the effectiveness of the modified genetic algorithm with the classic and mixed genetic algorithms, as well as ordinary block method in a symmetric configuration and heterogeneous systems.

Matrix operations are an important part of many computing problems in various fields of science and technology. When processing matrices of large dimension in multiprocessor systems to accelerate computations require parallelization of the original problem. Among various matrix operations matrix multiplication is one of the operations, often used in research in the field of parallel programs in distributed data processing systems. Unlike existing block methods parallel multiplication techniques, including the search for optimal schedules, reduce the time of computations in heterogeneous systems.

In the course of the experiment a comparative analysis of methods based on the construction schedules and block method. The experiment proves that when using methods with the schedules time of the computational process is reduced.

Keywords: heterogeneous structure; parallelization; genetic algorithm; matrix multiplication; distributed data processing system

REFERENCES

1. Averchenkov V.I., Kazakov P.V. Evolyutsionnoe modelirovanie i ego primeneniye / 2-e izd., – M.: Flinta, 2011.
2. Babin D.V., Voronoy S.M., «Geneticheskiy algoritm postroeniya raspisaniy dlya mnogoprotsessornykh vychislitel'nykh sistem», Donetskii gosudarstvennyy institut iskusstvennogo intellekta, Ukraina «Iskusstvennyy intellekt», 2005.
3. Bellman R. Vvedeniye v teoriyu matrits / 2-e izd., – M.: Flinta, 2015.
4. Voevodin V. Resheniye bol'shikh zadach v raspredelennykh vychislitel'nykh sredakh/ Avtomat. i telemekh., 2007
5. Voevodin V.V., Voevodin V.I. Parallelnyye vychisleniya / SPb.: BVKh-Peterburg, 2002.
6. Jaspal Singh, Harsharanpal Singh, Efficient Tasks scheduling for heterogeneous multiprocessor using Genetic algorithm with Node duplication”, Indian Journal of Computer Science and Engineering Vol. 2 No. 3 Jun-Jul 2011.
7. Internet-Universitet Superkomp'yuternykh Tekhnologiy: Teoriya i praktika parallelnykh vychisleniy <http://www.intuit.ru/studies/courses/1156/190/lecture/4952>.
8. Korneev V.V. Parallelnyye vychislitel'nyye sistemy. - M.: Nolidzh, 1999.
9. Korneev V.V. Parallelnoye programmirovaniye v MPI. Moskva-Izhevsk: Institut komp'yuternykh issledovaniy, 2003.
10. Sizov V.A. Proektirovaniye programmnoy i informatsionnoy obespecheniya kompleksa svyazannykh zadach v seti EVM / Avtomatizirovannyye sistemy upravleniya, 1995.
11. Tanenbaum E. Sovremennyye operatsionnyye sistemy. 4-e izd. - SPb.: Piter, 2015.
12. Ural'skiy N.B., Sizov V.A., Kapustin N. K. Optimizatsiya vychislitel'nogo protsessa fitness-funktsii geneticheskogo algoritma v raspredelennykh sistemakh obrabotki dannykh / Internet-zhurnal «NAUKOVEDENIE» Tom 7, №6, 2015 <http://naukovedenie.ru>.
13. Khoroshevskiy V.G. Raspredelennyye vychislitel'nyye sistemy s programmirovannoy strukturoy / Vestnik SibGUTI №2, 2010.