УДК 004.75

Рыбаков А.А.

Межведомственный суперкомпьютерный центр Российской академии наук - филиал Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», г. Москва, Россия

РАСПРЕДЕЛЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ НАГРУЗКИ МЕЖДУ УЗЛАМИ СУПЕРКОМПЬЮТЕРНОГО КЛАСТЕРА ПРИ РАСЧЕТАХ ЗАДАЧ ГАЗОВОЙ ДИНАМИКИ С ДРОБЛЕНИЕМ РАСЧЕТНОЙ СЕТКИ

АННОТАЦИЯ

В статье рассматривается проблема распределения между узлами суперкомпьютерного кластера вычислительной нагрузки по обработке блочно-структурированной сетки. Приводится архитектура внутреннего представления сетки, описание механизма дробления сетки и распределения блоков сетки по вычислительным процессам.

КЛЮЧЕВЫЕ СЛОВА

Суперкомпьютер; математическое моделирование; блочно-структурированная расчетная сетка; балансировка вычислительной нагрузки.

Rybakov A.A.

Joint Supercomputer Center of the Russian Academy of Sciences - branch of Scientific Research Institute of System Analysis of the Russian Academy of Sciences, Moscow, Russia

COMPUTATIONAL WORKLOAD DISTRIBUTION BETWEEN SUPERCOMPUTER NODES FOR FLUID DYNAMICS CALCULATIONS WITH GRID FRAGMENTATION USING

ABSTRACT

In the article the problem of block-structured grid processing computational workload distribution between supercomputer nodes is considered. Grid inner representation architecture, grid fragmentation mechanism and distribution grid blocks between calculation processes are described.

KEYWORDS

Supercomputer; mathematical modeling; block-structured calculation grid; computational workload balancing.

При численном моделировании задач газовой динамики часто используют блочно-структурированные сетки [1,2]. Блочно-структурированные сетки состоят из отдельных блоков, каждый из которых представляет собой упорядоченный трехмерный массив условно кубических ячеек. Упорядоченность размещения ячеек позволяет быстрее производить вычисления и снижает требования к объему памяти, однако строить блочно-структурированные сетки гораздо сложнее, чем неструктурированные.

Блочно-структурированные сетки могут иметь сложную форму. Для доступа к отдельным ячейкам блока вводятся три индекса, связанные с криволинейной системой координат блока. Система координат задается тремя линиями (I, J, K), каждая из которых связывает пары противоположных граней блока. Блоки могут граничить между собой. Граница между блоками описывается интерфейсом. Один интерфейс описывает соприкосновение двух блоков прямоугольными подобластями своих граней. При этом системы координат двух соседних блоков не обязаны согласовываться между собой.

При использовании суперкомпьютера расчет задачи выполняется сразу в нескольких параллельных процессах, отсюда возникает задача распределения блоков расчетной сетки по различным вычислительным процессам [3]. При этом равномерность распределения блоков (по суммарному количеству ячеек) между процессами имеет важное значение, так как общее время работы всех процессов определяется по максимальному времени работы отдельного процесса.

Расчет газодинамической задачи носит итерационный по времени характер. Для каждой итерации по времени выполняется пересчет данных ячеек согласно той или иной вычислительной схеме. Во время обработки одной ячейки возникает потребность обращаться за данными к ячейкам ее окрестности (в простейшем случае это просто данные соседних по граням ячеек, но могут использоваться и используются более сложные структуры окрестностей. Во время произведения расчетов газодинамической задачи различные блоки сетки обрабатываются независимо друг от друга. При этом некоторые ячейки обрабатываемого блока должны обращаться за данными к ячейкам соседних блоков. Это происходит в том случае, если окрестность ячейки расположена сразу в нескольких блоках. Для эффективного счета необходимо обеспечить функционал быстрого обмена данными между блоками. Если два соседних блока обрабатываются на разных узлах суперкомпьютерного кластера, то для организации таких обменов можно использовать Message Passing Interface [4].

Одним из важнейших действий управления расчетной блочно-структурированной сеткой при выполнении вычислений на суперкомпьютере является дробление ее блоков. Так как при запуске задач на суперкомпьютере постоянно возрастает степень параллельности (используется все больше параллельных процессов обработки блоков сетки), то для сохранения равномерности распределения блоков по вычислительным процессам требуется уметь измельчать блоки. Блок сетки может быть разделен на два блока по любому из трех направлений: I, J, K.

Кроме блоков сетка содержит другие объекты, которые требуют корректировки после разделения блока. Сюда относятся интерфейсы, описывающие касание блоков друг друга. На границе расчетной области граничные условия задаются с помощью специальных объектов, которые также должны быть разделены в случае пересечения их линией разреза блока. Также должны быть по необходимости разделены области, описывающие начальные условия. Каждый из этих объектов имеет жесткую привязку к блоку, а значит после дробления может возникнуть необходимость разделения этого объекта.

Граничные условия и области начальных условий обрабатываются наиболее просто и похожим образом. Рассмотрим, например, граничные условия в двумерном случае (в координатах IJ).

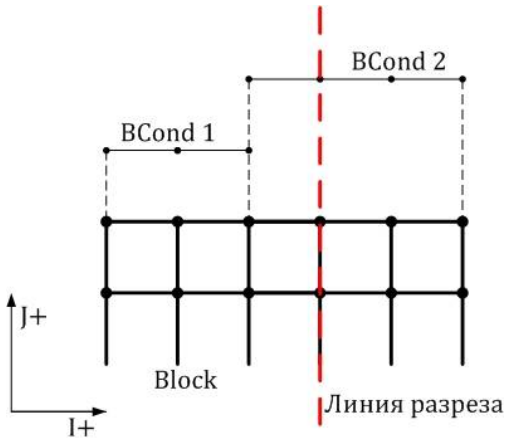

Рис.1. Дробление блока Block по линии разреза, обозначенной красным, провоцирует дробление граничного условия BCond 2, но не влияет на граничное условие BCond 1

Пусть блок Block должен быть разрезан по направлению I+ как показано на рис. 1. Пусть данный блок имеет граничные условия по направлению J+. Тогда возможны два варианта. Либо линия разреза не пересекает граничное условие, и тогда граничное условие целиком отходит одному из результирующих блоков (BCond 1). Если же линия разреза проходит через граничное условие, то данное граничное условие также должно быть разделено, и его части отойдут двум результирующим блокам (рис. 2).

Такая же ситуация может сложиться в отношении областей начальных условий. В случае дробления интерфейсов могут быть более сложные ситуации дробления, но их рассмотрение выходит за рамки данной статьи.

Для равномерного распределения блоков сетки по вычислительным процессам рассмотрим следующую задачу разделения множества весов на m различных множеств.

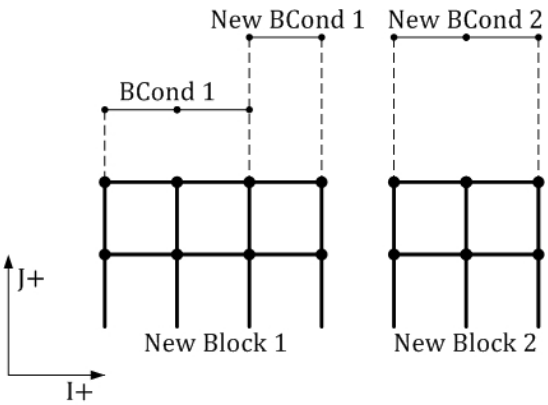

Рис.2. Результат дробления блока Block из рис. 1

Рассмотрим множество $\mathbb{X}$ вещественных чисел $x_i \geq 0$ для $i \in N$ где $N = [1, n]$. Рассмотрим также множество индексов $j \in M$, где $M = [1, m]$. Будем говорить, что определено разбиение множества $\mathbb{X}$ на m множеств, если введена функция $\gamma(i)$: $N \to M$. Множество всех функций разбиения будем обозначать $\Gamma(N, M)$. Веса результирующих множеств будем определять естественным образом для $j \in M$:

$$X_j = \sum_{i \in N, \gamma(i)=j} x_i.$$

Требуется найти такую функцию разбиения $\gamma \in \Gamma(N, M)$, чтобы минимизировать наиболее тяжелое из результирующих множеств.

$$\min_{\gamma \in \Gamma(N,M)} \max_{j \in M} X_j.$$

Задача может быть расширена на случай распределения вычислительной нагрузки между различными вычислителями суперкомпьютера (например процессорами Intel Xeon и сопроцессорами Inte Xeon Phi). При этом формулировка задачи меняется только в части приведения всех узлов к одному показателю с помощью весовых коэффициентов.

Коэффициентом приведения $\kappa(j)$ для $j \in M$ назовем такую положительную функцию $\kappa(j)$: $M \to \mathbb{R}_+$, что время выполнения нагрузки $\kappa(j)$ на узле j не зависит от j. Тогда в общем виде задача о равномерном разбиении множества весов на m множеств с коэффициентами приведения $\kappa(j)$ для $j \in M$ формулируется следующим образом. Требуется найти такую функцию разбиения $\gamma \in \Gamma(N, M)$, чтобы минимизировать наиболее тяжелое из результирующих множеств с учетом коэффициентов приведения:

$$\min_{\gamma \in \Gamma(N,M)} \max_{j \in M} \big(\kappa(j) X_j\big).$$

Данная задача имеет практическое применение при распределении вычислительной нагрузки между вычислителями гетерогенного суперкомпьютера. В данной статье рассматривается только задача разбиения множества весов на m множеств без учета коэффициентов приведения.

Опишем жадный алгоритм равномерного разбиения множества весов на m весов следующим образом. Все веса записываются в массив необработанных весов. После того, как веса будут обрабатываться, они будут удаляться из массива. Алгоритм работает до тех пор, пока массив необразованных весов не пуст. Если в массиве остались элементы, то берем самый тяжелый из них. Далее ассоциируем этот элемент с тем результирующим множеством, для которого сумма ассоциированных с ним весов на текущий момент минимальна. После чего удаляем этот элемент из массива необработанных и продолжаем работу.

Приведем без доказательства оценку эффективности работы алгоритма. Для удобства будем считать, что изначальное множество весов упорядочено по убыванию.

Определим остаточный член r_i для $i \in N$ по следующей формуле:

$$r_i = \max(x_i - \tfrac{1}{m} \textstyle\sum_{t=i}^{n} x_t\,, 0).$$

Тогда можно доказать следующее соотношение:

$$\max_{j \in M} X_j - \langle X \rangle \leq \max_{i \in N} r_i,$$

где

$$\langle X \rangle = \tfrac{1}{m} \textstyle\sum_{j \in M} X_j.$$

Таким образом, для оценки эффективности описанного жадного алгоритма распределения весов по m множествам достаточно проанализировать ряд остаточных членов, полученный из отсортированного массива распределяемых весов.

Задачу о равномерном распределении весов по результирующим множествам можно применить для равномерного распределения блоков по вычислительным процессам суперкомпьютера в предположении, что все процессы равнозначны (не рассматривается случай гетерогенной системы). Для этого отметим следующие моменты. В качестве веса блока нужно взять количество его ячеек. Так как абсолютно равномерное распределение блоков по вычислительным процессам не всегда возможно, то нужно принять порог максимально допустимого отклонения количества ячеек одного процесса от среднего значения, при достижении которого распределение можно считать успешным. Экспериментальным путем установлено, что порог максимального отклонения в 10% является достаточным для достижения эффективного распределения. Для оценки качества распределения текущего набора весов можно воспользоваться оценками эффективности, приведенными выше. Если текущее распределение не удовлетворяет допустимому порогу отклонения количества ячеек от среднего значения, то следует раздробить наиболее крупные блоки, после чего повторно применить алгоритм жадного распределения.

Предложенные методы равномерного распределения вычислительной нагрузки между узлами суперкомпьютерного кластера были опробованы на суперкомпьютере МВС-10П [5], находящемся в МСЦ РАН. Приведем некоторые результаты, которые были получены для двух расчетных сеток.

Первая рассматриваемая сетка, содержит 13 блоков, 80 интерфейсов, 148 граничных условий, 13 областей начальных условий и 5750102 ячейки. Размер вычислительной окрестности равен 3.

Ниже приведен график, на котором показана статистика общего количества ячеек блоков, а также количества внутренних, граничных и интерфейсных ячеек с учетом кратности.

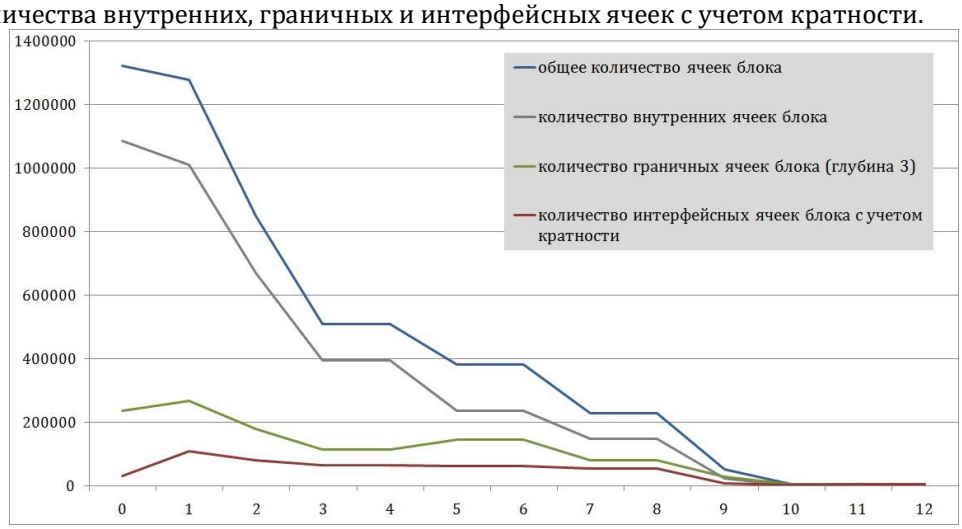

Рис.3. Статистика количества ячеек разных типов блоков первой сетки

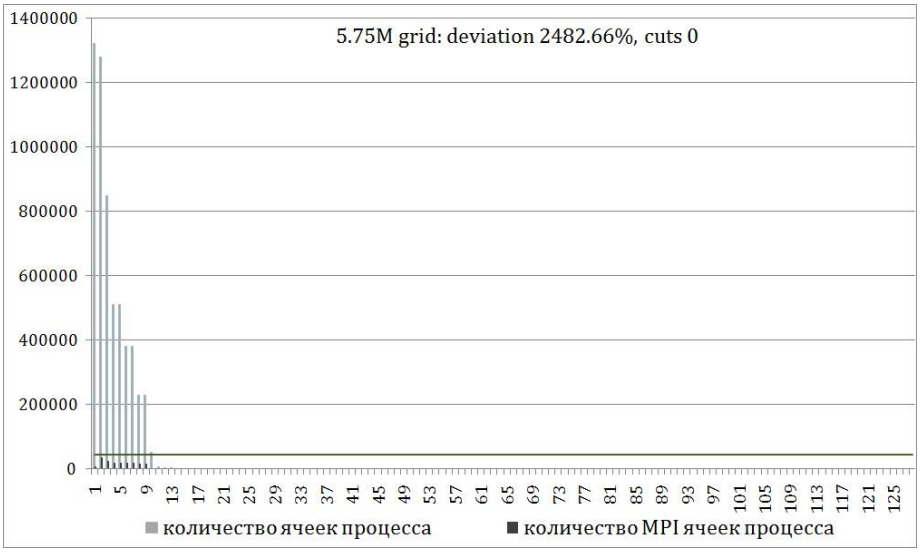

Рис.4. Распределение блоков первой сетки жадным алгоритмом на 128 процессов без использования дробления (зеленым показана линия среднего значения)

Для этой сетки было проведено распределение на 128 параллельных процессов. Приведем статистику распределения блоков по вычислительным процессам для двух различных вариантов: без использования дробления блоков и с использованием дроблений для достижения показателя качества распределения 10% максимального отклонения.

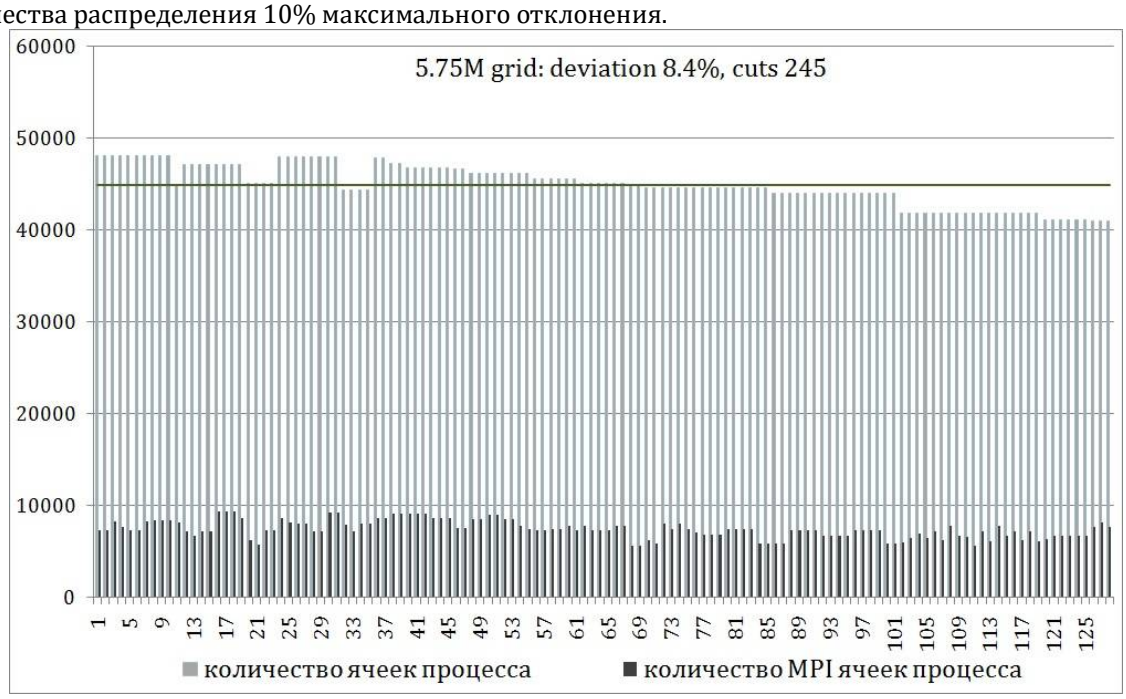

Рис.5. Распределение блоков первой сетки жадным алгоритмом на 128 процессов с использования дробления до показателя 10% отклонения от среднего (показано зеленым цветом)

Аналогичные тестовые запуски производились для сетки, содержащей 300 блоков, 1796 интерфейсов, 1643 граничных условия, 300 областей начальных данных и 94336290 ячейки. Размер вычислительной окрестности также равен 3.

Ниже приведен график, на котором показана статистика общего количества ячеек блоков, а также количества внутренних, граничных и интерфейсных ячеек с учетом кратности.

Рис.6. Статистика количества ячеек разных типов блоков второй сетки

Для этой сетки также было осуществлено распределение на 128 параллельных процессов. Приведем статистику распределения блоков по вычислительным процессам для двух различных вариантов: без использования дробления блоков и с использованием дроблений для достижения показателя качества распределения 10% максимального отклонения.

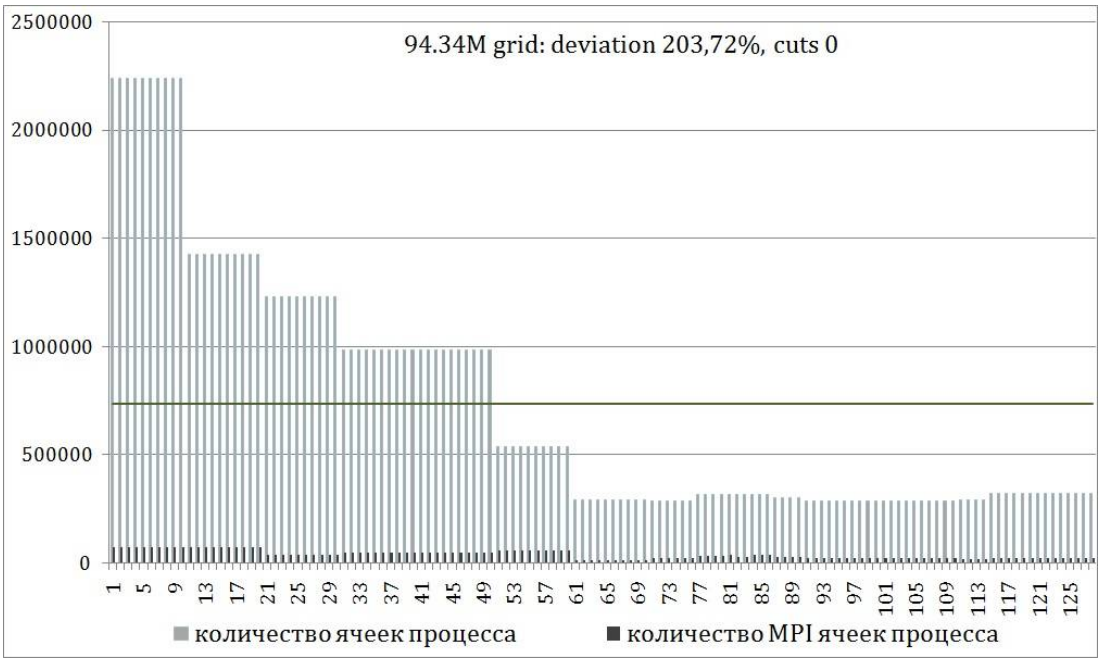

Рис.7. Распределение блоков второй сетки жадным алгоритмом на 128 процессов без использования дробления (зеленым показана линия среднего значения)

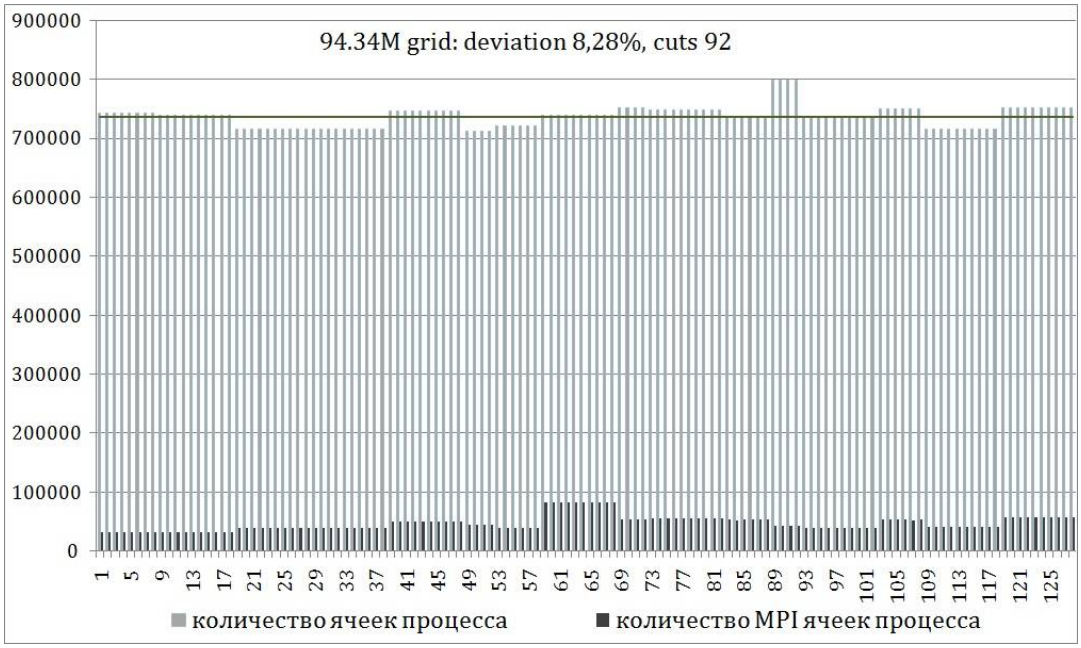

Рис.8. Распределение блоков второй сетки жадным алгоритмом на 128 процессов с использования дробления до показателя 10% отклонения от среднего (показано зеленым цветом)

Таким образом, применение дробления расчетной сетки привело к ускорению производимых расчетов для указанных сеток в 27.5 и 2.8 раз соответственно при запуске обработки сетки в 128 параллельных процессах.

Предложенный механизм равномерного распределения блоков блочно-структурированной сетки между узлами суперкомпьютерного кластера приводит к равномерной загрузке вычислительных ресурсов суперкомпьютерного кластера, что повышает эффективность его использования в расчетах задач газовой динамики. Часто применение такого подхода на большом количестве процессов приводит к кратному ускорению вычислений. Особенно это актуально для сеток содержащих небольшое количество блоков или для сеток, имеющих ярко выраженные крупные блоки.

Литература

1. Blazek J. Computational fluid dynamics: Principles and applications. - Elsevier, 2001.
2. Farrashkhalvat M., Miles J.P. Basic structured grid generation, with an introduction to unstructured grid generation. - Butterworth-Heinemann, 2003.
3. Шабанов Б.М., Телегин П.Н., Аладышев О.С. Особенности использования многоядерных процессоров. - Программные продукты и системы, №2 (82), сс. 7-9, 2008.
4. Queen M. Parallel programming in C with MPI and OpenMP. - Mc-Grow Hill, 2004.
5. Описание интерфейса пользователя, предназначенного для работы с интеловской гибридной архитектурой суперЭВМ (СК), где вместе с процессорами Intel Xeon используются сопроцессоры Intel Xeon Phi. URL http://www.jscc.ru/informat/MVS-10PInter.pdf

References

1. Blazek J. Computational fluid dynamics: Principles and applications. - Elsevier, 2001.
2. Farrashkhalvat M., Miles J.P. Basic structured grid generation, with an introduction to unstructured grid generation. - Butterworth-Heinemann, 2003.
3. Shabanov B.M., Telegin P.N., Aladyshev O.S. Osobennosti ispol'zovaniya mnogoyadernykh protsessorov. - Programmnye produkty i sistemy, №2 (82), ss. 7-9, 2008.
4. Queen M. Parallel programming in C with MPI and OpenMP. - Mc-Grow Hill, 2004.
5. Opisanie interfeysa pol'zovatelya, prednaznachennogo dlya raboty s intelovskoy gibridnoy arkhitekturoy superEVM (SK), gde vmeste s protsessorami Intel Xeon ispol'zuyutsya soprotsessory Intel Xeon Phi. URL http://www.jscc.ru/informat/MVS-10PInter.pdf

Поступила 18.10.2016

Об авторах:

Рыбаков Алексей Анатольевич, ведущий научный сотрудник Межведомственного суперкомпьютерного центра Российской академии наук - филиала Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», кандидат физико-математических наук, rybakov@jscc.ru.