УДК 004.021

# Повышение производительности многопроцессорных вычислительных систем с гетерогенной архитектурой

Колпаков А. А., Кропотов Ю. А., Проскуряков А. Ю.

***Постановка задачи.*** *Вопрос создания высокопроизводительных вычислительных комплексов на базе компьютерных систем является актуальным, так как объемы обрабатываемой информации, вычислений и исследований с большими массивами данных постоянно увеличиваются. В связи с этим возникает задача разработки алгоритмов повышения производительности компьютерных систем на основе архитектуры использующих дополнительные вычислительные производительные модули или с однородные модули на графических процессорах.* ***Целью работы*** *является разработка алгоритма повышения производительности параллельных вычислений в многопроцессорных вычислительных системах с гетерогенной архитектурой.* ***Используемые методы:*** *метод декомпозиции задачи на этапы, метод принятия решений о переносе вычислений на графические процессоры.* ***Новизна.*** *Элементами новизны представленного решения является модифицированная PRAM-модель для применения графических процессоров.* ***Результат.*** *Разработан алгоритм повышения производительности параллельных вычислений в многопроцессорных вычислительных системах с гетерогенной архитектурой. Данный алгоритм использует применение графических процессоров в качестве специализированных вычислительных модулей в составе гетерогенной многопроцессорной вычислительной системы. Его применение приводит к существенному повышению производительности вычислений в зависимости от числа обрабатываемых потоков.* ***Практическая значимость****. Представленное решение предполагается реализовать в виде программного модуля для компьютерной системы с использованием технологии CUDA.*

***Ключевые слова:*** *параллельные вычисления, алгоритм повышения производительности вычислений, PRAM-модель, гетерогенные вычислительные системы, графические процессоры.*

## Введение

Известно, что повышение эффективности вычислительных компьютерных систем осуществляется в зависимости от организации процесса решения задач [1, 2]. В общем случае задачи представляются параллельными программами и описываются рядом параметров, в числе которых: количество ветвей, ранг необходимой подсистемы, время решения и т.п. Режим функционирования высокопроизводительных вычислительных систем формируется мультипрограммным методом или в некоторых вычислительных компьютерных системах используется частичное применение вычислительных модулей, что в недостаточной степени обеспечивает повышение производительности вычислений [3].

В связи с этим возникает задача разработки методов повышения производительности компьютерных систем на основе модели архитектуры с использованием дополнительных вычислительных производительных модулей или с использованием однородных модулей на графических процессорах. Основной задачей повышения производительности такой вычислительной системы является решение проблемы принятия решений о переносе операций вычислений на специализированные вычислительные модули и кэшировании данных, что требует исследований и разработки соответствующих алгоритмов [4].

## Архитектура гетерогенных многопроцессорных вычислительных систем

Для рассмотрения особенностей обобщенной архитектуры специализированных вычислительных модулей и их взаимодействия с центральным процессором была разработана и исследована структурная схема архитектуры гетерогенной многопроцессорной вычислительной системы, которая изображена на рис. 1. Базовыми структурными элементами специализированных вычислительных модулей являются спецпамять (SpRAM), в которой отдельно можно выделить память констант и глобальную память, и множество мультипроцессоров. Чтобы обработать данные на специализированных вычислительных модулях, необходимо передать их из оперативной памяти компьютера в SpRAM в соответствии со структурной схемой архитектуры гетерогенной системы на рис. 1.

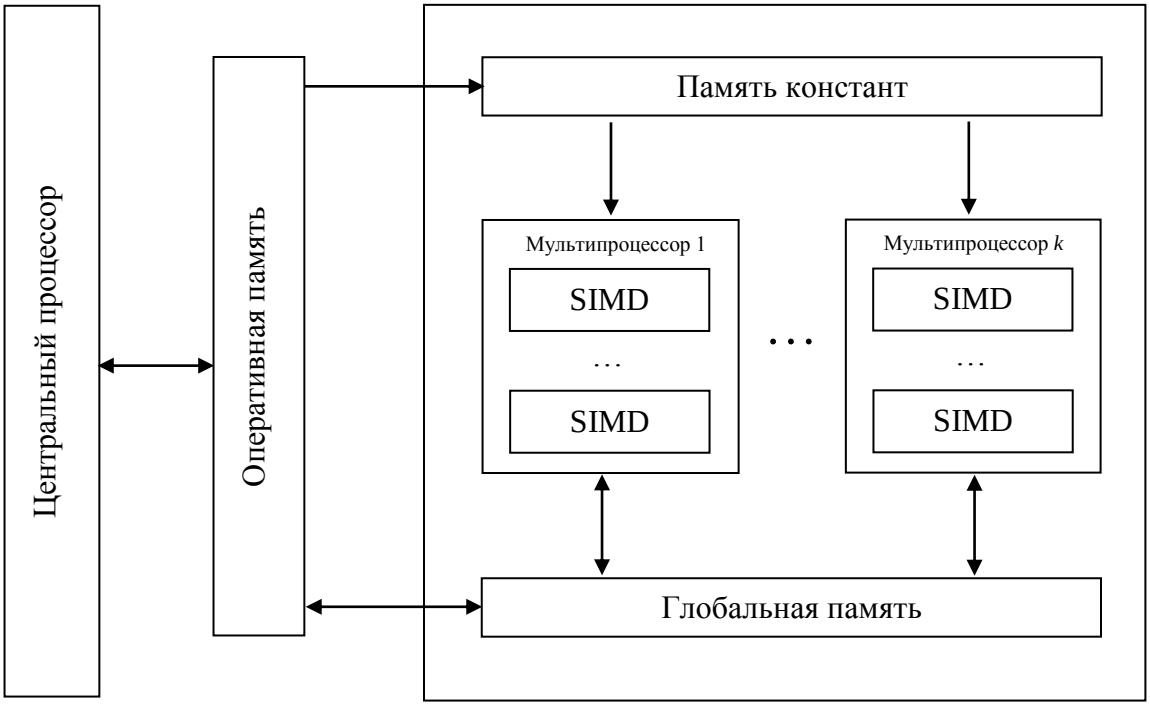


Рис. 1. Структурная схема архитектуры гетерогенной многопроцессорной вычислительной системы

Как видно из структурной схемы на рис. 1, связку «центральный процессор – графический процессор» можно отнести к модели с общей памятью. Основной моделью с общей памятью является модель PRAM (parallel random-access machine) – машина с параллельным произвольным доступом. Она является абстрактной идеализированной моделью параллельной синхронной машины с разделяемой общей памятью, которая использует допущения, приведенные ниже:

- количество процессоров ($q$) в машине не ограничено;
- каждый процессор имеет равнозначный доступ к любой ячейке общей памяти, размер которой не ограничен;
- отсутствует конкуренция по ресурсам;

– процессоры работают в режиме MIMD, но в частном случае может использоваться режим SIMD.

Все процессоры исполняют инструкции синхронно, причем выполнение любой инструкции занимает ровно 1 такт, называемый шагом PRAM-машины.

Чтобы оценить время выполнения алгоритма для $N$ входных данных на PRAM-машине с $p$ потоками, в работе [5] было получено выражение

$$T(N,p) = O\left(\frac{W(N)}{p} + S(N)\right), \quad (1)$$

где $O$ – верхняя асимптотическая оценка трудоёмкости алгоритма,

$N$ – количество входных данных алгоритма,

$S(N)$ – шаговая сложность алгоритма,

$W(N) = \sum_{i=1}^{S(n)} W_i(N)$ – рабочая сложность параллельного алгоритма, где $W_i(N)$ – количество параллельных операций на шаге $i$.

Формула (1) дает верхнюю асимптотическую оценку времени исполнения алгоритма с шаговой сложностью $S(N)$ и рабочей сложностью $W(N)$.

Из схемы, приведенной на рис. 1, можно отметить, что PRAM модель может быть применена к многопроцессорной системе с учётом следующих уточнений и дополнений:

1) все процессоры могут одновременно считывать данные из разделяемой памяти, но запись должна быть монопольной, т.к. порядок изменения ячейки разделяемой памяти при обращении на запись из нескольких скалярных процессоров не определён (PRAM – CREW (Concurrent Read, Exclusive Write));
2) количество скалярных процессоров в графическом мультипроцессоре ограничено сверху ($q_{max}$ процессоров). Для выполнения большего числа потоков используется система горизонтального параллелизма, аналогичная горизонтальной структуре в модели BSP: генерируется расписание последовательного исполнения потоков, разбитых на пучки по $q_{warp}$ скалярных процессоров;
3) размер разделяемой памяти мультипроцессора ограничен – $M_s$ байт;
4) все скалярные процессоры работают с одинаковой скоростью по принципу SIMD со скоростью $S_{GPU}$ элементарных операций в секунду;
5) должна иметь место дополнительная операция – обращение к оперативной памяти SpRAM специализированного вычислительного модуля на чтение или запись. Задержка при обращении $K$ определяется количеством элементарных операций, требуемых при обращении к одному числу одинарной точности в глобальной памяти специализированного вычислительного модуля.

Таким образом, PRAM модель с перечисленными уточнениями и дополнениями допускает применение графических процессоров в качестве специализированных вычислительных модулей для общих вычислений.

## Общий алгоритм оптимизации параллельных вычислений в многопроцессорных вычислительных системах с гетерогенной архитектурой

Для организации параллельных вычислений в многопроцессорных вычислительных системах с гетерогенной архитектурой «CPU – SCM», был разработан общий алгоритм оптимизации, который приведен на рис. 2. В качестве специализированного вычислительного модуля используется графический процессор GPU.

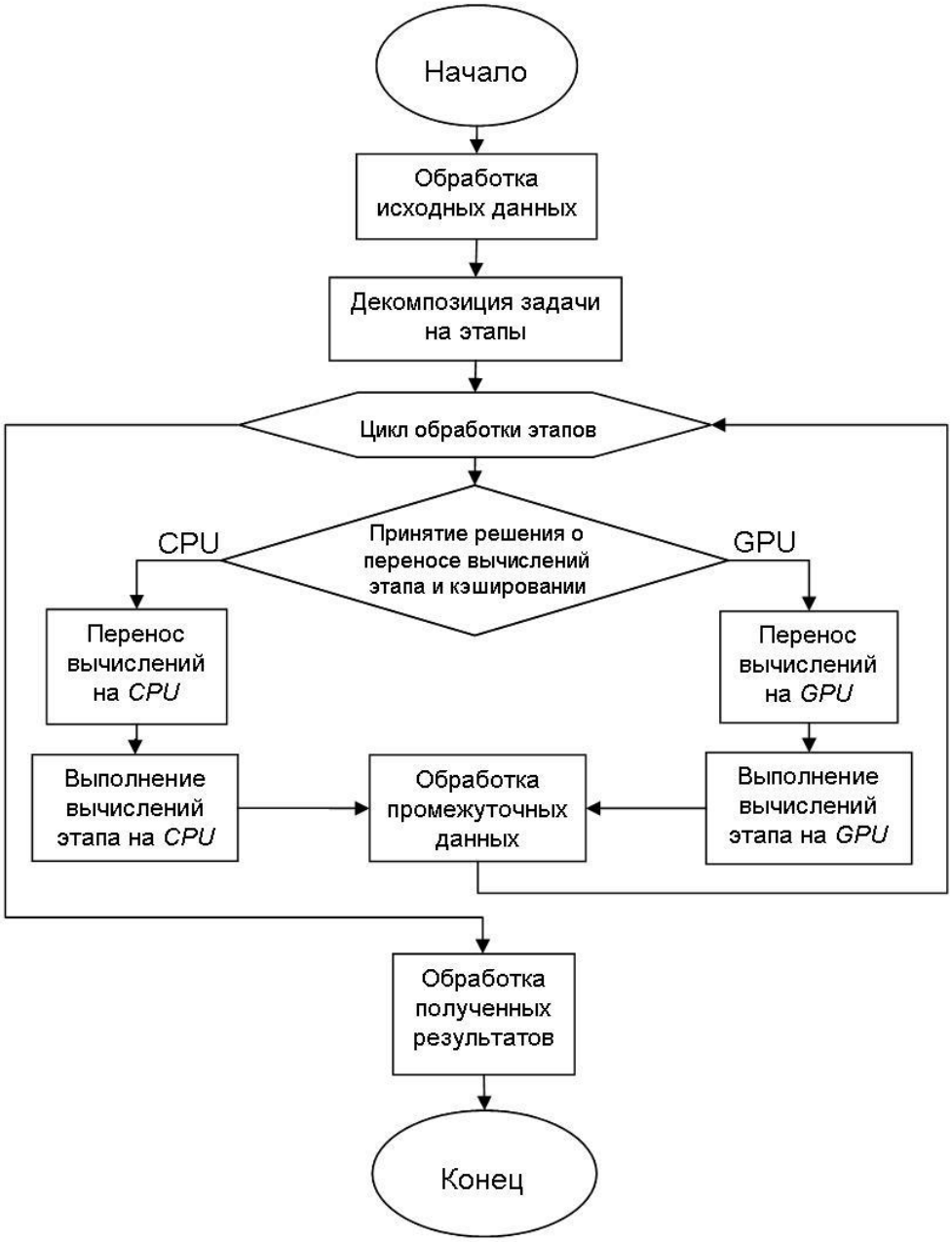


Рис. 2. Алгоритм повышения производительности параллельных вычислений в многопроцессорных вычислительных системах с гетерогенной архитектурой

Основным этапом разработанного алгоритма является блок принятия решения о переносе вычислений этапа на графический процессор. Для осуществления сравнения производительности этапа алгоритма на различных

вычислительных устройствах и последующего принятия решения о переносе вычислений используется модифицированная PRAM-модель.

## Модифицированная PRAM-модель

В соответствии с моделью специализированного мультипроцессора, которая является общей для всех моделей графических процессоров, сформирован абстрактный вычислительный мультипроцессор. Для абстрактного вычислительного мультипроцессора имеем следующее множество параметров $\{q_{max}, q_{warp}, M_s, S_{GPU}, K\}$, учитывающих основные характеристики реальных специализированных мультипроцессоров. Для разработки параллельного алгоритма под предложенную модель можно воспользоваться методом создания расписания распределения потоков вычислений, который применяется в базовой PRAM модели, учитывая изложенные выше уточнения и дополнения. В этом случае формулу (1) для верхней оценки времени исполнения алгоритма на PRAM машине следует скорректировать. PRAM модель теперь должна быть представлена в виде одного абстрактного вычислительного мультипроцессора, на котором все скалярные процессоры работают пучками по принципу горизонтального параллелизма. Выражение вычисления верхней оценки временной сложности алгоритма принимает вид

$$T_C(N, p) = O\left(\frac{W(N)}{p} \cdot \left\lceil \frac{p}{q_{warp}} \right\rceil + S(N)\right), \quad (2)$$

где $p$ – число потоков алгоритма, предназначенных для обработки $N$ элементов данных, $p<q_{max}$.

Основным объектом исследования является учет операций обращения к глобальной памяти графического процессора. Необходимо ввести дополнительный параметр алгоритма – сложность обращения к глобальной памяти $R(N)$, которой является суммарное количество обращений на чтение и запись из глобальной памяти графического процессора, требуемое для обработки $N$ элементов данных. Данный вид операций должен присутствовать в любом параллельном алгоритме для графических процессоров, который обрабатывает входные данные. Вследствие того, что процессоры работают в режиме SIMD и выполняют команды последовательно пучками по принципу горизонтального параллелизма, то формула верхней оценки времени исполнения параллельного алгоритма на одном абстрактном вычислительном мультипроцессоре принимает вид

$$T_C^{GPU}(N, p) = O\left(\frac{W(N) + R(N)}{p} \cdot \left\lceil \frac{p}{q_{warp}} \right\rceil + S(N)\right). \quad (3)$$

Исходя из выражения (3), более высокая производительность будет у того алгоритма, который будет иметь меньшее количество обращений к SpRAM. Тогда выражение для определения верхней оценки времени исполнения алгоритма на одном абстрактном вычислительном мультипроцессоре имеет вид

$$T_M(N, p) = \frac{W_M(N) + R_M(N) \cdot K}{S_{GPU} \cdot p} \cdot \left\lceil \frac{p}{q_{warp}} \right\rceil, \quad (4)$$

где $W_M(N)$ – количество элементарных операций одного процессора абстрактного вычислительного мультипроцессора в PRAM,

$R_M(N)$ – количество обращений к SpRAM из одного процессора абстрактного вычислительного мультипроцессора в PRAM.

На основании формулы (3) можно записать выражение для верхней оценки времени исполнения этапа алгоритма

$$T_G(N) = \left\lceil \frac{p}{q_{warp}} \right\rceil \cdot T_M(M, p), \tag{5}$$

Для учета передачи данных между оперативной памяти и памятью SpRAM, следует ввести ещё два дополнительных параметра: суммарное количество входных данных этапа алгоритма в байтах $N_{HD}$ и суммарное количество выходных данных этапа алгоритма в байтах $N_{DH}$. Тогда выражение вычисления общего времени работы этапа алгоритма принимает вид

$$T_{GPU}(N) = \frac{N_{iHD}(N)}{S_{HD}} + T_G(N) + \frac{N_{iDH}(N)}{S_{DH}}, \tag{6}$$

где $S_{HD}$ и $S_{DH}$ – константы скорости передачи данных между RAM и SpRAM (байт/с).

На основе полученной модели показано, что для анализа и сравнения параллельных алгоритмов необходимо использовать следующие параметры алгоритма:

1) суммарная шаговая сложность $S(N)$

$$S(N) = \sum_{i=1}^{B(N)} S_i(N); \tag{7}$$

2) суммарная рабочая сложность $W(N)$

$$W(N) = \sum_{i=1}^{B(N)} W_i(N); \tag{8}$$

3) суммарная сложность обращения к глобальной памяти специализированного вычислительного модуля $R(N)$

$$R(N) = \sum_{i=1}^{B(N)} R_i(N); \tag{9}$$

4) суммарный объём данных, передаваемых между оперативной памятью вычислительной компьютерной системы и глобальной памятью специализированного вычислительного модуля $N_{HD}$ и $N_{DH}$

$$\begin{aligned} N_{HD}(N) &= \sum_{i=1}^{B(N)} N_{i\ HD}(N) \\ N_{DH}(N) &= \sum_{i=1}^{B(N)} N_{i\ DH}(N) \end{aligned}. \tag{10}$$

С учетом выражений (7)-(10) верхняя оценка времени работы алгоритма на графическом процессоре в среде CPU-GPU вычисляется следующим выражением

$$T_{GPU}(N) = \frac{N_{HD}(N)}{S_{HD}} + \sum_{i=1}^{B(N)} T_{i\,G}(N) + \frac{N_{DH}(N)}{S_{DH}}. \quad (11)$$

При принятии решения о переносе вычислений на GPU, предварительно производится оценка времени выполнения алгоритма на CPU в соответствие с (2) и оценка времени выполнения алгоритма на GPU в соответствие с (10). После этого осуществляется сравнение полученных временных показателей и по результату принимается решение о переносе вычислений.

## Экспериментальное исследование разработанного алгоритма

В качестве тестовой задачи, использовалась задача нахождения нулевых битовых векторов, которая решается с применением генетических алгоритмов [5]. При решении указанной задачи основное время работы занимают параллельные вычисления значений функций приспособленности различных особей, операций скрещивания и мутации. Используемый алгоритм ее решения имеет свойства, характерные для многих генетических алгоритмов:

1) представление особи в виде битовой строки;
2) малое число логических операций при вычислении функции приспособленности, выполнении мутации и скрещивания;
3) последовательный доступ к памяти.

Данные свойства позволяют эффективно использовать вычисления на графическом процессоре.

Для проведения экспериментальной оценки эффективности работы алгоритма оптимизации [6, 7] использовалась тестовая компьютерная система следующей конфигурации: центральный процессор Intel Core 2 Quad Q9400 (2.66GHz), ОЗУ 8 Гбайт, графическая карта Nvidia GeForce GTX560 2 Гбайт 336 потоков, операционная система Windows 7 x64**,** компилятор MS Visual Studio 2008 в release режиме.

При исследовании производительности тестовой задачей изменялось количество 32-битных целых чисел в массиве ($M$) и число параллельных потоков ($N$) [8, 9].

Исследовалось среднее время $t$, потраченное на получение нового поколения для различного количества 32-битных целых чисел в массиве и числа параллельных потоков. Исследования проводились с использованием технологий OpenCL и NVIDIA CUDA.

Результаты экспериментальных исследований приведены в таблице 1.

Таблица 1 – Время генерации многопроцессорной системой одного поколения, $N$ =10, значения приведены в мс

| Процессор | Кол-во особей в поколении | | | | |
|---|---|---|---|---|---|
| | 128 | 1024 | 10240 | 102400 | 1024000 |
| CPU - Q9400 | 0,38 | 0,56 | 2,5 | 22,2 | 416,2 |
| CUDA GPU - GTX460 | 0,08 | 0,14 | 1,03 | 13 | 237,4 |

Как видно из результатов экспериментального исследования, применение разработанного алгоритма оптимизации дает рост производительности относительно центрального процессора – в случае применения NVIDIA CUDA время обработки сокращается с 0,38 мс до 0,08 мс для 128 потоков и с 416,2 мс до 237,4 мс для 102400 потоков [10, 11].

## Заключение

Таким образом, на основе модифицированной PRAM-модели, разработан алгоритм повышения производительности параллельных вычислений на специализированных вычислительных модулях, который включает в себя алгоритм принятия решения о переносе вычислений на графический процессор.

Методом оценивания производительности были осуществлены сравнительные экспериментальные исследования разработанного алгоритма. Результаты оценивания алгоритма показывают повышение производительности не менее, чем в 2-4 раза в зависимости от числа исследуемых потоков.

## Литература

1. Современные проблемы вычислительной математики и математического моделирования. Т. 1: Вычислительная математика / Под ред. Бахвалова Н. С., Воеводина В. В. – М.: Наука, 2005. – 342 с.

2. Graham R. L. Bounds on Multiprocessing Timing Anomalies // SIAM Journal on Applied Mathematics. 1969. Vol. 17. No. 2. C. 416-429.

3. Колпаков А. А. Аспекты оценки увеличения производительности вычислений при распараллеливании процессоров вычислительных систем // Методы и устройства передачи и обработки информации. 2011. № 1 (13). С. 124-127.

4. Колпаков А. А. Теоретическая оценка роста производительности вычислительной системы при использовании нескольких вычислительных устройств // В мире научных открытий. 2012. № 1. С. 206-209.

5. Капустин Д. С. Ржеуцкая С. Ю. Модификация абстрактной модели параллельных вычислений PRAM с учетом существенных особенностей графических процессоров // Естественные и технические науки. 2011. № 5 (55). С. 336-342.

5. Колпаков А. А. Оптимизация генетических алгоритмов при использовании вычислений на графических процессорах на примере задачи нулевых битовых векторов // Информационные системы и технологии. 2013. № 2 (76). С. 22-28.

6. Кропотов Ю. А. Экспериментальные исследования закона распределения вероятности амплитуд сигналов систем передачи речевой информации // Проектирование и технология электронных средств. 2006. Т. 4. С. 37-42.

7. Кропотов Ю. А., Быков А. А. Алгоритм подавления акустических шумов и сосредоточенных помех с формантным распределением полос режекции // Вопросы радиоэлектроники. 2010. Т. 1 № 1. С. 60-65.

8. Кропотов Ю. А. Временной интервал определения закона распределения вероятности амплитуд речевого сигнала // Радиотехника. 2006. № 6. С. 97-98.

9. Ермолаев В. А., Кропотов Ю. А. О корреляционном оценивании параметров моделей акустических эхо-сигналов // Вопросы радиоэлектроники, 2010. Т. 1 № 1. С. 46-50.

10. Кропотов Ю. А., Проскуряков А. Ю., Белов А. А., Колпаков А. А. Модели, алгоритмы системы автоматизированного мониторинга и управления экологической безопасности промышленных производств // Системы управления, связи и безопасности. 2015. № 2. С. 184-197.

11. Кропотов Ю. А., Белов А. А., Проскуряков А. Ю., Колпаков А. А. Методы проектирования телекоммуникационных информационно–управляющих систем аудиообмена в сложной помеховой обстановке // Системы управления, связи и безопасности. 2015. № 2. С. 165-183.

## References

1. *Sovremennye problemy vychislitelnoj matematiki i matematicheskogo modelirovaniya. vol. 1, Vychislitelnaya matematika*. [Modern Problems of Computational Mathematics and Mathematical Modelling. Vol. 1, Computational Mathematics]. Moscow, Science, 2005. 342 p. (in Russian).

2. Graham R. L. Bounds on Multiprocessing Timing Anomalies. *SIAM Journal on Applied Mathematics*, 1969, vol. 17, no. 2, pp. 416-429.

3. Kolpakov A. A. Kropotov Y. A. Aspects of the assessment increase performance of computations in parallel processors of the computing system. *Metody i ustroistva peredachi i obrabotki informatsii*, 2011, vol 13, no. 1, pp 124-127 (in Russian).

4. Kolpakov A. A. Theoretical evaluation of growth performance computing systems from the use of multiple computing devices. *V mire nauchnykh otkrytii*, 2012, no. 1, pp. 206-209 (in Russian).

5. Kolpakov A. A. Optimizing the use of genetic algorithms for computing graphics processors for the problem of zero bit vector *Informatsionnye sistemy i tekhnologii*, 2013, vol. 76, no. 2, pp. 22-28 (in Russian).

6. Kropotov Y. A. Experimental study of the law of distribution of probability of amplitudes of signals of systems of transmission of voice information *Proektirovanie i tekhnologiia elektronnykh sredstv*, 2006, vol. 4, pp. 37-42 (in Russian).

7. Kropotov Y. A. Bykov A. A. Algorithm for suppression of acoustic noise and concentrated interference with the distribution of the formant bands of rejection *Voprosy radiojelektroniki*, 2010, vol. 1, no. 1, pp. 60-65 (in Russian).

8. Kropotov Y. A. The Time Interval of a Definition the Regularity Distribution Probability Amplitudes of Speech Signals. *Radiotehnika*, 2006, no. 6, pp. 97-98 (in Russian).

9. Ermolaev V. A., Kropotov Y. A. On the correlation estimation of parameters of models of acoustic echo-signals *Voprosy radiojelektroniki*, 2010, vol. 1, no. 1, pp. 46-50 (in Russian).

10. Kropotov Y. A., Proskuryakov A. Y., Belov A. A., Kolpakov A. A. Models, Algorithms System of Automated Monitoring and Management of Ecological Safety Industrial Plants. *Systems of Control, Communication and Security*, 2015, no. 2, pp. 184-197. Available at: http://journals.intelgr.com/sccs/archive/2015-02/08-Kropotov.pdf (accessed 24 September 2016) (in Russian).

11. Kropotov Y. A., Belov A. A., Proskuryakov A. Y., Kolpakov A. A. Methods of Designing Telecommunication Information and Control Audio Exchange Systems in Difficult Noise Conditions. *Systems of Control, Communication and Security*, 2015, no. 2, pp. 165-183. Available at: http://journals.intelgr.com/sccs/archive/2015-02/07-Kropotov.pdf (accessed 24 September 2016) (in Russian).

**Статья поступила 7 сентября 2016 г.**

## Информация об авторах

*Колпаков Александр Анатольевич* – кандидат технических наук. Доцент кафедры «Электроники и вычислительной техники». Муромский институт (филиал) «Владимирского государственного университета имени Александра Григорьевича и Николая Григорьевич Столетовых». Область научных интересов: параллельные и распределенные вычислительные системы. Тел.: +7 492 347 72 72. E-mail: kaf-eivt@yandex.ru

*Кропотов Юрий Анатольевич* – доктор технических наук, профессор. Зав. кафедрой «Электроники и вычислительной техники». Муромский институт (филиал) «Владимирского государственного университета имени Александра Григорьевича и Николая Григорьевич Столетовых». Область научных интересов: телекоммуникационные информационно-управляющие системы. Тел.: +7 492 347 72 72. E-mail: kaf-eivt@yandex.ru

*Проскуряков Александр Юрьевич* – кандидат технических наук. Доцент кафедры «Электроники и вычислительной техники». Муромский институт (филиал) ФГБОУ ВПО «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевич Столетовых». Область научных интересов: телекоммуникационные системы мониторинга и прогнозирования, обработка информации. Тел.: +7 492 347 72 72. E-mail: kaf-eivt@yandex.ru

Адрес: Россия, 602264, г. Муром, ул. Орловская, д. 23.

---

# Improving the Performance of Multiprocessor Computer Systems with Heterogeneous Architecture

A. A. Kolpakov, Y. A. Kropotov, A. Y. Proskuryakov

***Purpose.*** *The task of creating a high-performance computing systems based on computer systems is important because the volume of processed information is constantly increasing. This raises the task of developing algorithms to improve the performance of computer systems. The high performance ensured by architectures with additional computational modules or with homogeneous modules on GPUs. The paper*

*had offered to develop the algorithm for improving performance of parallel computation in multiprocessor computing systems with heterogeneous architecture.* ***The purpose of paper*** *is modification of the PRAM model for the application of graphical processors.* ***Methods.*** *A method of decomposition of the task into stages, the method of making decisions about the transfer calculations on accelerating the processors are used in paper.* ***Novelty.*** *The new PRAM-model takes into account GPUs. Result. The algorithm for increase of performance of parallel computations in multiprocessor computing systems with heterogeneous architecture is developed in paper. This algorithm based on application of graphical processors as specialized computational modules in the heterogeneous multiprocessor computer system. Its use increased productivity not less than 2-4 times depending on the number of streams under study.* ***Practical relevance.*** *The algorithm and the new PRAM-model can be implemented as a software solution for computer system with the CUDA technology.*

***Key words:*** *parallel computing, algorithm of improving computing performance, PRAM-model, heterogeneous computing systems, graphics processors.*

## Information about Authors

*Alexsandr Anatolievich Kolpakov* – Ph.D. of Engineering Sciences. Associate Professor at the Department of Electronics and Computer Science. Murom Institute (branch) of the «Vladimir State University named after Alexander and Nickolay Stoletovs». Field of research: parallel and distributed computing systems. Ph.: +7 492 347 72 72. E-mail: kaf-eivt@yandex.ru

*Yurij Anatolievich Kropotov* – Dr. habil. of Engineering Sciences, professor, Head of Chair «Electronics and Computer Science». Murom Institute (branch) of the «Vladimir State University named after Alexander and Nickolay Stoletovs». Field of research: telecommunication information and control systems. Ph.: +7 492 347 72 72. E-mail: kaf-eivt@yandex.ru

*Alexander Jurievich Proskuryakov* – Ph.D. of Engineering Sciences, Associate Professor at the Department of Electronics and Computer Science. Murom institute (branch) of the «Vladimir State University named after Alexander and Nickolay Stoletovs». Field of research: telecommunications monitoring and forecasting system, information processing. Ph.: +7 492 347 72 72. E-mail: kaf-eivt@yandex.ru

Address: Russia, 602264, Murom, st. Orlovskaya, h. 23.