

Раздел III. Вычислительные и информационно-управляющие системы

УДК 004.75

DOI 10.18522/2311-3103-2024-6-164-176

В.М. Глушань, Л.А. Попов, А.А. Целых

О РЕАЛЬНЫХ ВОЗМОЖНОСТЯХ СОВРЕМЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ ДЛЯ РАСПРЕДЕЛЕННОГО УМНОЖЕНИЯ МАТРИЦ БОЛЬШИХ РАЗМЕРНОСТЕЙ

Потребности практики постоянно требуют повышения производительности вычислительных систем. Достаточно долгое время основным путём построения вычислительных систем сверхвысокой производительности были многопроцессорные системы. При создании таких систем возникает много сложных проблем. Они связаны с необходимостью распараллеливания вычислительного процесса для эффективной загрузки процессоров системы, преодолением конфликтов при попытках нескольких процессоров использовать один и тот же ресурс системы, уменьшением влияния конфликтов на производительность системы и т.д. С преодолением микроэлектроникой рубежа в миллиард транзисторов на кристалле кремния, появилась новая парадигма многоядерности процессоров. Одновременно с этим возникла проблема соотношения многоядерности и многопоточности в современных вычислителях. Это связано с дилеммой предпочтения между ними. Многоядерный процессор содержит два и более электронных вычислительных ядра, размещённых на одном полупроводниковом кристалле. Каждое ядро многоядерного процессора является полноценным микропроцессором. Многоядерность является очевидным и традиционным методом распределенного решения многих сложных задач. Но этого нельзя сказать о многопоточности, которая опирается на использование очень быстрой кэш-памяти, связанной с основной памятью и служащей для уменьшения среднего времени доступа к основной памяти процессора. Относительная новизна современных подходов к построению вычислительных систем требует сравнительных экспериментальных исследований их возможностей. Перспективным и удобным математическим объектом для этих целей является распределенное умножение матриц больших размерностей. В статье приводятся практические результаты распределенного умножения квадратных матриц с размерами от 300×300 до 2000×2000 и случайно сгенерированными значениями элементов в матрицах в диапазоне от -100 до $+100$. Исходя из экспериментальных данных, представленных в соответствующих таблицах и графиках, получены гиперболические соотношения для зависимости времени умножения матриц от числа виртуальных машин (ядер) в используемом ноутбуке. Аналогичные результаты получены при умножении квадратных матриц на однопроцессорных компьютерах, подключённых к локальной сети. Аналитические выражения в этом случае также представляют гиперболические временные зависимости. Но численные значения в них существенно превосходят таковые для гиперболической формулы, полученной для ноутбука. Исходя из полученных результатов, проведенные исследования позволяют сделать вывод, что использование однопроцессорного вычислителя, подключенного к локальной сети, для умножения матриц больших размерностей уступает быстрдействию ноутбука. Это связано со значительными затратами времени перемещения данных по локальной сети.

Матрицы; однопядерный и многоядерный процессор; многоядерность; многопоточность; гиперболическая зависимость.

V.M. Glushan, L.A. Popov, A.A. Tselykh

ABOUT THE REAL POSSIBILITIES OF MODERN COMPUTING SYSTEMS FOR DISTRIBUTED MULTIPLICATION OF LARGE-DIMENSIONAL MATRICES

The needs of practice constantly require improving the performance of computing systems. For quite a long time, multiprocessor systems have been the main way to build ultra-high performance computing systems. When creating such systems, many difficult problems arise. They are related to the need to

parallelize the computing process in order to efficiently load the system processors, overcome conflicts when several processors try to use the same system resource, reduce the impact of conflicts on system performance, etc. With microelectronics overcoming the milestone of a billion transistors on a silicon chip, a new paradigm of multicore processors has emerged. At the same time, the problem of the ratio of multicore and multithreading in modern computers arose. This is due to the dilemma of preference between them. A multicore processor contains two or more electronic computing cores placed on a single semiconductor crystal. Each core of a multicore processor is a full-fledged microprocessor. Multicore is an obvious and traditional method of distributed solution of many complex tasks. But this cannot be said about multithreading, which relies on the use of very fast cache memory associated with the main memory and serves to reduce the average access time to the main memory of the processor. The relative novelty of modern approaches to the construction of computing systems requires comparative experimental studies of their capabilities. A promising and convenient mathematical object for these purposes is the distributed multiplication of matrices of large dimensions. The article presents practical results of distributed multiplication of square matrices with sizes from 300×300 to 2000×2000 and randomly generated values of elements in the matrices in the range from -100 to $+100$. Based on the experimental data presented in the corresponding tables and graphs, hyperbolic relations are obtained for the dependence of the matrix multiplication time on the number of virtual machines (cores) in the laptop used. Similar results were obtained by multiplying square matrices on single-processor computers connected to a local network. Analytical expressions in this case also represent hyperbolic time dependencies. But the numerical values in them significantly exceed those for the hyperbolic formula obtained for the laptop. Based on the results obtained, the conducted research allows us to conclude that the use of a single-processor computer connected to a local network for multiplying matrices of large dimensions is inferior to the performance of a laptop. This is due to the significant time spent moving data over the local network.

Matrices; single-core and multi-core processor; multicore; multithreading; hyperbolic dependence.

Введение. Принято считать, что эра компьютеризации началась в 1975 году, когда фирма MITS (Micro Instrumentation and Telemetry Systems) выпустила в свет первый персональный компьютер «Альтаир 8800». Это стало толчком для все увеличивающегося выпуска новых операционных систем, дисководов для оперативной записи и считывания информации с большого многообразия дисков. Компьютеризация стала шагать по планете с невиданной скоростью. В 1983 году появился первый удобный в использовании ПК Apple Macintosh. Наша страна тоже старалась не отставать, выпустив первый отечественный ПК «АГАТ».

Потребности практики постоянно требовали повышения производительности вычислительных систем. Достаточно долгое время основным путём построения вычислительных систем сверхвысокой производительности были многопроцессорные системы [1, 2]. При создании таких систем возникает много сложных проблем, связанных с необходимостью распараллеливания вычислительного процесса (программ) для эффективной загрузки процессоров системы, преодолением конфликтов при попытках нескольких процессоров использовать один и тот же ресурс системы (например, некоторый модуль памяти), уменьшением влияния конфликтов на производительность системы и т.д.

На рубеже тысячелетий, когда микроэлектроника преодолела рубеж в миллиард транзисторов на кристалле кремния, появилась новая парадигма многоядерности процессоров. Практически одновременно возникла проблема соотношения многоядерности и многопоточности в современных вычислителях. Это связано с дилеммой предпочтения между ними.

Считается, что широкое распространение многоядерная архитектура получала лишь в 2005 году, когда компании AMD и Intel выпустили свои первые двухъядерные процессоры. Многоядерный процессор содержит две и более электронные вычислительные схемы. Эти схемы называют ядрами и размещают их на одном полупроводниковом кристалле или в одном корпусе. Каждое ядро многоядерного процессора является полноценным микропроцессором, для которого характерны черты всех современных процессоров.

Если многоядерность является очевидным и традиционным методом распределенного решения многих сложных задач, то этого нельзя сказать о многопоточности. Многопоточность опирается на использование очень быстрой кэш-памяти (сверх оперативной памяти), связанной с основной памятью и служащей для уменьшения среднего вре-

мени доступа к основной памяти процессора. Кэш-память хранит копии часто используемых данных из основной памяти. Если большая часть запросов в основную память будет обрабатываться посредством использования кэш-памяти, то средняя задержка обращения к основной памяти будет определяться задержкой именно кэш-памяти, а не основной памяти [3].

Парадигма многопоточности стала очень привлекательной для многих производителей процессоров. Поэтому многие производители многопоточных процессоров в рекламных целях акцентируют внимание потребителей на максимальном суммарном числе потоков в их процессорах. И нередко многочисленные потоки представляются как «логические процессоры» или «виртуальные ядра» [4].

Следует заметить, что многопоточность не всегда приводит к положительному эффекту. А он проявляется лишь тогда, когда все исполняемые ядром вычислителя потоки принадлежат одной и той же программе. Но это возможно лишь при условии слабой взаимозависимости потоков по данным. А это значит, что запуск нескольких программ разными потоками на одном ядре может лишь ухудшить ситуацию из-за постоянных конфликтов в кэшах. «Поэтому нет ничего удивительного в том, что нередко при решении на многопоточном процессорном ядре тестовых задач, запрограммированных для разного числа потоков, самая высокая производительность ядра достигалась при работе с одним единственным потоком» [5].

В добавление ко всему следует отметить, что параллельные вычисления наталкиваются на ограничения, обусловленные законом Амдала [6–9], предложенного им в 1967 году. До сих пор в специализированной литературе по параллельным вычислениям этот закон является основополагающим. Он определяет теоретическое ограничение для скорости выполнения некоторой предназначенной для этого программы при распараллеливании процесса решения соответствующей задачи.

В завершение вводной части настоящей статьи для её читателей будет интересен, по нашему мнению, многозначный вывод из [10]: «Изготовители процессоров иногда рекламируют многопоточность своих изделий как альтернативу многоядерности, поток не эквивалентен процессорному ядру и тем более процессору, а многопоточность не заменяет многоядерность. В результате на рынке представлены процессорные ядра с разным числом потоков. Есть и примеры принципиального игнорирования многопоточности. Если многопоточность – не замена многоядерности, то многоядерность во многих случаях может успешно заменить многопоточность. Она более универсальна, поскольку применима для распараллеливания и потоков одной задачи, и целиком задач».

Экспериментальные исследования умножения матриц больших размерностей на двух типах вычислительных систем. Умножение матриц используется во многих практических задачах. При умножении матриц небольших размерностей особых трудностей не возникает. Они появляются, когда приходится умножать матрицы с тысячами и миллионами строк и столбцов. Умножение матриц является той областью вычислений, которая позволяет эффективно использовать все вычислительные ресурсы современных процессоров.

В [11] отмечается, что «распределенные вычислительные системы – это сформировавшаяся сфера высокопроизводительных вычислений, обладающая своей спецификой, ярко выраженным классом решаемых задач и методами их решения. Разрабатываются и внедряются новые концепции построения распределенных систем, расширяется круг решаемых ими задач, упрощается процесс организации, разрабатываются более простые методы использования ресурсов конечными пользователями».

В качестве примера достаточно обратиться к статьям [12–15], в которых рассматриваются вопросы оптимального блочного разбиения исходных матриц и их умножение на заданном числе процессоров. Отличие указанных статей от всех известных работ носит двойственный характер. Во-первых, речь идет об умножении не квадратных, а о матрицах более общего вида – прямоугольных согласованных матрицах. Во-вторых, рассматривается вопрос об наилучшем (оптимальном) разбиении таких матриц на блоки и соответствующее их умножение на заданном числе процессоров.

В указанных работах вычислительные процессы были ориентированы на использование однопроцессорных клиент-серверных технологий. В настоящей статье приводятся сравнительные результаты умножения квадратных матриц на компьютере (ноутбуке) с многоядерным процессором и на нескольких однопроцессорных компьютерах, подключенных к локальной сети.

Экспериментальные результаты умножения квадратных матриц на ноутбуке с многоядерным процессором. Для распределённого умножения был разработан код программы [16] на языке Java. Используя данную программу проведены экспериментальные исследования умножения квадратных матриц на одном многоядерном процессоре [17]. В процессе экспериментов измерялось время умножения матриц в зависимости от числа задействованных в процессоре ядер. Цель эксперимента заключалась в практическом определении эффективности использования многоядерного процессора при умножении квадратных матриц. Следует отметить, что предварительные результаты экспериментов при одних и тех же исходных данных имели некоторый дрейф. Поэтому с целью исключения его влияния на точность экспериментов замеры времени вычислений для одних и тех же исходных данных проводились многократно, а затем усреднялись. О целесообразности применения такой рекомендации в проведении экспериментов с применением многоядерных процессоров рекомендована и в работе [18].

В экспериментах использовался ноутбук со следующими характеристиками:

- операционная система Windows 10;
- процессор AMD Ryzen 5 Mobile 4600H: 6 ядер (12 логических процессоров) [19];
- оперативная память 40 Гбайт, 3200 МГц.

Для эмуляции работы программы на одном компьютере было использовано средство виртуализации Oracle VM Virtual Box версии 7.0.12. В эксперименте было задействовано 5 виртуальных машин со следующими параметрами:

- операционная система Windows 10;
- одно ядро процессора (2 логических процессора);
- оперативная память 4 Гбайт, 3200 МГц.

Для перемножения использовались две случайно сгенерированные целочисленные матрицы размером 300×300 . Значения отдельных элементов матриц находились в диапазоне $-100 < x < 100$.

В процессе проведения эксперимента учитывалось время всего процесса перемножения. Здесь нужно заметить, что временем разбиения матриц на блоки, переноса данных по виртуальной сети и построения результирующей матрицы из блоков можно пренебречь, поскольку оно занимает значительно меньше времени, чем время процесса собственно умножения.

В процессе выполнения эксперимента гостевая машина выполняла роль сервера, а виртуальные машины – роль клиентов. В задачи сервера входило разбиение исходных матриц на блоки, передача полученных блочных столбцов и строк на клиенты, получение результирующих блоков с клиентов и построение результирующей матрицы. Перемножение блоков на гостевую машину (сервер) не возлагалось. Клиенты получали блочные строки и столбцы с сервера, выполняли вычисление своего блока и отправляли полученный результат обратно серверу.

На k -ой итерации эксперимента было задействовано k клиентов и один сервер. Таким образом, в первой итерации участвовали гостевая машина и одна виртуальная, во второй – гостевая и две виртуальные, и т.д. Всего было проведено 5 итераций. При этом на каждой итерации было произведено 5 замеров времени, а затем вычислено его среднее значение. Результаты экспериментов приведены в табл. 1

Таблица 1

Число серверов	Общее время выполнения (мс)					
	1	2	3	4	5	Среднее
1	79	115	93	79	91	91,4
2	65	53	57	52	55	56,4
3	47	50	42	49	42	46
4	45	44	40	39	41	41,8
5	39	40	39	36	41	39

Графические результаты приведены на рис. 1. На оси абсцисс указано число задействованных виртуальных машин (ядер). Оно совпадает с номером итерации эксперимента, но на единицу меньше числа задействованных в эксперименте ядер процессора. На оси ординат указано среднее время умножения матриц в миллисекундах.

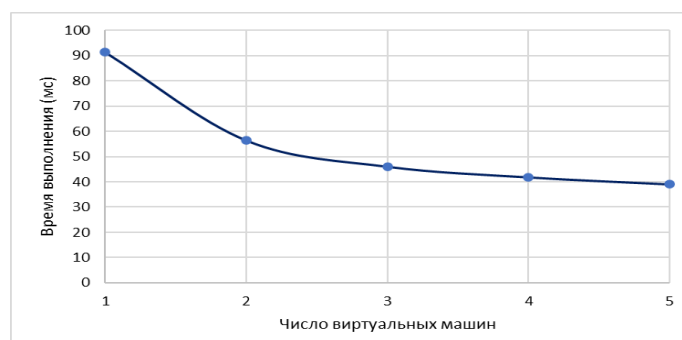


Рис. 1. Изменение времени умножения матриц от числа виртуальных машин

Используя метод гиперболической регрессии [20], получена формула зависимости времени умножения $f(x)$ от числа задействованных виртуальных машин x

$$f(x) = 24,78 + \frac{66,01}{x}. \quad (1)$$

Статистические характеристики (коэффициент корреляции – КК и коэффициент детерминации – R^2) для формулы (1) принимают соответственно значения: КК = 0,9989, $R^2 = 0,9977$. Эти значения свидетельствуют о высокой степени согласованности формулы (1) с экспериментальными данными.

Полученные результаты показывают, что увеличение числа задействованных в экспериментах ядер процессора имеет смещенную на постоянную величину обратно пропорциональную зависимость (гиперболическую) уменьшения времени распределённого умножения матриц. Постоянная величина (const) 24,78 в приведенной формуле вызвана неизбежными, но экспериментально трудно и однозначно учитываемыми накладными расходами в решении задачи.

Как с теоретической, так и практической точек зрения интерес представляет зависимость времени умножения матриц от числа процессоров (виртуальных машин), рассчитанных по формуле (1) как без учета, так и с учетом постоянной составляющей. В этом случае эта зависимость, являясь идеализированной, будет показывать в «размах» увеличение быстродействия в зависимости от использованного числа ядер. Эти результаты приведены в табл. 2, исходя из анализа табл. 1

Таблица 2

№	Среднее значение эксперимента	Расчётное значение времени по формуле (1) без const	В размах без const	Расчётное значение времени по формуле (1) с const	В размах с const
1	91,4	65,9	1,0	90,76	1,0
2	56,4	32,95	2,0	57,81	1,57
3	46,0	21,96	3,0	46,82	1,93
4	41,8	16,48	4,0	41,34	2,19
5	39,0	13,18	5,0	38,68	2,34

Наиболее вероятной предполагаемой причиной появления постоянной составляющей является использование в современных многоядерных процессорах кэш-памяти (быстрой). Её использование объясняется тем, что оперативное запоминающее устройство (ОЗУ) работает гораздо медленнее, чем арифметическое устройство (АУ) процессора. Это приводит к тому, что АУ часто оказывается в состояниях ожидания прихода из ОЗУ нужных для обработки данных. Для предотвращения простоев в полупроводниковый кристалл АУ встраивается дополнительная быстрая, так называемая кэш-память. Она является буфером между ОЗУ и процессором и предназначена для хранения данных, с которыми процессор должен работать в текущий момент времени. Таким образом, кэш-память сглаживает разницу в скорости работы АУ и ОЗУ. Благодаря этому в целом вычисления производятся быстрее, чем это было бы без применения кэш-памяти.

Однако полностью исключить простои невозможно, но их желательно свести к минимуму. Это достигается использованием в системе процессорного кэша специального контроллера. Основная задача контроллера состоит в минимизации числа простоев АУ. Для этого контроллер должен уметь предсказывать ситуации для своевременного считывания из ОЗУ необходимых данных для их обработки в АУ. Этот процесс называется «интеллектуальной стратегией кэширования», которая основана на анализе уже имеющихся ошибок. Эта стратегия строится на основе искусственного интеллекта, позволяющего контроллеру заблаговременно определить, какие данные могут понадобиться АУ. Если он «угадал», то запоминает это событие как удачное, если не «угадал», то бракует его, и меняет логику выбора данных для предварительной загрузки.

Экспериментальные результаты умножения квадратных матриц на однопроцессорных компьютерах, подключённых к локальной сети. Эксперименты проводились с однотипными компьютерами Intel Core i3-10100 3.6 ГГц/4 ядра/ОЗУ 8 Гб/SSD 240 Гб/ОС Windows 10, которые были подключены к локальной сети Ethernet со скоростью 1 Гбит/с. Всего в экспериментах использовалось 6 компьютеров.

При этом ставилась задача проведения двух типов экспериментов. В эксперименте №1 осуществлялось умножение матриц в локальной сети с использованием только одного сервера (процессора). Использовались случайно сгенерированные квадратные матрицы размерностью от 300×300 до 900×900 . Этот эксперимент должен был дать ответ на вопрос о быстродействии процесса блочного умножения матриц больших размерностей для случая, когда один из процессоров исполнял роль сервера (ведущего), а остальные 5 процессоров (ведомые серверы) выполняли умножение соответствующих блоков исходных матриц. Сервер (ведущий) выполнял задачу разбиения исходной матрицы на блоки, передачу их на 5 ведомых процессоров и осуществлял сборку результирующей матрицы.

В эксперименте №2 выполнялось умножение матриц в локальной сети с использованием от одного до 5 серверов. Один из серверов использовался в качестве управляющего. Для реализации процесса умножения были случайно сгенерированы квадратные матрицы размерностью 2000×2000 . Как и в эксперименте с ноутбуком значения отдельных элементов матриц находились в диапазоне $-100 < x < 100$.

Результаты эксперимента № 1. Табл. 3 результатов зависимости общего времени выполнения операции умножения (включая время на разбиение матриц на блоки и сборки результирующей матрицы, время на передачу данных (блоков) по сети и время на собственно умножения матриц) от размерности матриц имеет вид

Таблица 3

Размерность	Общее время выполнения (мс)					
	1	2	3	4	5	Среднее
300	187	171	484	484	484	362
400	312	593	609	624	629	553,4
500	796	812	749	827	515	739,8
600	1064	1046	754	781	1062	941,4
700	1421	1426	1093	1390	1109	1287,8
800	1874	1609	1922	1597	1874	1775,2
900	2530	2582	2484	2639	2624	2571,8

График полученной зависимости приведен на рис. 2

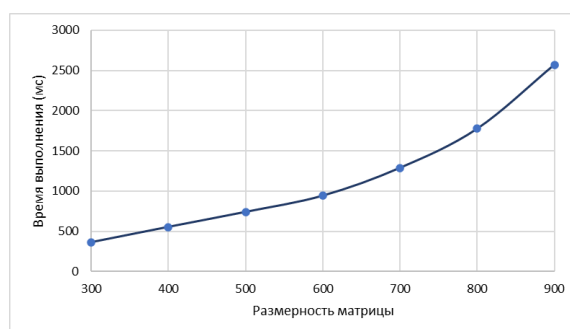


Рис. 2. Зависимость времени операции умножения от размерности матриц

В соответствии с методом гиперболической регрессии эта зависимость в аналитическом виде принимает вид

$$y = 0,00001x^3 - 0,01626x^2 + 8,8923x - 1175,7143. \quad (2)$$

Статистические характеристики для выражения (2) принимают значения: $KK = 0,9999$, $R^2 = 0,9998$, которые свидетельствуют о высокой степени согласованности формулы с экспериментальными данными.

Результаты времени передачи данных по сети в зависимости от размерности матриц приведены в табл. 4, а график – на рис. 3.

Таблица 4

Размерность	Время передачи данных на сети (мс)					
	1	2	3	4	5	Среднее
300	109	93	406	406	406	284
400	172	453	485	469	505	416,8
500	562	563	546	578	266	503
600	674	672	380	391	672	557,8
700	812	801	515	796	516	688
800	953	672	985	676	984	854
900	1141	1098	1141	1140	1156	1135,2

Необходимо подчеркнуть, что данные в табл. 4 имеют большой разброс, что тоже может сказываться на точности результатов экспериментов.

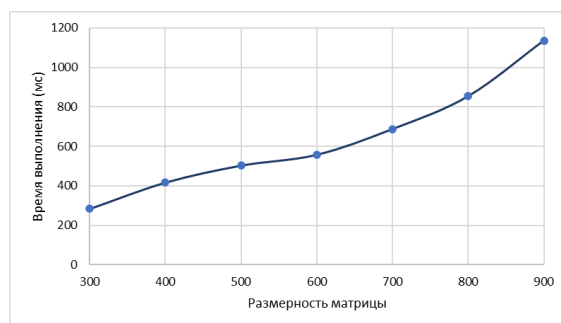


Рис. 3. Зависимость времени передачи данных по сети от размерности матрицы

Эта зависимость в аналитическом виде принимает вид

$$y = 0,00001x^3 - 0,00991x^2 + 5,86965x - 754,81905. \quad (3)$$

Статистические характеристики для выражения (3) принимают значения: $KK = 0,9997$, $R^2 = 0,9993$, которые также свидетельствуют о высокой степени согласованности формулы с экспериментальными данными.

Зависимость времени умножения матриц от их размерности без учета времени их передачи по сети приведена в табл. 5, а графическое представление этой зависимости приведено на рис. 4

Таблица 5

Размерность	Время умножения (мс)					
	1	2	3	4	5	Среднее
300	78	78	78	78	78	78
400	140	140	124	140	124	133,6
500	234	249	203	234	249	233,8
600	390	374	374	390	374	380,4
700	609	609	562	578	593	590,2
800	921	937	937	921	890	921,2
900	1374	1484	1327	1499	1468	1430,4

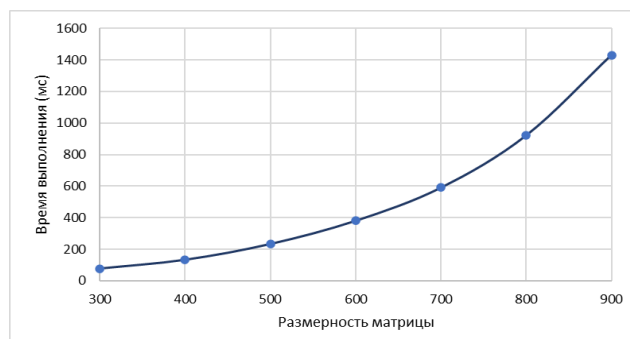


Рис. 4. Зависимость времени умножения матриц от их размерности без учета времени их передачи

Эта зависимость в аналитическом виде принимает вид

$$y = 0,00001x^3 - 0,00620x^2 + 2,91649x - 400,58095. \quad (4)$$

Статистические характеристики для выражения (4) принимают значения: $KK = 0,9999$, $R^2 = 0,9997$, которые также свидетельствуют о высокой степени согласованности формулы с экспериментальными данными.

Результаты эксперимента № 2. В табл. 6 приведены результаты общего времени выполнения операции умножения от числа задействованных серверов, а на рис. 5 – график полученной зависимости.

Таблица 6

Число серверов	Общее время выполнения (мс)					
	1	2	3	4	5	Среднее
1	49566	49958	49145	48976	49830	49495
2	27084	27690	29194	29717	29327	28602,4
3	22925	22514	23278	22848	22349	22782,8
4	21706	19434	19703	20267	19842	20190,4
5	18636	18260	18665	17782	18369	18342,4

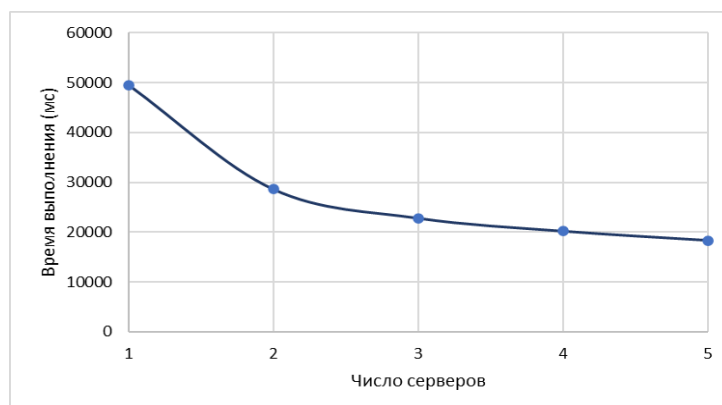


Рис. 5. Зависимость общего времени умножения от числа серверов

Эту зависимость в аналитическом виде можно представлять двумя равносильными выражениями:

$$y = 10043,3725 + \frac{39063,5639}{x}, \quad (5)$$

$$-1194,0833x^3 + 13697,8929x^2 - 52959,0238x + 89816,4000. \quad (6)$$

Статистические характеристики для выражения (5) принимают значения: $KK = 0,9988$, $R^2 = 0,9976$, а для выражения (6) соответственно: $KK = 0,9990$, $R^2 = 0,9981$. Эти значения свидетельствуют о статистической равнозначности формул (5) и (6) и их высокой степени согласованности с экспериментальными данными.

Поскольку таблица 6 и соответствующий рис. 5 отражают общее время выполнения процесса умножения матриц, то также представляет интерес получения временных результатов распределенного умножения матриц без учета времени передач данных по сети. Результаты передачи данных по сети представлены в табл. 7 и на рис. 6.

Таблица 7

Число серверов	Время передачи данных по сети (мс)					
	1	2	3	4	5	Среднее
1	4140	4454	4390	4425	4669	4415,6
2	3950	5618	5528	5145	5099	5068
3	5929	5658	6267	6274	5962	6018
4	7335	7391	7972	7520	7985	7640,6
5	8857	8544	8527	7995	8942	8573

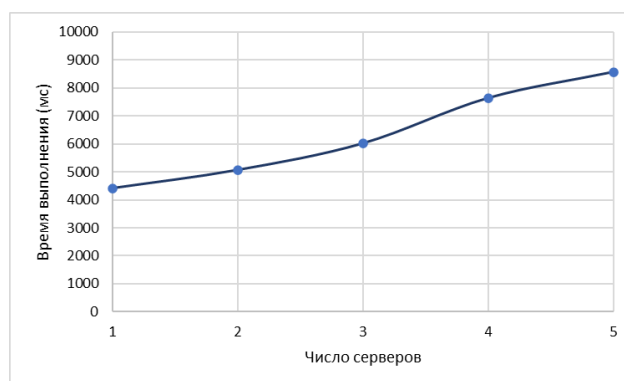


Рис. 6. Зависимость времени передачи данных по сети от числа задействованных серверов

Эта зависимость в аналитическом виде принимает вид

$$-82,4167x^3 + 829,8214x^2 - 1384,7619x - 5078,2000. \quad (7)$$

Статистические характеристики для выражения (7) принимают значения: $KK = 0,9982$, $R^2 = 0,9964$, которые свидетельствуют о высокой степени согласованности формулы с экспериментальными данными.

Зависимость времени умножения матриц от числа задействованных серверов без учета времени передачи данных по сети приведена в табл. 8, а графическое представление этих результатов приведено на рис. 7.

Эта зависимость в аналитическом виде принимает вид

$$y = 1138,4587 + \frac{44144,1051}{x}. \quad (8)$$

Статистические характеристики для выражения (8) принимают значения: $KK = 0,9992$, $R^2 = 0,9984$, которые также свидетельствуют о высокой степени согласованности формулы с экспериментальными данными.

Таблица 8

Число серверов	Общее время выполнения (мс)					
	1	2	3	4	5	Среднее
1	45426	45504	44739	44551	45145	45073
2	22619	22072	23650	24556	24213	23422
3	16996	16199	17011	16574	16371	16630,2
4	12950	12028	11528	12731	11700	12187,4
5	9310	8919	8951	9271	9427	9175,6

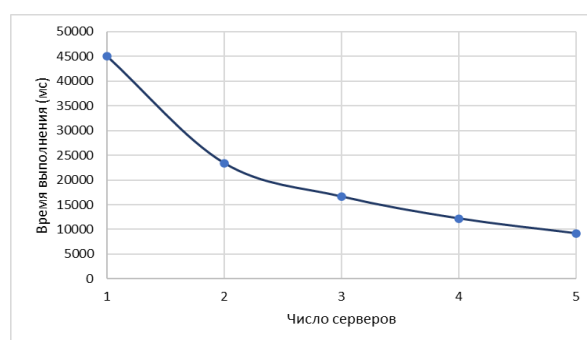


Рис. 7. Зависимость времени умножения матриц от числа серверов

Выводы и заключение. Приведенные материалы исследований процесса распределенного умножения матриц больших размерностей различными вычислительными средствами и их аналитическая обработка привели к неожиданному, в некоторой степени, результату. Он заключается в том, что используемый в исследованиях ноутбук с многоядерными процессорами оказался более быстродействующим вычислителем, чем однопроцессорный компьютер, подключенный к локальной сети. Одним из объяснений данного результата (феномена) можно считать близость нескольких процессорных ядер на одном кристалле в многоядерном процессоре. Поэтому электрические сигналы пробегает значительно меньшие расстояния, чем в локальной сети. Это позволяет благодаря использованию различных уровней кэш-памяти работать многоядерному процессору с гораздо более высокой тактовой частотой.

Проведенные исследования позволяют также сделать вывод, что использование однопроцессорного вычислителя в локальной сети для умножения матриц больших размерностей является не рациональным, поскольку время на перемещение данных по локальной сети перевешивает возможные выгоды от распараллеливания. Однако для окончательной убедительности необходимо проведение дополнительных исследований с различными моделями ноутбуков.

Кроме того, проведенные исследования наталкивают на мысль, что сравнительное распределенное умножение матриц больших размерностей можно использовать для оценки эффективности распределенных вычислений различными средствами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Каляев И.А. и др.* Реконфигурируемые мультиконвейерные вычислительные структуры. – 2-е изд., перераб. и доп. / под общ. ред. И.А. Каляева. – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. – 344 с.
2. *Воеводин И.Б., Воеводин Вл.В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2004. – 608 с.
3. *Ивашко Е.Е.* Современные технологии высокопроизводительных вычислений. – Петрозаводск, 2016. – 61 с.

4. <https://habr.com/ru/companies/intel/articles/243385/>.
5. Кадомский А.А., Захаров В.А.. Эффективность многопоточных приложений // Научный журнал. – 2016. – № 7 (8). – С. 26-28.
6. Закон Амдала. – URL:<https://dic.academic.ru/dic.nsf/ruwiki/433588>.
7. Закон Амдала и будущее многоядерных процессоров. – URL:<https://www.osp.ru/os/2009/04/9288815>.
8. Глушань В.М., Лаврик П.В. Ограничение быстродействия вычислительных систем в результате совмещения закона Амдала и гипотезы Минского // Тр. Конгресса по интеллектуальным системам и информационным технологиям «IS-IT'13». Научное издание в 4-х т. Т. 1. – М.: Физматлит, 2013. – С. 129-137.
9. Павлов А.В. Архитектура вычислительных систем. – СПб: Университет ИТМО, 2016. – 86 с.
10. <https://www.kakprosto.ru/kak-901137-mnogoyadernye-processory-principy-rabot>.
11. Радченко Г.И. Распределенные вычислительные системы. – Челябинск: Фотохудожник, 2012. – 184 с.
12. Глушань В.М., Красюк О.И., Лозовой А.Ю. Параллельное умножение матриц больших размерностей на заданном множестве процессоров // Вестник РГРТУ. – 2020. – № 74. – С. 42-55.
13. Глушань В.М. Расширение функционала клиент-серверной подсистемы распределенного проектирования СБИС // Информатизация и связь. – 2021. – № 3. – С. 33-38.
14. Glushan V.M., Lozovoy A.Yu. On Distributed Multiplication of Large-Scale Matrices // 2021 IEEE 15th International Conference on Application of Information and Communication Technologies (AICT). – 2021. – P. 1-4.
15. Глушань В.М. Иерархические клиент-серверные архитектуры в распределенном проектировании и параллельных вычислениях // Современные компьютерные технологии: Матер. II научно-методической конференции НПП. – Таганрог, 2021. – С. 14-18.
16. Глушань В.М., Попов Л.А. Распределенное умножение матриц на компьютерах с многоядерными процессорами // Тр. Международного научно-технического конгресса «Интеллектуальные системы и информационные технологии – 2024» («IS&IT»). Научное издание в 2-х т. Т. 2. – Таганрог: Изд-во Ступина С.А., 2024. – С. 65-70.
17. Глушань В.М., Попов Л.А. Программа умножения матриц на компьютере с многоядерными процессорами. Свидетельство о государственной регистрации программы для ЭВМ № 2024663001, зарегистрировано 03.06.2024.
18. Соснин В.В., Балакишин П.В. Введение в параллельные вычисления. – СПб: Университет ИТМО, 2015. – 51 с.
19. AMD Ryzen™ 5 4600H Drivers & Support. – <https://www.amd.com/en/support/apu/amd-ryzen-processors/amd-ryzen-5-mobile-processors-radeon-graphics/amd-ryzen-5-4600h>.
20. Онлайн калькулятор: Аппроксимация функции одной переменной // PlanetCalc. – Режим доступа: URL: <https://planetcalc.ru/5992/> (дата обращения: 29.04.2024).

REFERENCES

1. Kalyaev I.A. i dr. Rekonfiguriruemye mul'tikonveyernye vychislitel'nye struktury [Reconfigurable multiconveyor computing structures]. 2nd edition, reprint. and additional, Under the general editorship of I.A. Kalyaev. Rostov-on-Don: Izd-vo YuNTS RAN, 2009, 344 p.
2. Voevodin I.B., Voevodin V.V. Parallel'nye vychisleniya [Parallel computing]. Saint Petersburg: BKhV-Peterburg, 2004, 608 p.
3. Ivashko E.E. Sovremennyye tekhnologii vysokoproizvoditel'nykh vychisleniy [Modern technologies of high-performance computing]. Petrozavodsk, 2016, 61 p.
4. Available at: <https://habr.com/ru/companies/intel/articles/243385/>.
5. Kadomskiy A.A., Zakharov V.A. Effektivnost' mnogopotochnykh prilozheniy [Efficiency of multi-threaded applications], *Nauchnyy zhurnal* [Scientific journal], 2016, No. 7 (8), pp. 26-28.
6. Zakon Amdala [Amdahl's Law]. Available at: <https://dic.academic.ru/dic.nsf/ruwiki/433588>.
7. Zakon Amdala i budushchee mnogoyadernykh protsessorov [Amdahl's law and the future of multicore processors]. Available at: <https://www.osp.ru/os/2009/04/9288815>.
8. Glushan' V.M., Lavrik P.V. Ogranichenie bystrodeystviya vychislitel'nykh sistem v rezul'tate sovmeshcheniya zakona Amdala i gipotezy Minskogo [Limitation of the speed of computing systems as a result of combining Amdahl's law and the Minsk hypothesis], *Tr. Kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «IS-IT'13»* [Proceedings of the Congress on Intelligent Systems and Information Technologies "IS-IT'13"]. Scientific publication in 4 vol. Vol. 1. Moscow: Fizmatlit, 2013, pp. 129-137.
9. Pavlov A.V. Arkhitektura vychislitel'nykh sistem [Architecture of computing systems]. Saint Petersburg: Universitet ITMO, 2016, 86 p.
10. Available at: <https://www.kakprosto.ru/kak-901137-mnogoyadernye-processory-principy-rabot>.

11. Radchenko G.I. Rasprelenyie vychislitel'nye sistemy [Distributed computing systems]. Chelyabinsk: Fotokhudozhnik, 2012, 184 p.
12. Glushan' V.M., Krasnyuk O.I., Lozovoy A.Yu. Parallel'noe umnozhenie matrits bol'shikh razmernostey na zadannom mnozhestve protsessorov [Parallel multiplication of large-dimensional matrices on a given set of processors], *Vestnik RGTU* [Bulletin of the Russian State Technical University], 2020, No. 74, pp. 42-55.
13. Glushan' V.M. Rasshirenie funktsionala klient-servernoy podsistemy raspredelenogo proektirovaniya SBIS [Expanding the functionality of the client-server subsystem of distributed VLSI design], *Informatizatsiya i svyaz'* [Informatization and Communication], 2021, No. 3, pp. 33-38.
14. Glushan V.M., Lozovoy A.Yu. On Distributed Multiplication of Large-Scale Matrices, 2021 *IEEE 15th International Conference on Application of Information and Communication Technologies (AICT)*, 2021, pp. 1-4.
15. Glushan' V.M. Ierarkhicheskie klient-servernye arkhitektury v raspredelennom proektirovanii i parallel'nykh vychisleniyakh [Hierarchical client-server architectures in distributed design and parallel computing], *Sovremennye komp'yuternye tekhnologii: Mater. II nauchno-metodicheskoy konferentsii NPR* [Modern computer technologies: materials of the II scientific and methodological conference of the NPR]. Taganrog, 2021, pp. 14-18.
16. Glushan' V.M., Popov L.A. Raspredelennoe umnozhenie matrits na komp'yuterakh s mnogoyadernymi protsessorami [Distributed matrix multiplication on computers with multicore processors], *Tr. Mezhdunarodnogo nauchno-tekhnicheskogo kongressa «Intellectual'nye sistemy i informatsionnye tekhnologii – 2024» («IS&IT'»)* [Proceedings of the International Scientific and Technical Congress "Intelligent Systems and Information Technologies - 2024" ("IS&IT")]. Scientific publication in 2 vol. Vol. 2. Taganrog: Izd-vo Stupina S.A., 2024, pp. 65-70.
17. Glushan' V.M., Popov L.A. Programma umnozheniya matrits na komp'yutere s mnogoyadernymi protsessorami. Svidetel'stvo o gosudarstvennoy registratsii programmy dlya EVM № 2024663001, zaregistrirvano 03.06.2024 [Matrix multiplication program on a computer with multicore processors. Certificate of state registration of the computer program No. 2024663001, registered on 06/03/2024].
18. Sosnin V.V., Balakshin P.V. Vvedenie v parallel'nye vychisleniya [Introduction to parallel computing]. Saint Petersburg: Universitet ITMO, 2015, 51 p.
19. AMD Ryzen™ 5 4600H Drivers & Support. Available at: <https://www.amd.com/en/support/apu/amd-ryzen-processors/amd-ryzen-5-mobile-processors-radeon-graphics/amd-ryzen-5-4600h>.
20. Onlayn kal'kulyator: Approssimatsiya funktsii odnoy peremennoy [Online calculator: Approximation of a function of one variable], *PlanetCalc*. Available at: <https://planetcalc.ru/5992/> (accessed 29 April 2024).

Глушань Валентин Михайлович – Южный федеральный университет; e-mail: vmglushan@sfedu.ru; г. Таганрог, Россия; кафедра систем автоматизированного проектирования имени проф. В.М. Курейчика; профессор.

Попов Леонид Андреевич – e-mail: lpopov@sfedu.ru; кафедра систем автоматизированного проектирования имени проф. В.М. Курейчика; студент.

Целых Алексей Александрович – e-mail: tselykh@sfedu.ru; кафедра информационно-аналитических систем безопасности имени проф. Л.С. Берштейна; доцент.

Glushan Valentin Mihailovich – Southern Federal University; e-mail: vmglushan@sfedu.ru; Taganrog, Russia; the Department of Computer Aided Design named after V.M. Kureichik; professor.

Popov Leonid Andreevich – Southern Federal University, e-mail: lpopov@sfedu.ru; the Department of Computer Aided Design named after V.M. Kureichik; student.

Tselykh Alexey Alexandrovich – e-mail: tselykh@sfedu.ru, the Department of Information and Analytical Security Systems; associate professor.