Парфенов Н.В.

Смоленский государственный университет, г. Смоленск, Россия

АНАЛИЗ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА УМНОЖЕНИЯ МНОГОМЕРНЫХ МАТРИЦ

АННОТАЦИЯ

В работе дано описание вычислительного эксперимента, который проводился на различных программно-аппаратных комплексах для оценки алгоритма умножения матриц с различным числом измерений. Приведены результаты эксперимента и проведен их анализ.

КЛЮЧЕВЫЕ СЛОВА

Многомерные матрицы; параллельное программирование; программно-аппаратные комплексы.

Parfenov N.V.

Smolensk State University, Smolensk, Russia

THE ANALYSIS OF PARALLEL ALGORITHMS FOR MULTIDIMENSIONAL MATRICES MULTIPLICATION

ABSTRACT

The paper describes a computational experiment, which was conducted on various software and hardware complexes for the evaluation multidimensional matrix multiplication algorithm with different numbers of dimensions. The experimental results are presented and analyzed.

KEYWORDS

Multidimensional matrix; parallel programming; hardware and software complexes.

Операция умножения многомерных матриц определяется следующим образом. Пусть матрицы $A = \left\| a_{i_1 \dots i_p} \right\|$ и $B = \left\| b_{i_1 \dots i_q} \right\|$, p и q-мерные соответственно. Совокупности индексов этих матриц $i_1,\dots,i_p$ и $i_1,\dots,i_q$ разбиваются на четыре группы, содержащие соответственно κ, λ, μ и ν индексов (κ, λ, μ, $\nu \geq 0$). Причем $\kappa+\lambda+\mu=p$, а $\lambda+\mu+\nu=q$. Для полученных групп индексов используются обозначения: $l = (l_1,\dots,l_\kappa)$, $s = (s_1,\dots, s_\lambda)$, $c=(c_1,\dots c_\mu)$ и $m=(m_1,\dots m_\nu)$. Тогда матрицы A и B можно представить в виде в виде $A = \left\| a_{lsc} \right\|$ и $B=\left\| b_{scm} \right\|$. Индексы групп s и c в матрицах A и B полностью совпадают. Так же как в операции свертки, индексы разбиения c называются кэлиевыми. Индексы разбиения s называются скоттовыми, а индексы разбиения m, так же как и индексы разбиения l, – свободными. Матрица $C = \left\| c_{lsm} \right\|$, элементы которой вычисляются по формуле $c_{lsm} = \sum\limits_{(c)} a_{lsc} \times b_{scm}$, называется (λ, μ)-свернутым произведением матриц A и B и обозначается ${}^{\lambda,\mu}(A \times B)$ [1].

Для умножения использовался блочный параллельный алгоритм умножения многомерных матриц, полное описание которого приведено в [2, 3, 5].

Эксперимент проводился для трехмерных (A_{lsc}, B_{scm}) и четырехмерных $(A_{ls_1s_2c}, B_{s_1s_2cm})$ числовых матриц. Далее приводится описание работы алгоритма для трехмерных матриц при разбиении на 27 потоков.

В случае разбиении на 27 потоков процессорная матрица представляет собой трехмерную матрицу с тремя процессорными слоями и девятью потоками на каждом из слоев (таблица 1).

Таблица 1. Слои процессорной решетки

P_{000}	P_{100}	P_{200}	P_{010}	P_{110}	P_{210}	P_{020}	P_{120}	P_{220}
P_{001}	P_{101}	P_{201}	P_{011}	P_{111}	P_{211}	P_{021}	P_{121}	P_{221}
P_{002}	P_{102}	P_{202}	P_{012}	P_{112}	P_{212}	P_{022}	P_{122}	P_{222}

На первом этапе производится "обнуление" (заполнение значениями нейтрального элемента) блоков матрицы Clsm в соответствующих процессорах решетки и распределение блоков матриц Alsc и Bscm по процессорам решетки. Каждому слою процессорной решетки соответствуют свои сечения матриц Alsc, Bscm и Clsm. Индекс процессора совпадает с индексом вычисляемого им блока. Для вычислений процессоры s-слоя получают блоки сечения матрицы Alsc и Bscm, после чего блоки этих сечений подвергаются циклическим сдвигам между процессорами по следующим правилам:

- Блоки Aisc (i=0, …, q–1) циклически сдвигаются в порядке убывания номеров строк на i позиций.
- Блоки Bscj (j=0, …, q–1) циклически сдвигаются в порядке убывания номеров столбцов на j позиций.

Таблица 2. Начальное размещение блоков матриц A и B

P_{000}	C_{000} A_{000} B_{000}	P_{100}	C_{100} A_{101} B_{010}	P_{200}	C_{210} A_{202} B_{020}
P_{001}	C_{001} A_{001} B_{011}	P_{101}	C_{101} A_{102} B_{021}	P_{201}	C_{211} A_{200} B_{001}
P_{002}	C_{002} A_{002} B_{022}	P_{102}	C_{102} A_{100} B_{002}	P_{202}	C_{212} A_{201} B_{012}
P_{010}	C_{010} A_{010} B_{100}	P_{110}	C_{110} A_{111} B_{110}	P_{210}	C_{210} A_{212} B_{120}
P_{011}	C_{011} A_{011} B_{111}	P_{111}	C_{111} A_{112} B_{121}	P_{211}	C_{211} A_{210} B_{101}
P_{012}	C_{012} A_{012} B_{122}	P_{112}	C_{112} A_{110} B_{102}	P_{212}	C_{212} A_{211} B_{112}
P_{020}	C_{020} A_{020} B_{200}	P_{120}	C_{120} A_{121} B_{210}	P_{220}	C_{220} A_{222} B_{220}
P_{021}	C_{021} A_{021} B_{211}	P_{121}	C_{121} A_{122} B_{221}	P_{221}	C_{221} A_{220} B_{201}
P_{022}	C_{022} A_{022} B_{222}	P_{122}	C_{122} A_{120} B_{202}	P_{222}	C_{222} A_{221} B_{212}

Второй этап состоит из *q* итераций, на каждой из которых выполняются три действия:

- Каждый процессор Plsm выполняет умножение соответствующих ему блоков матриц Alsc и Bscm и складывает результат умножения с соответствующим ему блоком матрицы Clsm.
- Блоки Aisc (i=0, …, q–1) циклически сдвигаются в порядке убывания номеров строк на i позиций.
- Блоки Bscj (j=0, …, q–1) циклически сдвигаются в порядке убывания номеров столбцов на j позиций.

На последней итерации выполняется только пункт 1.

Результаты итераций показаны в таблицах.

Очевидно, что слои процессорной решетки, в которых производятся умножения сечений матриц A_{lsc} и B_{scm} должны иметь топологию – трехмерный тор. Выполнение этого требования необходимо потому, что пересылки блоков сечений матриц между процессорами слоя циклические как по "вертикали", так и по "горизонтали".

Таблица 3. Результат первой итерации

P_{000}	$C_{000}\ A_{001}\ B_{010}$	P_{100}	$C_{100}\ A_{102}\ B_{020}$	P_{200}	$C_{200}\ A_{200}\ B_{000}$
P_{001}	$C_{001}\ A_{002}\ B_{021}$	P_{101}	$C_{101}\ A_{100}\ B_{001}$	P_{201}	$C_{201}\ A_{201}\ B_{011}$
P_{002}	$C_{002}\ A_{000}\ B_{002}$	P_{102}	$C_{102}\ A_{101}\ B_{012}$	P_{202}	$C_{202}\ A_{202}\ B_{022}$
P_{010}	$C_{00}\ A_{011}\ B_{110}$	P_{110}	$C_{110}\ A_{112}\ B_{020}$	P_{210}	$C_{210}\ A_{210}\ B_{100}$
P_{011}	$C_{011}\ A_{012}\ B_{121}$	P_{111}	$C_{111}\ A_{110}\ B_{001}$	P_{211}	$C_{211}\ A_{211}\ B_{111}$
P_{012}	$C_{012}\ A_{010}\ B_{102}$	P_{112}	$C_{112}\ A_{111}\ B_{012}$	P_{212}	$C_{212}\ A_{212}\ B_{122}$
P_{020}	$C_{020}\ A_{021}\ B_{210}$	P_{120}	$C_{120}\ A_{122}\ B_{020}$	P_{220}	$C_{220}\ A_{220}\ B_{200}$
P_{021}	$C_{021}\ A_{022}\ B_{221}$	P_{121}	$C_{121}\ A_{120}\ B_{001}$	P_{221}	$C_{221}\ A_{221}\ B_{211}$
P_{022}	$C_{022}\ A_{020}\ B_{202}$	P_{122}	$C_{122}\ A_{121}\ B_{012}$	P_{222}	$C_{222}\ A_{222}\ B_{222}$

Таблица 4. Результат второй итерации

P_{000}	$C_{000}\ A_{002}\ B_{020}$	P_{100}	$C_{100}\ A_{100}\ B_{000}$	P_{200}	$C_{200}\ A_{201}\ B_{010}$
P_{001}	$C_{001}\ A_{000}\ B_{001}$	P_{101}	$C_{101}\ A_{101}\ B_{011}$	P_{201}	$C_{201}\ A_{202}\ B_{021}$
P_{002}	$C_{002}\ A_{001}\ B_{012}$	P_{102}	$C_{102}\ A_{102}\ B_{022}$	P_{202}	$C_{202}\ A_{200}\ B_{002}$
P_{010}	$C_{010}\ A_{012}\ B_{120}$	P_{110}	$C_{110}\ A_{110}\ B_{000}$	P_{210}	$C_{210}\ A_{211}\ B_{110}$
P_{011}	$C_{011}\ A_{010}\ B_{101}$	P_{111}	$C_{111}\ A_{111}\ B_{011}$	P_{211}	$C_{211}\ A_{212}\ B_{121}$
P_{012}	$C_{012}\ A_{011}\ B_{112}$	P_{112}	$C_{112}\ A_{112}\ B_{022}$	P_{212}	$C_{212}\ A_{210}\ B_{102}$
P_{020}	$C_{020}\ A_{022}\ B_{220}$	P_{120}	$C_{120}\ A_{120}\ B_{000}$	P_{220}	$C_{220}\ A_{221}\ B_{210}$
P_{021}	$C_{021}\ A_{020}\ B_{201}$	P_{121}	$C_{121}\ A_{121}\ B_{011}$	P_{221}	$C_{221}\ A_{222}\ B_{221}$
P_{022}	$C_{022}\ A_{020}\ B_{212}$	P_{122}	$C_{122}\ A_{122}\ B_{022}$	P_{222}	$C_{222}\ A_{220}\ B_{202}$

В случае четырехмерных матриц схема разбиения матриц на блоки будет аналогична трехмерному случаю, только размерность каждого блока будет увеличена на 1.

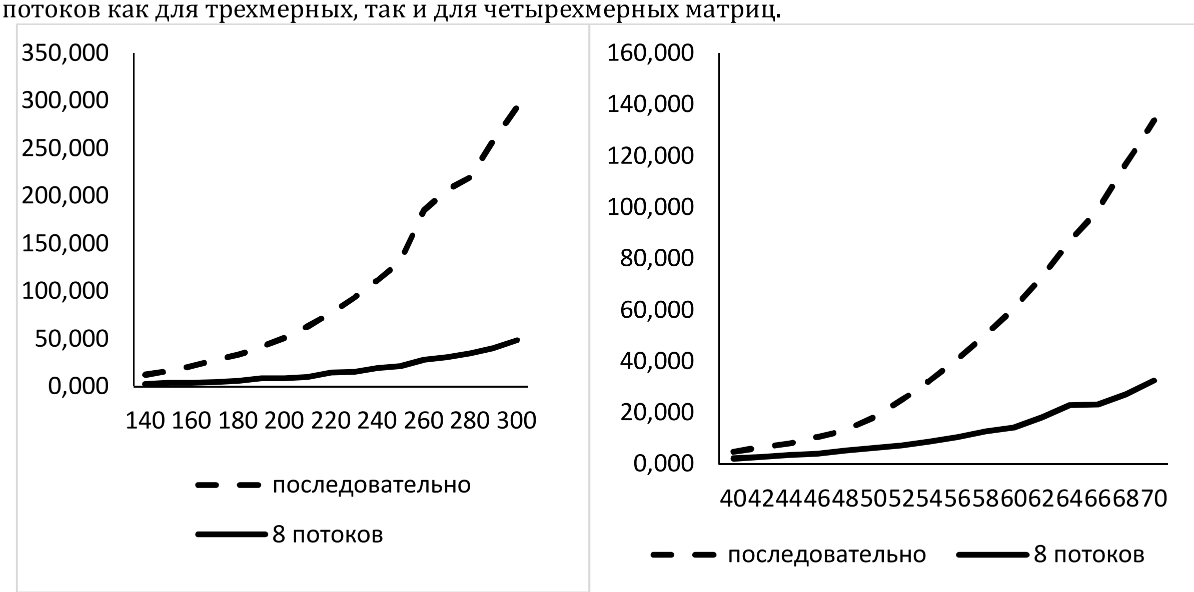
При проведении экспериментов выполнялись следующие условия:

- Эксперимент проводился для трехмерных и четырехмерных числовых матриц.
- Для трехмерных матриц выполнялось (1, 1)-свернутое произведение (Alsc, Bscm)
- Для четырехмерных выполнялось (2, 1)- свернутое произведение (Als1s2c, Bs1s2cm)
- Индексы трехмерных матриц принимали значения от 10 до 300 с шагом 10.
- Индексы четырехмерных матриц принимали значения 10 до 70 с шагом 2.
- На каждом шаге эксперимента выполнялись последовательное и параллельное (8 и 27 потоков, как 23 и 33) умножение многомерных матриц.

Таким образом, для трехмерных матриц выполнялось (1, 1)-свернутое произведение где один келлиев и один скотов индексы, а для четырехмерных – (2, 1)- свернутое произведение – два скоттовых и один келлиев. Все индексы имели одинаковую размерность.

Программно-аппаратные комплексы создавались как виртуальные машины Microsoft Azure на основе процессора Xeon E5-2673. Использовались конфигурации с восьмью и шестнадцатью ядрами.

Результаты эксперимента, приведенные на рисунке 1, показывают, что с увеличением размерности индексов возрастает эффективность параллельного алгоритма при разделении на 8 потоков как для трехмерных, так и для четырехмерных матриц.



А) трехмерные матрицы *Б) четырехмерные матрицы*

Рис. 1. Зависимость производительности алгоритма от размерности индексов

На рисунке 2 приведены результаты анализа производительности алгоритма параллельного умножения многомерных матриц в зависимости от соотношения числа потоков и числа вычислителей (ядер). Как можно заметить графики различаются. Разделение на 8 потоков более эффективно на восьми ядерной машине, а разделение на двадцать семь потоков более эффективно на шестнадцати ядерной машине.

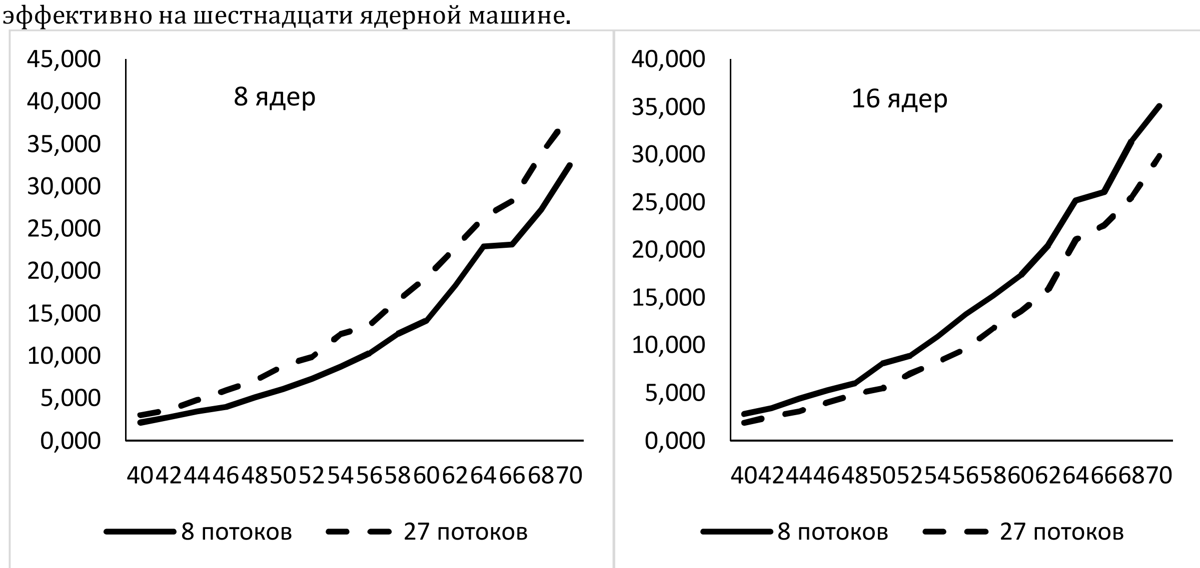


Рис. 2. Зависимость производительности алгоритма от числа вычислителей

Таким образом можно утверждать, что число параллельных потоков, реализующих алгоритм умножения многомерных матриц, должно быть близко к числу ядер (процессоров), на которых реализуется эта операция.

Проведенные исследования показали эффективность распараллеливания умножения многомерных матриц и позволили установить связь между физической и логической структурой программно-аппаратного комплекса, реализующего умножение.

Таким образом, можно утверждать, что получена архитектура программно-аппаратного комплекса, которая позволяет решать проблему массовой параллельной обработки больших объемов, интенсивно используемых структурированных данных, в тех случаях, когда эти данные могут быть представлены в виде не разреженных многомерных матриц.

Данная статья написана под руководством В. И. Мунермана.

Литература

1. Мунерман В.И. Многомерно-матричная модель массовой обработки данных. – Системы высокой доступности, № 3, 2012, т.8, с. 19-22.
2. Захаров В.Н., Мунерман В.И. Параллельная реализация обработки интенсивно используемых данных на основе алгебры многомерных матриц // Аналитика и управление данными в областях с интенсивным использованием данных: XVII Международная конференция DAMDID/RCDL'2015 (Обнинск, 13 - 16 октября 2015 года, Россия): Труды конференции - Обнинск: ИАТЭ НИЯУ МИФИ, 2015, с. 217 - 223. ISBN 978-5-9530-0398.
3. Захаров В.Н., Мунерман В.И. Параллельный алгоритм умножения многомерных матриц. – Современные информационные технологии и ИТ-образование. Т. 2 (№ 11), 2015. –614 с. – с. 384-391. (ISSN 2411-1473).
4. Мунерман В.И. Построение архитектур программно-аппаратных комплексов для повышения эффективности массовой обработки данных. – Системы высокой доступности, № 4, 2014, т.10, с. 3-16.
5. Мунерман В.И., Парфенов Н.В. Анализ параллельного алгоритма умножения многомерных матриц. Системы компьютерной математики и их приложения: материалы XVII Международной научной конференции. – Смоленск: Изд-во СмолГУ, 2016. – Вып. 17. – 268 с. ISBN 978-5-88018-445-3, продолжающееся издание. – с. 68-71.

References

1. V.I. Munerman. A multidimesional matrix model of mass data processing. – Highly available systems, № 3, 2012, t.8, s. 19-22.
2. Zakharov V. N., Munerman V. I. Parallel realization of processing of intensively used data on the basis of algebra of multidimensional matrixes//Analytics and data management in areas with intensive use of data: XVII International conference DAMDID/RCDL'2015 (Obninsk, on October 13 - 16, 2015, Russia): Works of a conference - Obninsk: IATE NIYaU MEPhI, 2015, page 217 - 223. ISBN 978-5-9530-0398.
3. Zakharov V. N., Munerman V. I. Parallel algorithm of multiplication of multidimensional matrixes. – Modern information technologies and IT education. T. 2 (No. 11), 2015. –614 pages – with. 384-391. (ISSN 2411-1473).
4. V.I. Munerman. Construction of hardware-software complexes architecture to improve massively data processing. – Highly available systems, № 4, 2014, t.10, s. 3-16.
5. V.I. Munerman, N.V. Parfenov. The analysis of parallel algorithms for multidimensional matrices multiplication. Systems of computer mathematics and their application: materials XVII of the International scientific conference. – Smolensk: Smolgu's publishing house, 2016. – Issue 17. – 268 pages of ISBN 978-5-88018-445-3, the proceeding edition. – page 68-71.

Поступила: 12.09.2016

Об авторе:

Парфенов Николай Владимирович, студент физико-математического факультета Смоленского государственного университета, npzxcf@gmail.com.