



DOI: <http://dx.doi.org/10.15688/jvolsu1.2015.4.3>

УДК 004.94, 519.63

ББК 22.193

## ТРЕХМЕРНЫЙ ПАРАЛЛЕЛЬНЫЙ ЧИСЛЕННЫЙ ГАЗОДИНАМИЧЕСКИЙ КОД НА ОСНОВЕ СМЕШАННОГО ЛАГРАНЖЕВО-ЭЙЛЕРОВА ПОДХОДА<sup>1</sup>

**Татьяна Сергеевна Шушкевич**

Студентка Института математики и информационных технологий,  
Волгоградский государственный университет  
infomod@volsu.ru  
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

**Николай Михайлович Кузьмин**

Кандидат физико-математических наук, доцент кафедры информационных систем и  
компьютерного моделирования,  
Волгоградский государственный университет  
nmkuzmin@gmail.com, infomod@volsu.ru  
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

**Мария Анатольевна Бутенко**

Старший преподаватель кафедры информационных систем  
и компьютерного моделирования,  
Волгоградский государственный университет  
butenkoma@gmail.com, infomod@volsu.ru  
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

**Аннотация.** В данной работе описан созданный на основе смешанного лагранжево-эйлерова подхода параллельный численный код для интегрирования уравнений газодинамики в трехмерном случае. Представлена методика распараллеливания, позволяющая эффективно использовать вычислительные ресурсы компьютерного кластера, состоящего из многопроцессорных узлов. Приведены результаты тестирования производительности, показавшие существенный рост скорости вычислений по сравнению с последовательной версией программы.

**Ключевые слова:** численное моделирование, газодинамика, лагранжево-эйлеров подход, параллельные технологии, высокопроизводительные вычисления.

## Введение

Изучение многих природных и технических процессов связано с решением начально-краевых задач для системы уравнений газодинамики. При этом практически единственным способом получить решение является применение различных вычислительных алгоритмов. Одним из наиболее распространенных подходов являются сеточные методы. Однако численное решение трехмерных задач на подробных сетках требует большого объема памяти и высокой скорости вычислений.

Современные высокопроизводительные вычислительные машины слишком дороги и недоступны большинству исследователей. В то же время, на рынке представлено множество моделей персональных ЭВМ, производительность которых сравнительно невелика, однако, благодаря массовому производству, соотношение цена/производительность выгодно отличает их от специализированных вычислительных систем. Кроме того, на данный момент существуют широко доступные технологии передачи данных, позволяющие создавать сети персональных ЭВМ, отличающиеся высокой скоростью и малыми задержками прохождения данных.

Это позволяет создавать так называемые «кластеры» — наборы сравнительно дешевых серийных ЭВМ, соединенных высокопроизводительной сетью передачи данных и работающих под управлением специализированного программного обеспечения. Подобные системы характеризуются достаточно высокой производительностью для многих задач, при этом оставаясь доступными по стоимости для многих пользователей.

Поэтому создание параллельной программы для численного решения трехмерных газодинамических задач современными методами представляется актуальным.

### 1. Основные уравнения и численный метод

Динамику невязкого нетеплопроводного газа в отсутствии внешних сил можно описать с помощью уравнений

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \mathbf{0}, \quad (1)$$

где

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e + p)u \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (e + p)v \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (e + p)w \end{pmatrix}.$$

Здесь используются стандартные обозначения:  $x, y, z$  — декартовы координаты;  $t$  — время;  $\rho$  — плотность;  $\mathbf{V} = (u, v, w)$  — вектор скорости;  $p$  — давление;  $e$  — объемная плотность энергии. Система уравнений (1) замыкается уравнением на объемную плотность энергии:

$$e = \frac{p}{\gamma - 1} + \frac{\rho \mathbf{V}^2}{2}, \quad (2)$$

где  $\gamma$  — показатель адиабаты.

Для решения системы (1) был выбран метод cSPH—TVD, изначально предложенный в работе [3] для системы уравнений Сен-Венана, описывающих динамику поверхностных вод в приближении мелкой воды, и в дальнейшем обобщенный на случай полной системы уравнений невязкой газодинамики для идеального газа в отсутствии внешних сил [1]. В работе [4] была исследована точность этого метода в зависимости от вида функции ограничителя наклонов и способов приближенного решения задачи Римана. Метод основан на совместном использовании лагранжева (SPH) и эйлерова (TVD) подходов. Он обладает такими свойствами, как консервативность, второй порядок аппроксимации по временной и пространственным координатам, хорошая сбалансированность, сквозной устойчивый расчет динамической границы «газ — вакуум».

## 2. Технологии распараллеливания

Параллельные вычисления — это такая организация вычислений, при которой программы представляют собой совокупность одновременно выполняемых и взаимодействующих процессов. В настоящее время существует две основные архитектуры параллельных вычислительных систем:

- параллельные вычислительные системы с общей памятью или симметричные многопроцессорные системы (symmetric multiprocessing — SMP); они состоят из нескольких однородных серийно выпускаемых микропроцессоров, каждый из которых имеет свою кэш-память, и массива общей памяти RAM, подключение к которой производится с помощью общей шины;
- параллельные вычислительные системы с распределенной памятью или массивно-параллельные системы (massive parallel processing — MPP); они состоят из нескольких вычислительных узлов, каждый из которых содержит процессор, имеющий собственную оперативную и кэш-память, а для обмена информацией между узлами используется высокопроизводительная сеть передачи данных.

При создании параллельных программ, предназначенных для SMP-систем, обычно используется технология OpenMP. Она позволяет реализовать параллельные вычисления с помощью многопоточности, в которой «главный» поток создает набор «подчиненных», и задача распределяется между ними. Предполагается, что потоки выполняются параллельно на машине с несколькими процессорами, причем количество процессоров не обязательно должно быть больше или равно количеству потоков.

Массивно-параллельные системы в основном программируются с использованием технологии MPI. Интерфейс передачи сообщений MPI представляет собой набор из около 300 функций управления параллельными вычислениями на основе механизма обмена данными по сети.

С одной стороны, современные компьютеры имеют многоядерные процессоры, то есть они являются SMP-системами. С другой стороны, компьютерный вычислительный кластер является MPP-системой. Таким образом, мы получаем MPP-систему, состоящую из SMP-узлов. Поэтому перспективным представляется применение гибридного подхода к распараллеливанию: применение MPI для обмена данными между узлами кластера, каждый из которых выполняет распараллеленный при помощи OpenMP код.

### 3. Архитектура вычислительного кода

Существует несколько общепринятых алгоритмов организации работы параллельных программ. Поскольку сеточные методы предполагают наличие однотипных вычислений для большого набора элементов, то наиболее подходящей является методика SIMD (Single Instruction – Multiple Data). Она предполагает, что все параллельные процессы используют один и тот же программный код, обрабатывая различные участки общего массива данных.

С учетом вышенаписанного, применение методики SIMD для связки подходов MPI+OpenMP следует понимать в широком смысле, который иллюстрирует рисунок 1. На нем представлена примерная схема работы кода, состоящего из четырех процессов P0, P1, P2, P3, каждый из которых выполняется на отдельном узле кластера (технология MPI). Сначала процессы абсолютно идентичны и выполняют одинаковый код. Затем в рамках каждого отдельного узла кластера происходит разделение процессов на потоки, которые выполняются так же параллельно (технология OpenMP). В дальнейшем процессы взаимодействуют друг с другом, обмениваются необходимой информацией. Для этого нужны точки синхронизации (C0, C1 и так далее).

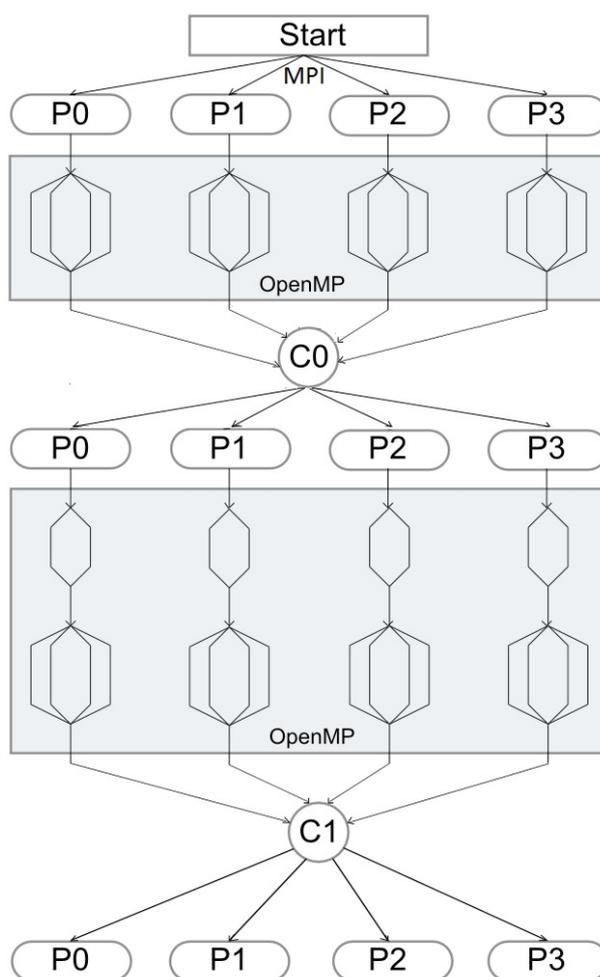


Рис. 1. Схема работы параллельной программы

В модели SIMD можно разрабатывать параллельные вычислительные программы,

использующие разбиение вычислительной сетки на подобласти. Именно такой тип распараллеливания и был выбран для создания численного кода.

Для адаптации вычислительной схемы к архитектуре кластера необходимо разбить вычислительную сетку на подобласти. При этом каждая область обрабатывается одним узлом вычислительной системы, а взаимодействие процессов требуется только при переходе к следующему временному шагу. В используемой нами численной схеме для вычисления значений переменных в ячейке на следующем временном шаге необходимо знать их значения в соседних ячейках. При разбиении обрабатываемого объема будут возникать ячейки, для вычисления параметров в которых необходимо знать значения параметров в ячейках, лежащих за пределами их подобласти. То есть для получения правильного решения процессам, обрабатывающим соседние подобласти, придется обмениваться информацией.

На рисунке 2 представлены два процесса, обрабатывающие смежные области. Фактически каждую область обрабатывает отдельный вычислительный код, не содержащий модификаций по сравнению с однопроцессорным вариантом.

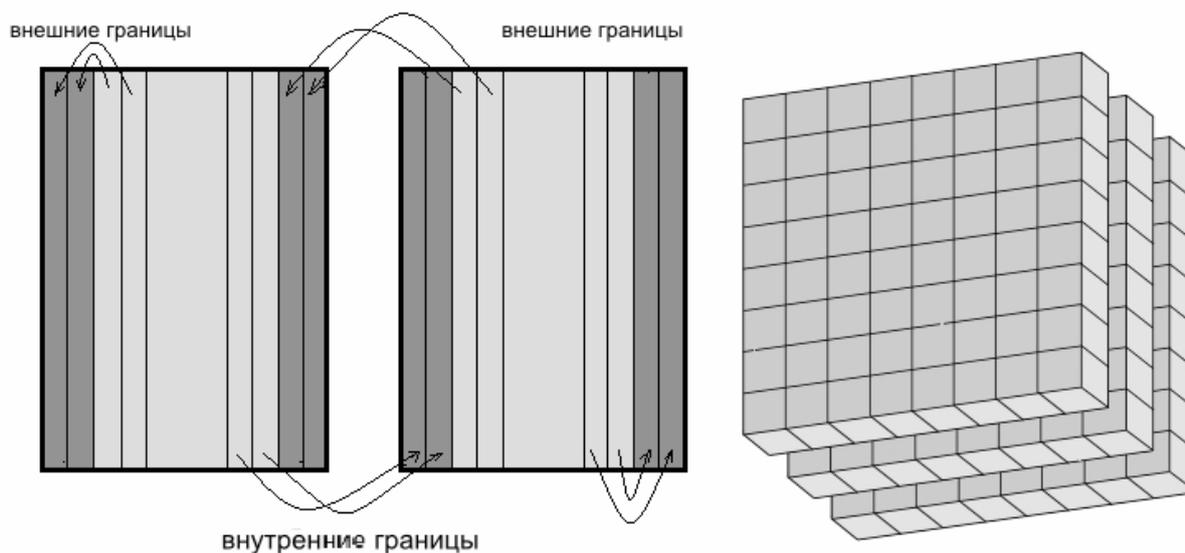


Рис. 2. Одномерное разбиение

Для наложения граничных условий вводятся дополнительные, «фиктивные», ячейки (в данном случае их по две слева и справа), которые формально находятся за границей области и не обрабатываются кодом. Величины в этих ячейках задаются постоянными значениями величин соседних, «реальных» ячеек.

При разбиении возникают еще и внутренние границы, которые разделяют подобласти. Для того чтобы сумма решений в двух смежных подобластях полностью совпадала с решением непараллельного варианта кода, достаточно правильно задать внутренние граничные условия. То есть в качестве внешних ячеек необходимо брать ячейки соседнего процесса, непосредственно прилегающие к данной границе с другой стороны (на рисунке 2 области таких ячеек обозначены темно-серым цветом).

Кроме того, для обеспечения правильного «сшивания» решения на границах, все процессы должны использовать одинаковое значение для временного шага  $dt$  [2]. Это можно сделать следующим образом: каждый процесс вычисляет собственное значение

шага по времени, после чего процессы выбирают наименьший из них и используют его на следующем шаге. Простую вычислительную сетку  $L \times M \times N$  можно разделить на подобласти множеством способов. Простой и эффективный способ — разбиение вдоль одной оси (рис. 2).

Упрощенная UML-диаграмма деятельности одного процесса представлена на рисунке 3.

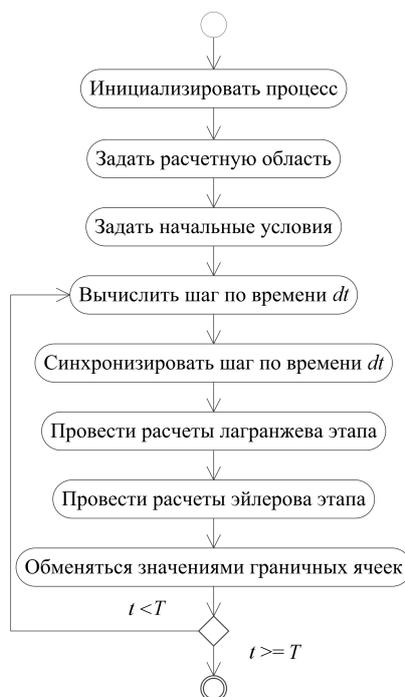


Рис. 3. Диаграмма деятельности одного процесса

В начале работы программы происходит инициализация MPI-процесса. Затем задается расчетная область и начальное распределение всех необходимых величин. Далее в цикле, пока время его выполнения меньше заданного, происходят основные вычисления. Сначала вычисляется шаг по времени  $dt$  и его значение синхронизируется. Затем выполняются лагранжев и эйлеров этапы расчета, после чего следует обмен значениями граничных ячеек.

#### 4. Результаты тестирования

Подробные результаты тестирования точности схемы, выбранной для создания параллельного численного кода, приведены в работе [4]. В данной статье опишем только результаты тестирования производительности вычислений. Укажем лишь, что был выбран вариант схемы с использованием ограничителя наклонов van Leer [6] и метода HLL для решения задачи Римана [5].

Тестирование программы проводилось на двухъядерной, четырехъядерной и восьмиядерной машинах. Результаты увеличения производительности, определяемые как отношение времени выполнения последовательной программы ко времени выполнения параллельной программы, для сеток разных размеров приведены в таблице 1.

Анализ полученных данных показал, что производительность увеличивалась почти линейно по отношению к росту количества процессорных ядер с коэффициентом 0,93.

Таблица 1

## Увеличение производительности вычислительного кода на SMP-системах

Количество ядер	Количество ячеек	Увеличение производительности, разы
2	32 × 32 × 32	1,706
	64 × 64 × 64	1,718
	100 × 100 × 100	1,738
4	32 × 32 × 32	3,217
	64 × 64 × 64	3,398
	100 × 100 × 100	3,403
8	32 × 32 × 32	7,368
	64 × 64 × 64	7,511
	100 × 100 × 100	7,532

Также проводилось тестирование программы на вычислительном кластере, состоящем из одного, двух, четырех и восьми узлов. В таблице 2 приведены результаты увеличения производительности программы на двух, четырех и восьми узлах кластера, для сеток разных размеров, в сравнении с результатами работы на одном узле. Узлы вычислительного кластера, используемые для проведения расчетов, имеют следующие характеристики: Intel(R) Core(TM) 2 CPU E 8400, 2 ГБ ОЗУ; коммуникационная среда — Gigabit Ethernet 1000Base-T.

Таблица 2

## Увеличение производительности вычислительного кода на MPP-системах

Количество узлов	Количество ячеек	Увеличение производительности, MPI, разы	Увеличение производительности, MPI+OpenMP, разы
2	32 × 32 × 32	1,587	1,910
	64 × 64 × 64	1,609	1,992
	100 × 100 × 100	1,618	1,971
4	32 × 32 × 32	3,198	3,796
	64 × 64 × 64	3,274	3,893
	100 × 100 × 100	3,356	3,907
8	32 × 32 × 32	7,269	7,732
	64 × 64 × 64	7,278	7,897
	100 × 100 × 100	7,516	7,949

Анализ полученных данных показал, что производительность увеличивалась почти линейно по отношению к росту количества узлов вычислительного кластера с коэффициентом 0,9.

При комбинации технологий OpenMP+MPI был достигнут коэффициент увеличения производительности 0,98, что говорит об очень хорошем качестве распараллеливания вычислительного кода.

На рисунке 4 показана зависимость ускорения работы программы от количества ядер машины для сетки 100 × 100 × 100. На рисунке 5 представлен график зависимости ускорения работы программы от количества узлов вычислительного кластера для той же сетки.

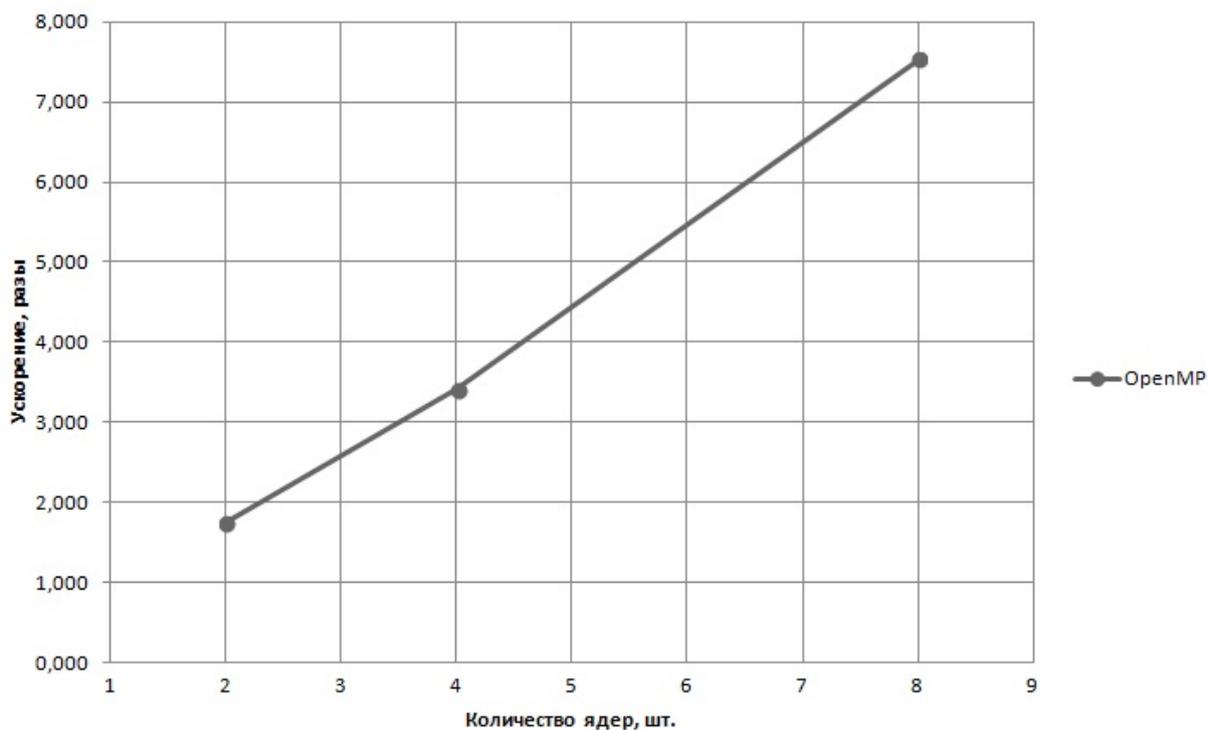


Рис. 4. График зависимости ускорения работы программы от количества ядер машины

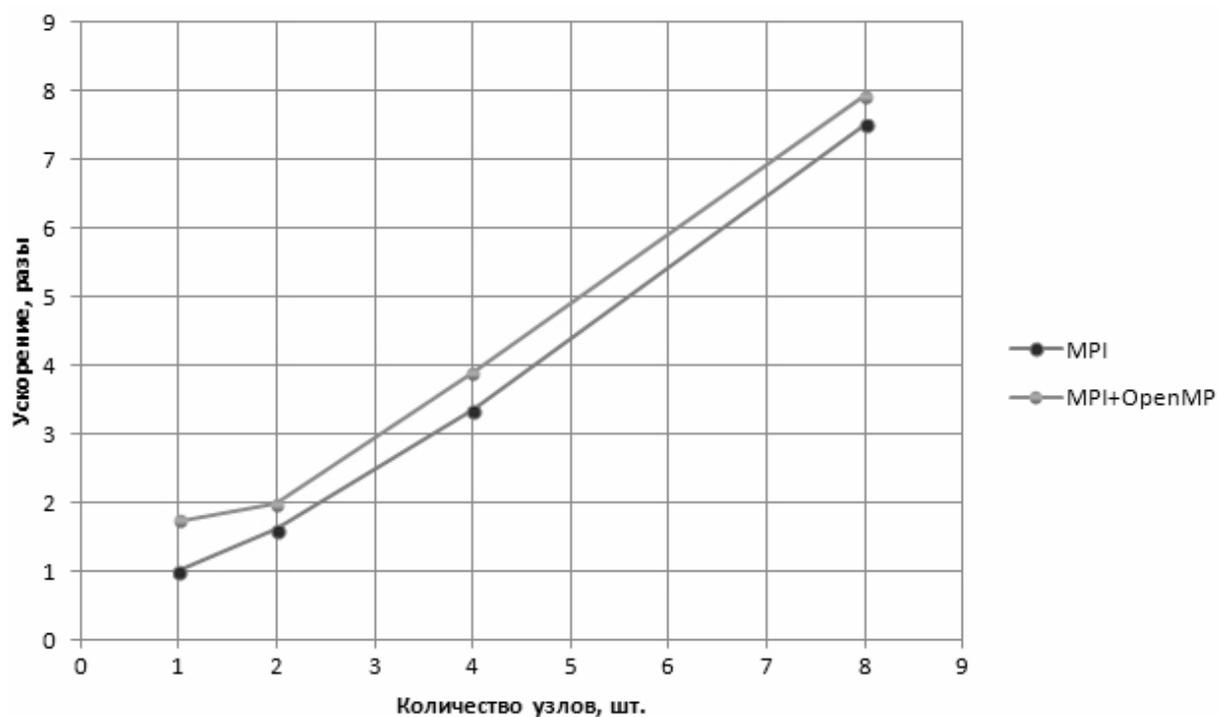


Рис. 5. График зависимости ускорения работы программы от количества узлов вычислительного кластера

### **Заключение**

Вычислительный код был распараллелен достаточно эффективно:

- применение технологии OpenMP привело к ускорению вычислений относительно числа процессорных ядер с коэффициентом 0,93;
- применение технологии MPI привело к ускорению вычислений относительно числа узлов вычислительного кластера с коэффициентом 0,9;
- комбинация технологий OpenMP+MPI позволила достичь ускорения вычислений с коэффициентом 0,98.

При комбинации подходов MPI+OpenMP на двух узлах выполнение программы ускорилось примерно в 1,96 раз (в зависимости от размеров расчетной сетки); на четырех узлах — примерно в 3,87 раз; на восьми узлах — примерно в 7,86 раз. Полученное ускорение увеличилось в среднем на 10–15% по сравнению с результатами применения одной технологии MPI.

Созданная программа может быть использована для численного решения трехмерных уравнений газодинамики на подробных сетках на вычислительных кластерах, состоящих из многоядерных узлов.

### **ПРИМЕЧАНИЕ**

<sup>1</sup> Работа выполнена при поддержке гранта РФФИ № 14-08-97044 p\_поволжье\_a.

### **СПИСОК ЛИТЕРАТУРЫ**

1. Жумалиев, А. Г. Численная схема cSPH-TVD: моделирование фронта ударной волны / А. Г. Жумалиев, С. С. Храпов // Вестник Волгоградского государственного университета. Серия 1, Математика. Физика. — 2012. — № 2 (17). — С. 60–67.
2. Кайгородов, П. В. Практические вопросы адаптации вычислительных кодов, использующих явные методы, к многопроцессорной архитектуре / П. В. Кайгородов. — М. : Препринт ИПМ им. М. В. Келдыша, 2002. — 28 с.
3. Численная схема для моделирования динамики поверхностных вод на основе комбинированного SPH-TVD подхода / С. С. Храпов, А. В. Хоперсков, Н. М. Кузьмин, А. В. Писарев, И. А. Кобелев // Вычислительные методы и программирование. — 2011. — Т. 12, № 1. — С. 282–297.
4. Численная схема cSPH-TVD: исследование влияния ограничителей наклонов / Н. М. Кузьмин, А. В. Белоусов, Т. С. Шушкевич, С. С. Храпов // Вестник Волгоградского государственного университета. Серия 1, Математика. Физика. — 2014. — № 1 (20). — С. 22–34.
5. Harten, A. On upstream differencing and Godunov type methods for hyperbolic conservation laws / A. Harten, P. Lax, B. van Leer // SIAM Review. — 1983. — Vol. 25, № 1. — P. 35–61.
6. Leer, B. van. Towards the ultimate conservative difference scheme. III. Upstream-centered finite-difference schemes for ideal compressible flow / B. van Leer // Journal of Computational Physics. — 1977. — Vol. 23, № 3. — P. 263–275.

## REFERENCES

1. Zhumaliev A.G., Khrapov S.S. Chislennaya skhema cSPH-TVD: modelirovanie fronta udarnoy volny [Numerical scheme cSPH-TVD: front of shock wave simulation]. *Vestnik Volgogradskogo gosudarstvennogo universiteta. Seriya 1, Matematika. Fizika* [Science Journal of Volgograd State University. Mathematics. Physics], 2012, no. 2 (17), pp. 60-67.
2. Kaygorodov P.V. *Prakticheskie voprosy adaptatsii vychislitelnykh kodov, ispolzuyushchikh yavnye metody, k mnogoprotsessornoy arkhitekture* [Practical questions about adaptation explicit numerical codes to multiprocessor architecture]. Moscow, Preprint IPM im. M. V. Keldysha Publ., 2002. 28 p.
3. Khrapov S.S., Khoperskov A.V., Kuzmin N.M., Pisarev A.V., Kobelev I.A. Chislennaya skhema dlya modelirovaniya dinamiki poverkhnostnykh vod na osnove kombinirovannogo SPH-TVD podkhoda [The numerical scheme for simulation of dynamics of surface water on the base of mixed SPH-TVD approach]. *Vychislitelnye metody i programmirovaniye* [Numerical methods and programming], 2011, vol. 12, no. 1, pp. 282-297.
4. Kuzmin N.M., Belousov A.V., Shushkevich T.S., Khrapov S.S. Chislennaya skhema cSPH-TVD: issledovanie vliyaniya ogranichiteley naklonov [Numerical scheme cSPH-TVD: investigation of influence slope limiters]. *Vestnik Volgogradskogo gosudarstvennogo universiteta. Seriya 1, Matematika. Fizika* [Science Journal of Volgograd State University. Mathematics. Physics], 2014, no. 1 (20), pp. 22-34.
5. Harten A., Lax P., Leer B. van. On Upstream Differencing and Godunov Type Methods For Hyperbolic Conservation Laws. *SIAM Review*, 1983, vol. 25, no. 1, pp. 35-61.
6. Leer B. van. Towards the Ultimate Conservative Difference Scheme. III. Upstream-Centered Finite-Difference Schemes For Ideal Compressible Flow. *Journal of Computational Physics*, 1977, vol. 23, no. 3, pp. 263-275.

**THE THREE-DIMENSIONAL PARALLEL NUMERICAL CODE ON THE BASE OF MIXED LAGRANGE-EULERIAN APPROACH**

**Tatyana Sergeevna Shushkevich**

Student, Institute of Mathematics and Information Technologies,  
Volgograd State University  
infomod@volsu.ru  
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

**Nikolay Mikhaylovich Kuzmin**

Candidate of Physical and Mathematical Sciences, Associate Professor,  
Department of Information Systems and Computer Simulation,  
Volgograd State University  
nmkuzmin@gmail.com, infomod@volsu.ru  
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

**Mariya Anatolevna Butenko**

Senior Lecturer, Department of Information Systems  
and Computer Simulation,  
Volgograd State University  
butenkoma@gmail.com, infomod@volsu.ru  
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

**Abstract.** In this paper the parallel numerical code developed for integration of gas-dynamics equations in the three-dimensional case on the base of mixed Lagrange-Eulerian approach was described. The method of paralleling, allowing effective use of computation properties of computer cluster with multiprocessor

nodes, was shown. The results of performance testing, which shown significant growth of computations comparing to serial version of the software was shown.

For parallelization of numerical code, combination of OpenMP and MPI technologies was used: on each node of computer cluster runs multithread calculation process. Methodology of data parallelism was implemented (in other words, single instruction — multiple data). For dividing data between calculation processes, partitioning of computational domain along one axis was used.

OpenMP technology leads to acceleration of calculations relative to number of processor cores with coefficient 0.93.

MPI technology leads to acceleration of calculations relative to number of computation cluster nodes with coefficient 0.9.

Combination of OpenMP+MPI leads to acceleration of calculations with coefficient 0.98.

Testing of computational performance was made on computer cluster with these characteristics: Intel(R) Core(TM) 2 CPU E 8400, 2 Gb RAM, communication media — Gigabit Ethernet 1000Base-T. The results have shown that the numerical code was effectively parallelized: with using combination MPI+OpenMP on two-node cluster computational performance growth to approximately in 1.96 times (computational grid size depended); on four nodes — approximately in 3.87 times; on eight nodes — approximately in 7.86 times. Total computational performance has grown on the average by 10–15 % instead of results obtained by using only MPI.

The developed computer program can be used for numerical solving of the three-dimensional gas-dynamics equations with detailed grids with computational clusters of multiprocessor nodes.

**Key words:** numerical simulation, gas-dynamics, Lagrange-Eulerian approach, parallel technologies, high-performance computations.