

А.И. Захаров, Г.А. Брякалов, К.А. Неретина

---

ВЛИЯНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ И СТРУКТУРЫ  
АЛГОРИТМОВ РЕШАЕМЫХ ЗАДАЧ НА ОПЕРАТИВНОСТЬ  
ОБРАБОТКИ ИНФОРМАЦИИ В МНОГОПРОЦЕССОРНЫХ  
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

---

Рассмотрена проблематика декомпозиции структур алгоритмов обработки информации в многопроцессорных системах. Представлены этапы процесса распараллеливания алгоритмов, связанного с их дальнейшей программной реализацией. Выделены группы алгоритмов решаемых задач. В качестве примера анализа одного из алгоритмов приведено решение задачи дистанционного зондирования Земли из космоса вместе с ее математической постановкой. Выявлено, что параллельные структуры алгоритмов решаемых задач и аппаратных средств, а также параллельное программирование способствуют достижению необходимого параллелизма вычислительных процессов, росту производительности работы вычислительных средств и повышению оперативности обработки информации. Статья носит прикладной характер и может быть полезна для лиц, интересующихся вопросами повышения оперативности обработки информации.

*Ключевые слова:* распараллеливание алгоритмов, обработка информации, многопроцессорные системы, параллельное программирование.

A.I. Zakharov, G.A. Bryakalov, K.A. Neretina

---

INFLUENCE OF PARALLEL CALCULATIONS AND STRUCTURE  
OF ALGORITHMS SOLVABLE TASKS ON EFFICIENCY OF INFORMATION  
PROCESSING IN MULTIPROCESSOR COMPUTING SYSTEMS

---

The article is devoted to the decomposition algorithms structure's problematics of information processing operation in multiprocessor systems. The stages of the process of parallelization of algorithms associated with their further software implementation are presented. Allocated groups of algorithms for the tasks to be solved. As an example of the analysis of one of the algorithms, the solution of the problem of remote sensing of the Earth from space, together with its mathematical formulation, is presented. It has been revealed that parallel structures of algorithms for solved problems and hardware, as well as parallel programming, contribute to the achievement of the necessary parallelism of computing processes, an increase in the performance of computing tools and an increase in the efficiency of information processing. The article is of an applied nature and can be useful for people interested in improving the efficiency of information processing.

*Keywords:* parallelization of algorithms, information processing operation, multiprocessor systems, parallel programming.

*Вводные замечания*

Общеизвестно, что решение специальных задач, в частности задач космической деятельности, основывается на законах баллистики, механики и теории полета, а также не-

---

### Анализ данных и интеллектуальные системы

разрывно связано с обработкой больших объемов информации за расчетное время как на этапах создания и запуска космических аппаратов (КА), так и в ходе сеансов управления ими [8]. В связи с этим возникает необходимость сокращения времени, затрачиваемого многопроцессорными вычислительными системами (МВС) на обработку данных, необходимых для развертывания и поддержания в работоспособном состоянии конкретной орбитальной группировки. Иными словами, речь идет об увеличении скорости выполнения математических вычислений и, как следствие, повышении оперативности обработки информации в многопроцессорных системах.

Следует отметить, что переход к многопроцессорным системам требует одновременного перехода к параллельным вычислениям, поскольку задействовать вычислительный потенциал многоядерных процессоров можно только в том случае, если разделить выполняемые вычисления на информационно независимые блоки и организовать их выполнение на разных процессорах [4]. Подобный подход позволяет выполнять необходимые вычисления с меньшими затратами времени и тем самым получать максимальное ускорение процесса, которое ограничивается только числом имеющихся процессоров и количеством «независимых» блоков в выполняемых вычислениях [2].

Однако необходимость организации параллельных вычислений приводит к усложнению процесса эффективного применения многопроцессорных компьютерных систем. Это обстоятельство связано с тем, что для использования параллельных вычислений требуется в рассматриваемых алгоритмах (задачах) не только определять саму возможность распараллеливания, выделять независимые блоки и организовывать их выполнение, но и оценивать получаемый выигрыш во времени, если таковой имеется, по сравнению с последовательным решением поставленной задачи [4].

#### *Общий подход к распараллеливанию алгоритмов решаемых задач*

Одна из основных проблем параллельного программирования – выявление возможности распараллеливания существующих последовательных алгоритмов решаемых задач, а также непосредственно сам процесс распараллеливания алгоритмов, связанный с их дальнейшей программной реализацией.

Решение этой проблемы целесообразно проводить в несколько этапов.

1. Поскольку целью распараллеливания является повышение оперативности решения различных прикладных математических задач, то основополагающим является обобщение этих задач и отнесение их к соответствующим разделам математики для дальнейшего детального рассмотрения по этим разделам.

2. Последующий анализ задач, решаемых в рамках конкретного раздела математики, связанный с определением возможности их распараллеливания и формирования единой параллельно-последовательной задачи. Это позволяет увеличить оперативность решения всех рассматриваемых задач раздела.

3. Аналогичное рассмотрение остальных исходных последовательных алгоритмов решения задач по каждому из рассматриваемых разделов математики и составление их параллельно-последовательных алгоритмов для решения поставленных задач.

4. В заключение процесса преобразования алгоритмов происходит их программная реализация и оценка эффективности процедуры распараллеливания.

---

Влияние параллельных вычислений и структуры алгоритмов решаемых задач...

Анализ алгоритмов указанных выше задач показывает, что их условно можно разделить на три группы:

1) *полностью последовательные алгоритмы* – самостоятельные, независимые алгоритмы, которые невозможно распараллелить, но можно реализовывать программно на любом из процессоров многопроцессорной вычислительной системы (ВС) в первоначальном виде [3];

2) *смешанная группа алгоритмов*, структура которых может быть представлена в виде параллельно-последовательного графа (ПП-графа) и частично реализуется средствами параллельного программирования на многопроцессорных ВС [6];

3) *полностью параллельные алгоритмы* – группа алгоритмов, представляющих собой параллельную структуру, состоящую из независимых ветвей (потоков), одновременная программная реализация которых на многопроцессорных ВС вообще не вызывает затруднений [3].

*Содержательная постановка задачи*

На содержательном уровне постановку задачи можно сформулировать следующим образом.

1. Имеется ограниченный набор разнотипных по структуре алгоритмов решения специальных задач.

2. Требуется провести анализ структур алгоритмов, выявить необходимость и возможность их распараллеливания.

Не вдаваясь в подробности и тонкости работы с различными группами алгоритмов, для обобщения их анализа остановимся на самой большой и сложной группе – смешанной, позволяющей достаточно наглядно продемонстрировать весь процесс параллельно-последовательной обработки данной группы алгоритмов.

Как уже отмечалось, решение специальных задач обычно связано с обработкой больших объемов информации на различных этапах существования сложных технических систем – при их проектировании, создании и эксплуатации. Это в полной мере относится к техническим системам космического назначения. Создание указанных систем неразрывно связано с разработкой алгоритмов их функционирования и определением путей возможного распараллеливания вычислений, что, в свою очередь, влечет за собой увеличение скорости выполнения математических вычислений и, как следствие, повышение оперативности обработки информации в многопроцессорных ВС.

Рассмотрим в качестве примера анализа одного из указанных выше типов алгоритмов решение задачи дистанционного зондирования Земли из космоса (ДЗЗ) [8]. При этом съемка и сжатие изображений на борту космического аппарата (КА) должны осуществляться с автоматизированным выбором коэффициента сжатия изображений, что будет уменьшать временные затраты на сжатие графических изображений на борту и при передаче их на Землю [8]. При этом задача повышения качества сжатия графических изображений решается за счет выбора наиболее подходящего метода сжатия, а повышение скорости передачи графических изображений на Землю – за счет организации параллельной обработки информации на борту КА. Совместное решение этих двух задач в целом существенно влияет на повышение оперативности обработки информации в бортовых вычислительных системах (БВС) [5].

## Анализ данных и интеллектуальные системы

Анализ [1, 7] показал, что среди методов сжатия изображений целесообразно выбрать наиболее подходящий метод по качеству сжатия. Сжатие данных без потерь позволяет восстанавливать исходные несжатые данные с точностью до бита. Эта характеристика, как правило, необходима для текстовых файлов, баз данных, двоичных объектных файлов.

Цель любого алгоритма сжатия с потерями – достижение высоких коэффициентов сжатия за счет потери некоторых битов таким образом, чтобы восстановленные данные были достаточно близки к оригиналу.

Широко распространенным алгоритмом сжатия информации с потерями является алгоритм, основанный на методе дискретного косинусного преобразования (ДКП), двумерный вариант которого применяется для матриц элементов изображения размером  $8 \times 8$  пикселей [7].

Дискретное косинусное преобразование физически представляет собой преобразование массива пикселей изображения в массив значений пространственной частоты. Это преобразование обратимо с точностью до ошибок округления. Такой алгоритм сжатия изображений с потерями позволяет обеспечить существенно более высокую степень сжатия по сравнению с алгоритмами сжатия без потерь. Суть метода заключается в том, что при преобразовании выполняется вычисление косинусных коэффициентов [7].

Так, например, массив из  $N$  пикселей может быть представлен в виде взвешенной суммы  $N$  косинусных функций с различными частотами, а матрица из  $N \times N$  пикселей – взвешенной суммой  $N \times N$  косинусных функций [9].

Реализация предложенной схемы проверки выбранного алгоритма из группы смешанных алгоритмов для упомянутой задачи дистанционного зондирования Земли опирается на методику параллельного программирования, которая объединяет в себе перечисленные ниже аппаратно-программные компоненты [4]:

- многоядерные процессоры или микропроцессоры (например, четырехядерные компьютеры с процессором Intel Core 2 Quad Q8300 2,5 GGz);
- операционные системы (например, Windows, Linux, бортовые ОС);
- программы создания приложений – проектов ОС (например, Visual Studio);
- интерфейс параллельной программы (например, Open MP).

Технология проведения исследований параллельно-последовательного алгоритма сводится к составлению параллельной программы и серии нескольких вариантов компьютерных прогонов с помощью программного диспетчера задач, меняющего для каждого прогона число процессорных ядер компьютера и число обрабатываемых блоков данных.

В ходе реализации прогонов выполняются следующие действия.

#### *Прогон № 1*

Группа заданных снимков дистанционного зондирования Земли в виде последовательных блоков размером  $8 \times 8$  пикселей обрабатывается программно на одном из ядер многоядерного компьютера. Такой размер блока выбирается вследствие двух причин [8]. Во-первых, сложность вычислений быстро растет с увеличением размера обрабатываемого блока, поэтому блоки больших размеров оказывают чрезмерную нагрузку на вычислительные ресурсы. Во-вторых, исследования различных изображений показали, что ограничение обрабатываемой области этими размерами не приводит к потерям.

## Влияние параллельных вычислений и структуры алгоритмов решаемых задач...

В результате обработки программы происходит сжатие данных, а на ее входе и выходе с помощью процедуры *omp\_get\_wtime (void)* средств интерфейса Open MP фиксируется время выполнения.

*Прогон № 2*

Осуществляется распараллеливание группы указанных блоков на два потока, диспетчер задач программно распределяет потоки для выполнения их на двух любых свободных ядрах. Программа выполняется, происходит сжатие данных, снова фиксируется время выполнения программы.

*Прогон № 3*

Вся группа из четырех блоков данных распараллеливается на четыре потока по одному блоку и направляется для реализации на четыре свободных ядра компьютера. Программа выполняется, происходит сжатие данных, время выполнения программы фиксируется.

Результаты проведенного исследования всех компьютерных прогонов анализируются, сравниваются, затем делаются практические выводы.

Конкретные числовые данные и математические формулы представлены ниже.

*Математическая постановка задачи*

Пусть при дистанционном зондировании Земли на борту КА проводится оптико-электронная цифровая съемка земной поверхности. При обработке осуществляется сжатие данных, и на пункт сбора отправляются снимки отдельных участков Земли.

*Дано:*

1) группа из четырех блоков квадратных массивов данных  $V[n, k]$ , состоящих из 64 пикселей ( $8 \times 8$ ) одноцветного изображения каждый;

2) формула прямого двумерного дискретного косинусного преобразования [7]

$$T[i, j] = c(i, j) \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} V[n, k] \cos \frac{(2n+1)i\delta}{2N} \cos \frac{(2k+1)j\delta}{2N}, \quad i, j = 0, \dots, N,$$

где  $c(i, j) = \frac{1}{N}$ ,  $i$  или  $j = 0$ ;  $c(i, j) = \frac{2}{N}$ ,  $i \neq 0$ ,  $j \neq 0$ ;  $T[i, j]$  – массив данных в виде пикселей в сжатом состоянии;  $i, j, n, k$  – список целочисленных переменных для организации программной циклической обработки данных по формуле прямого двумерного дискретного косинусного преобразования;  $N$  – размер стороны квадратного массива данных в пикселях [ $8 \times 8$ ];  $V[n, k]$  – исходный массив данных, состоящий из 64 пикселей [ $8 \times 8$ ];

3) четырехъядерный компьютер;

4) параллельная программа прямого дискретного косинусного преобразования.

*Требуется:*

По алгоритму прямого двумерного дискретного косинусного преобразования программно реализовать прогоны заданных массивов данных на четырехъядерном компьютере.

Провести исследование зависимости повышения оперативности обработки информации от числа ядер компьютера и блоков данных, обрабатываемых параллельно-последовательно.

Организация прогонов программы в ходе исследования при разном соотношении состава блоков данных и количества задействованных процессорных ядер выявила следующие результаты.

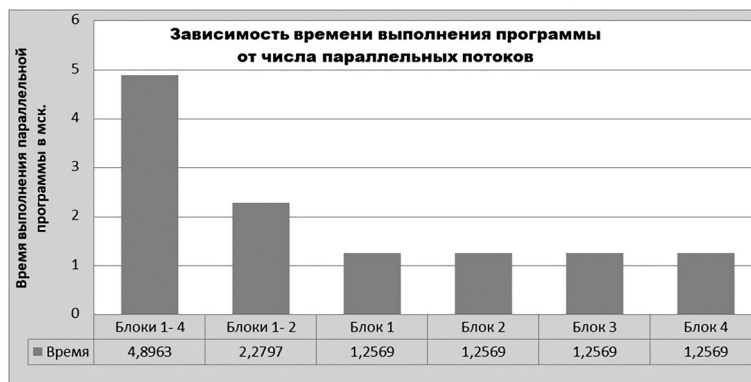
## Анализ данных и интеллектуальные системы

1. При первом прогоне программы по сжатию изображений на одном процессорном ядре компьютера было последовательно обработано четыре блока данных за 4,8963 мск.

2. При втором прогоне программы по сжатию изображений обработка проведена двумя параллельными потоками по два последовательных блока в каждом на двух процессорных ядрах компьютера за 2,2797 мск.

3. Третий прогон был проведен четырьмя параллельными потоками с участием четырех блоков данных на четырех процессорных ядрах компьютера за 1,2569 мск.

Результаты исследования приведены на рисунке.



Зависимость времени выполнения программы от числа ее параллельных блоков и количества ядер процессора

### Заключение

Итак, параллельные структуры алгоритмов решаемых задач и аппаратных средств вычислительной техники, а также параллельное программирование – это главные составляющие достижения необходимого параллелизма вычислительных процессов и, как следствие, роста производительности работы вычислительных средств и повышения оперативности обработки информации.

Целью проведенного исследования было показать зависимость времени решения задачи от структуры алгоритма и возможности ее распараллеливания с учетом технических возможностей многопроцессорных ВС.

Следует отметить, что алгоритмы задач, отнесенные к группе полностью параллельных алгоритмов, не вызывают затруднений при параллельном программировании, и реализация их на многоядерных вычислительных средствах ограничена только количеством задействованных процессоров (ядер). В этом случае повышение производительности многопроцессорных ВС возможно в основном за счет масштабирования (наращивания числа процессоров или ядер).

Когда имеет место последовательная или параллельно-последовательная обработка данных, повышение производительности многопроцессорных ВС возможно за счет трех составляющих процесса: 1) распараллеливания последовательных или параллельно-последовательных алгоритмов; 2) сжатия данных и 3) масштабирования (наращивания числа процессоров или ядер компьютера).

## Литература

1. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М.: Диалог-МИФИ, 2003. 384 с.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 584 с.
3. Вычислительные системы: практикум / сост.: А.Г. Басыров, А.С. Дудкин, И.В. Захаров, А.С. Швецов и др. СПб.: Изд-во Военной академии имени А.Ф. Можайского, 2016. 108 с.
4. Гергель В.П. Теория и практика параллельных вычислений. М.: БИНОМ. Лаборатория знаний, 2007. 464 с.
5. Захаров А.И., Лохвицкий В.А., Старобинец Д.Ю., Хомоненко А.Д. Оценка влияния параллельной обработки изображений на оперативность функционирования БКУ КА дистанционного зондирования Земли // Современные проблемы дистанционного зондирования Земли из космоса. 2019. Т. 16, № 1. С. 61–71.
6. Захаров А.И., Пореченский М.А., Чмыхова Я.В. Имитационная модель исследования влияния распараллеливания информационных процессов на рост производительности многоядерных вычислительных систем // Сборник алгоритмов и программ прикладных задач. СПб.: Изд-во Военной академии имени А.Ф. Можайского, 2017. Вып. 34. С. 173–181.
7. Хомоненко А.Д. Методы сжатия изображений: учеб. пособие. СПб.: Изд-во Петербургского государственного университета путей сообщения, 2009. 31 с.
8. Шовенгердт Р. Дистанционное зондирование. Методы и модели обработки изображений. М.: Техносфера, 2010. 560 с.
9. Loeffler C., Ligtenberg A., Moschytz G. Practical Fast 1-D DCT Algorithms with 11 Multiplications // Proc. Int'l. Conf. on Acoustics, Speech, and Signal Processing 1989 (ICASSP'89). Pp. 988–991.

## Literatura

1. Vatin D., Ratushnyak A., Smirnov M., Yukin V. Metody szhatiya dannykh. Ustrojstvo arkhivatorov, szhatie izobrazhenij i video. M.: Dialog-MIFI, 2003. 384 s.
2. Voevodin V.V., Voevodin V.I.V. Parallel'nye vychisleniya. SPb.: BKhV-Peterburg, 2002. 584 s.
3. Vychislitel'nye sistemy: praktikum / sost.: A.G. Basyrov, A.S. Dudkin, I.V. Zakharov, A.S. Shvetsov i dr. SPb.: Izd-vo Voennoj akademii imeni A.F. Mozhajskogo, 2016. 108 s.
4. Gergel' V.P. Teoriya i praktika parallel'nykh vychislenij. M.: BINOM. Laboratoriya znaniy, 2007. 464 s.
5. Zakharov A.I., Lokhvitskij V.A., Starobinets D.Yu., Khomonenko A.D. Otsenka vliyanija parallel'noj obrabotki izobrazhenij na operativnost' funktsionirovaniya BКУ КА distantsionnogo zondirovaniya Zemli // Sovremennye problemy distantsionnogo zondirovaniya Zemli iz kosmosa. 2019. T. 16, № 1. S. 61–71.
6. Zakharov A.I., Porechenskij M.A., Chmykhova Ya.V. Imitatsionnaya model' issledovaniya vliyanija rasparrallelivaniya informatsionnykh protsessov na rost proizvoditel'nosti mnogoyadernykh vychislitel'nykh sistem // Sbornik algoritmov i programm prikladnykh zadach. SPb.: Izd-vo Voennoj akademii imeni A.F. Mozhajskogo, 2017. Vyp. 34. S. 173–181.
7. Khomonenko A.D. Metody szhatiya izobrazhenij: ucheb. posobie. SPb.: Izd-vo Peterburgskogo gosudarstvennogo universiteta putej soobshcheniya, 2009. 31 s.
8. Shovengerdt R. Distantsionnoe zondirovanie. Metody i modeli obrabotki izobrazhenij. M.: Tekhnosfera, 2010. 560 s.
9. Loeffler C., Ligtenberg A., Moschytz G. Practical Fast 1-D DCT Algorithms with 11 Multiplications // Proc. Int'l. Conf. on Acoustics, Speech, and Signal Processing, 1989 (ICASSP'89). Pp. 988–991.