

Декомпозиция поверхностной неструктурированной расчетной сетки для масштабирования вычислений на суперкомпьютере

А. А. Рыбаков*, А. Д. Чопорняк

ФГУ «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», г. Москва, Российская Федерация

117218, Российская Федерация, г. Москва, пр. Нахимовский, д. 36, корп. 1

* rybakov@jssc.ru

Аннотация

В настоящее время при решении сложных вычислительных задач компьютерного моделирования используются расчетные сетки, содержащие десятки и сотни миллионов ячеек. Для выполнения вычислений такого объема требуется использование суперкомпьютерных кластеров, состоящих из многих вычислительных узлов, связанных между собой высокоскоростной коммуникационной сетью. При этом необходимо выполнять декомпозицию расчетной сетки на отдельные домены. Эти домены распределяются по вычислительным узлам суперкомпьютера и обрабатываются параллельно, независимо друг от друга. Для синхронизации вычислений после каждой итерации обработки ячеек производятся обмены данными на границах между соседними соприкасающимися доменами. Для эффективности выполнения вычислений и масштабирования их на большое количество вычислительных узлов требуется разработка эффективных алгоритмов декомпозиции расчетных сеток, порождающих множество доменов с накладываемыми на них требованиями по количеству ячеек, равномерности распределения ячеек по доменам, связности доменов и размеру границ между ними. В статье в качестве объекта исследования рассматривается неструктурированная поверхностная сетка, используемая для расчета процессов взаимодействия объемного тела с окружающей средой. Для сетки такого вида рассматриваются различные классы алгоритмов декомпозиции и предлагается иерархический алгоритм декомпозиции с выбором оптимального критерия разбиения доменов. Предложенный алгоритм реализован в программном коде, приводятся результаты сравнения работы данного алгоритма на тестовых сетках с алгоритмами других классов.

Ключевые слова: суперкомпьютер, поверхностная неструктурированная расчетная сетка, декомпозиция, домен, высокопроизводительные вычисления, масштабирование вычислений.

Авторы заявляют об отсутствии конфликта интересов.

Финансирование: настоящая работа выполнена в Межведомственном суперкомпьютерном центре Российской академии наук - филиале ФГУ «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук» в рамках государственного задания по теме 580-2021-0016. Разработанные в рамках данной работы программные коды были использованы для запусков на суперкомпьютере МВС-10П ОП, находящемся в МСЦ РАН.

Для цитирования: Рыбаков, А. А. Декомпозиция поверхностной неструктурированной расчетной сетки для масштабирования вычислений на суперкомпьютере / А. А. Рыбаков, А. Д. Чопорняк. – DOI 10.25559/SITITO.16.202004.851-861 // Современные информационные технологии и ИТ-образование. – 2020. – Т. 16, № 4. – С. 851-861.

© Рыбаков А. А., Чопорняк А. Д., 2020



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Decomposition of a Surface Unstructured Computational Mesh for Scaling Computations on a Supercomputer

A. A. Rybakov*, A. D. Chopornyak

Scientific Research Institute for System Analysis of the Russian Academy of Sciences, Moscow, Russian Federation 36/1 Nakhimovsky Ave., Moscow 117218, Russian Federation

* rybakov@jssc.ru

Abstract

Currently, when solving complex computational problems of computer modeling, computational grids containing tens and hundreds of millions of cells are used. To perform computations of such amount, it is required to use supercomputer clusters consisting of many computational nodes connected by a high-speed communication network. In this case, it is necessary to decompose the computational mesh into separate domains. These domains are distributed among the computational nodes of the supercomputer and are processed in parallel, independently of each other. To synchronize computations, after each iteration of cell processing, data exchanges are performed at the boundaries between adjacent contiguous domains. To efficiently perform computations and scale them to a large number of computational nodes, it is necessary to develop efficient algorithms for decomposition of computational meshes that generate many domains with imposed requirements for the number of cells, the uniformity of the distribution of cells across domains, the connectivity of domains and the size of boundaries between them. The article considers an unstructured surface mesh as an object of research, which is used to calculate the processes of interaction of a volumetric body with the environment. For a mesh of this type, various classes of decomposition algorithms are considered, and a hierarchical decomposition algorithm is proposed with the choice of the optimal criterion for partitioning domains. The proposed algorithm is implemented in the program code, the results of comparing the operation of this algorithm on test meshes with algorithms of other classes are presented.

Keywords: supercomputer, surface unstructured computational mesh, decomposition, domain, high performance computing, computation scaling.

Funding: This work was carried out at the Interdepartmental Supercomputer Center of the Russian Academy of Sciences in the branch of the Federal State Institution "Federal Scientific Center Research Institute for System Research of the Russian Academy of Sciences" within the framework of the state assignment on topic 580-2021-0016. The program codes developed within the framework of this work were used for launches on the MBC-10П ОП supercomputer located in the Interdepartmental Supercomputer Center of the Russian Academy of Sciences.

The authors declare no conflict of interest.

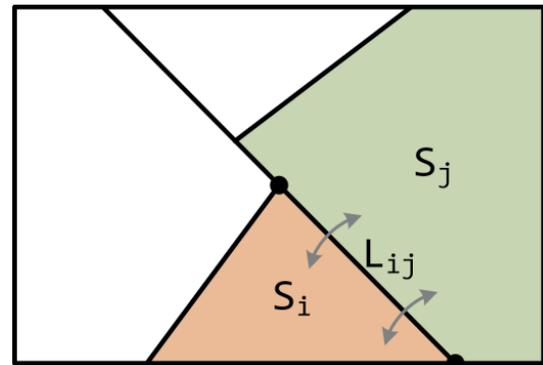
For citation: Rybakov A.A., Chopornyak A.D. Decomposition of a Surface Unstructured Computational Mesh for Scaling Computations on a Supercomputer. *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2020; 16(4):851-861. DOI: <https://doi.org/10.25559/SITITO.16.202004.851-861>



Введение

Современные расчетные приложения крайне требовательны к вычислительным ресурсам¹ [1], [2], [3]. Для больших задач не представляется возможным выполнение их на отдельно взятом вычислителе (одном микропроцессоре или одном сервере) за приемлемое время. Возникает потребность использовать для вычислений суперкомпьютерные кластеры, состоящие из многих вычислительных узлов. Для того, чтобы выполнить задачу на суперкомпьютере, необходимо разделить ее расчетную область на отдельные подобласти, называемые доменами, и обрабатывать эти домены параллельно и независимо друг от друга [4]. Для повышения эффективности суперкомпьютерных приложений внутри вычислительного узла применяются различные методы подготовки данных [5] и распараллеливания исполнения для систем с общей памятью [6], [7], а также низкоуровневые оптимизации программного кода, такие как векторизация [8], [9], позволяющие существенно повысить скорость выполнения приложений. Конечно, на границах соприкосновения доменов возникает необходимость синхронизации вычислений, что достигается путем обмена данными (например, с использованием MPI [10-12]). Таким образом, выполнение суперкомпьютерных вычислений состоит из двух чередующихся шагов: параллельная обработка ячеек расчетной области и обмен данными на границах соприкосновения доменов, обрабатываемых различными процессами. Эффективность выполнения суперкомпьютерных приложений существенным образом зависит от качества декомпозиции расчетной сетки и ее распределения между разными вычислительными узлами.

Данная статья посвящена задаче декомпозиции поверхностной неструктурированной расчетной сетки для распределения между узлами гомогенного суперкомпьютерного кластера (то есть кластера, состоящего из одинаковых вычислительных узлов) для повышения эффективности масштабирования вычислений [13]. Пусть дана поверхностная сетка, состоящая из S расчетных ячеек, пусть также суперкомпьютер состоит из n вычислительных узлов с одинаковыми характеристиками. Также будем считать, что скорость обмена данными между любыми двумя вычислительными узлами одинакова для всех узлов. Распространение задачи распределения вычислительной нагрузки на гетерогенный вычислительный кластер достигается путем ввода весовых коэффициентов для вычислительных узлов и для каналов обмена данными, как это описано в [14]. Если представить скорость обработки расчетных ячеек на одном вычислительном узле как a^{-1} , то время выполнения одной итерации расчетов на одном вычислительном узле будет равно $T_1 = aS$. Пусть теперь расчетная область разбита на n доменов, содержащих по S_i ячеек ($i = 1, n$). Обозначим через L_{ij} количество ребер, составляющих границу между доменами S_i и S_j (см. рис. 1).



Р и с. 1. Иллюстрация разбиения расчетной сетки на домены
F i g. 1. Illustration of dividing the computational grid into domains

Будем считать, что каждый домен обрабатывается на своем вычислительном узле. Таким образом, все домены обрабатываются параллельно, и время обработки всех ячеек определяется временем обработки самого крупного домена. Кроме обработки всех расчетных ячеек после проведения итерации расчета необходимо выполнить обмен данными между всеми парами доменов по всем границам между ними. Пусть скорость передачи данных между узлами определяется величиной b^{-1} , и все обмены выполняются параллельно, тогда время выполнения всех обменов определяется временем обмена данными через самую длинную границу. Исходя из этого, можно определить суммарное время выполнения одной итерации расчета при выполнении на n вычислительных узлах:

$$T_n = a \max_{i=1,n} S_i + b \max_{j=1,n} L_{ij}$$

Критерием оптимизации декомпозиции расчетной сетки является сокращение времени выполнения расчетов, то есть величины T_n . Время выполнения расчетов напрямую зависит от размера самого крупного домена, однако мы будем рассматривать не абсолютный размер домена, а его отклонение от теоретического оптимального значения. Очевидно, что в идеальном случае при декомпозиции все домены должны иметь размер S/n , а относительное отклонение от идеального размера в процентах можно вычислить по формуле

$$D = 100\% \left(\frac{n}{S} \max_{i=1,n} S_i - 1 \right)$$

Вторым важным критерием качества выполненной декомпозиции является наибольшее значение длины границы между парами доменов. В данном случае можно использовать абсолютную характеристику, так как длину границы предсказать довольно сложно, и у нее вообще говоря нет теоретического минимума (в зависимости от геометрии рассматриваемой сетки теоретически длина границы между доменами может быть нулевой). То есть в качестве критерия сравнения различных методов декомпозиции сетки будем использовать следующую величину:

$$L = \max_{i=1,n} L_{ij}$$

Несмотря на то, что в наших предположениях все обмены данными между доменами выполняются одновременно, общий

¹ Bell C. G. Supercomputers: The Amazing Race. Technical Report MSR-TR-2015-2, 2015. [Электронный ресурс]. DOI: <https://doi.org/10.13140/2.1.3425.0562>



объем всех обменов существенно влияет на скорость обмена данными, поэтому этот параметр также необходимо учитывать. Введем его в следующем виде. Общее число ребер расчетной сетки остается неизменным вне зависимости от алгоритма декомпозиции и количества доменов (обозначим общее количество ребер сетки через E). Среди этих ребер есть граничные ребра, имеющие только одну инцидентную ячейку, их количество также неизменно и равно E_B . Остальные ребра имеют две инцидентные ячейки. Если обе ячейки, инцидентные некоторому ребру, принадлежат одному и тому же домену, то будем называть такое ребро внутренним ребром этого домена (обозначим их количество через E_{INN} от слова inner), в противном случае ребро входит в границу между двумя доменами — будем называть такое ребро междоменным (обозначим их количество через E_{INT} от слова inter). Ребер других видов быть не может. Таким образом, выполняется соотношение $E = E_B + E_{INN} + E_{INT}$. В качестве параметра оценки качества декомпозиции будем рассматривать долю междоменных ребер в общем количестве ребер сетки, то есть величину

$$I = 100\% \left(\frac{E_{INT}}{E} \right)$$

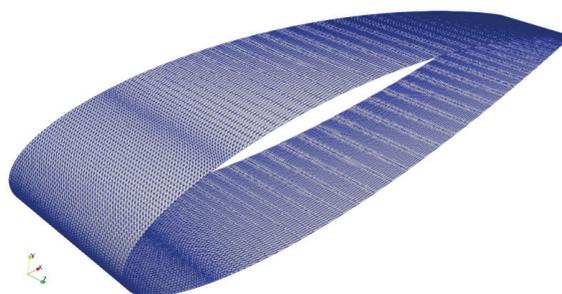
Для оценки качества декомпозиции расчетной сетки будет использовать все три описанных параметра: D — отклонение размера максимального домена от идеального значения, I — долю междоменных ребер в общем числе ребер расчетной сетки и L — длину наиболее протяженной границы между парами доменов. Чем ниже значения данных критериев, тем более качественным является декомпозиция. На основании этого выполним анализ некоторых известных алгоритмов декомпозиции, которые могут быть применены к поверхностным неструктурированным расчетным сеткам.

Описание некоторых методов декомпозиции расчетных сеток

Все описанные в работе алгоритмы декомпозиции рассматриваются на примере тестовой трехмерной поверхностной сетки (wing), состоящей из ячеек-треугольников. Сетка представляет собой профиль крыла летательного аппарата и содержит 9900 узлов, 29205 ребер и 19306 ячеек. Внешний вид тестовой расчетной сетки представлен на рис. 2. Данный трехмерный профиль был сгенерирован из двумерного профиля NASA 0012².

В качестве первого и самого простого алгоритма декомпозиции представим алгоритм случайного распределения ячеек расчетной сетки по доменам. Результат применения данного алгоритма к тестовой сетке показан на рис. 3. Данный алгоритм имеет достаточно низкое значение по критерию D , то есть распределение ячеек по доменам равномерное, однако при случайном распределении практически все ребра с высокой долей вероятности являются междоменными, что порождает огромный объем данных, которыми нужно обмениваться при синхронизации вычислений. Также следует отметить, что при случайном распределении ячеек по доменам любые два домена оказываются соседними, граф связности доменов яв-

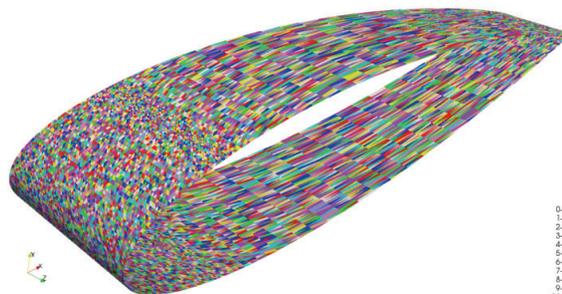
ляется полным. Таким образом, каждый домен должен обмениваться данными со всеми остальными доменами по достаточно протяженным границам. Также данный алгоритм становится совершенно непригодным, когда возникает необходимость создания в доменах буферных зон вблизи границ с глубиной больше единицы. Использование таких буферных зон может потребоваться для реализации численных методов повышенной точности.



Р и с. 2. Внешний вид расчетной сетки wing, используемой для тестирования алгоритмов декомпозиции

Fig. 2. Appearance of the wing computational mesh used for testing decomposition algorithms

На данном этапе уместно будет упомянуть механизм частичного сокрытия издержек на обмены данными за основными вычислениями. Данный механизм описан в [15]. Все ячейки каждого домена можно разделить на внутренние и граничные. Внутренние ячейки домена — ячейки, все соседи которых также принадлежат тому же домену. Граничные ячейки — это все остальные ячейки. Ценность внутренних ячеек состоит в том, что они не участвуют в обменах данными с соседними доменами (если не используются разностные схемы с повышенным порядком точности), поэтому обработку внутренних ячеек на следующей итерации расчетов можно начинать, не дожидаясь завершения обменов данными на ребрах, составляющих границу между доменами. Понятно, что данный подход неприменим при использовании случайного распределения ячеек по доменам, так как в большинстве случаев множества внутренних ячеек доменов оказываются практически пустыми.



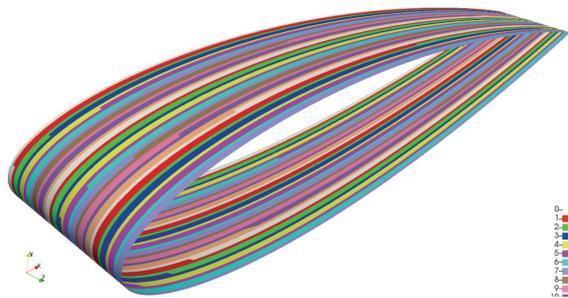
Р и с. 3. Результат декомпозиции расчетной сетки на 32 домена с помощью случайного распределения ячеек по доменам

Fig. 3. The result of decomposition of the computational grid into 32 domains using a random distribution of cells by domains

² NASA 0012 AIRFOILS (n0012-il) [Электронный ресурс] // Airfoil Tools. URL: <http://airfoiltools.com/airfoil/details?airfoil=n0012-il> (дата обращения 16.09.2020).



Рассмотрим семейство алгоритмов декомпозиции, в которых учитываются только индексы распределяемых ячеек и никак не учитываются другие их данные. Под индексом в данном случае понимается номер ячейки в общем массиве ячеек расчетной сетки. Уже по этому факту становится понятно, что такие алгоритмы не могут претендовать на высокое качество, так как их результат существенным образом зависит просто от порядка хранения данных расчетной сетки. Самым простым из алгоритмов данного класса является алгоритм линейного распределения ячеек по доменам. Так как в каждый домен в среднем должно войти по S/n ячеек (для простоты не будем обращать внимание на то, что это число может быть нецелым), то можно ячейки с номерами из диапазона $[0, S/n - 1]$ отнести к первому домену, ячейки с номерами $[S/n, 2S/n - 1]$ — ко второму домену и так далее. При такой декомпозиции, конечно, можно добиться равномерного распределения ячеек по доменам, однако значения параметров I и L в общем случае предугадать невозможно. Например, на рассматриваемой тестовой сетке видно, что деление на домены вдоль профиля крыла (как это показано на рис. 4) порождает очень длинные границы между соседними доменами. Было бы гораздо эффективнее производить деление сетки поперек профиля.



Р и с. 4. Результат декомпозиции расчетной сетки на 32 домена с помощью линейного разделения множества индексов ячеек
F i g. 4. Result of decomposition of the computational grid into 32 domains using linear division of the cell indices set

В общем случае ячейки с непрерывным диапазоном индексов не обязаны составлять не то что компактные домены с границами небольшой протяженности, а вообще могут порождать несвязные домены, с артефактами, выколотыми ячейками и так далее. Применение таких алгоритмов оправдано только при обладании некоторой априорной информацией о структуре расчетной сетки (например, если известно, что сетка на самом деле является блочно-структурированной, в которой отдельные блоки соответствуют непрерывным диапазонам индексов ячеек).

Большой класс алгоритмов формируется из так называемых алгоритмов наращивания доменов. В основе данных алгоритмов лежит следующий принцип: вначале выбирается ячейка (или несколько ячеек), от которой далее производится наращивание домена путем последовательного добавления к ней соседних ячеек. В рамках данного класса алгоритмов при последовательном создании доменов размера S/n получим жад-

ный алгоритм Фархата, результатом которого являются домены с очень протяженными границами [16]. При использовании одновременного роста сразу нескольких доменов от случайных ячеек расчетной сетки получаем алгоритм пузырькового роста [17], который требуется запускать на расчетной сетке итерационно несколько раз для получения приемлемых характеристик качества разбиения. При этом алгоритм пузырькового роста не гарантирует сбалансированного разбиения ячеек сетки по доменам (для достижения этого необходимо производить дополнительную коррекцию). Также следует отметить инкрементальный алгоритм декомпозиции графов [18], особенностью которого является возможность освобождения части ячеек, уже распределенных по доменам, с последующим повторением роста доменов.

Мы в данном случае рассматриваем простой алгоритм наращивания доменов, в котором все домены наращиваются одновременно, начиная с некоторых случайно выбранных ячеек сетки. При этом поддерживается связность доменов — если в какой-то момент домену больше некуда расти, то он прекращает свое наращивание и больше не расширяется. Таким образом, возможно генерация крайне неравномерных по количеству ячеек доменов. Пример результата применения описанного алгоритма приведен на рис. 5.



Р и с. 5. Результат декомпозиции расчетной сетки на 32 домена с помощью алгоритма наращивания доменов от случайных ячеек
F i g. 5 The result of decomposition of the computational grid into 32 domains using the algorithm for growing domains from random cells

Кроме приведенных алгоритмов существует еще большое количество подходов к декомпозиции расчетных сеток, среди них алгоритмы, основанные на методе спектральной бисекции [19], диффузионные и генетические алгоритмы [20], иерархические алгоритмы [21] и многие другие. Наиболее полный обзор различных алгоритмов декомпозиции расчетных сеток можно найти в [22]³ и [23]. Большинство из существующих алгоритмов декомпозиции либо не достаточно полно удовлетворяют критериям качества декомпозиции, либо должны выполняться итерационно, требуют локального уточнения и других корректирующих действий, что требует существенно вычислительного времени.

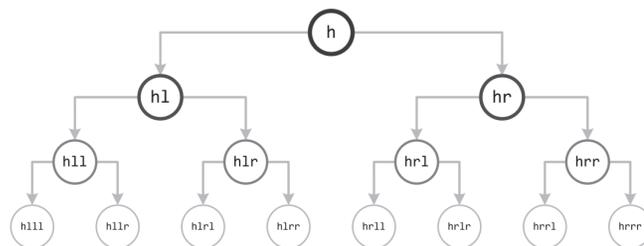
Возникает потребность создания простого быстрого алгоритма декомпозиции поверхностной неструктурированной расчетной сетки, в результате которого формируются связные домены с границами малой протяженности между ними.

³ Также детальный анализ алгоритмов декомпозиции расчетных сеток можно найти в диссертационной работе: Головаченко Е. Н. Декомпозиция расчетных сеток для решения задач механики сплошных сред на высокопроизводительных вычислительных системах: дисс. ... канд. физ.-мат. наук. -



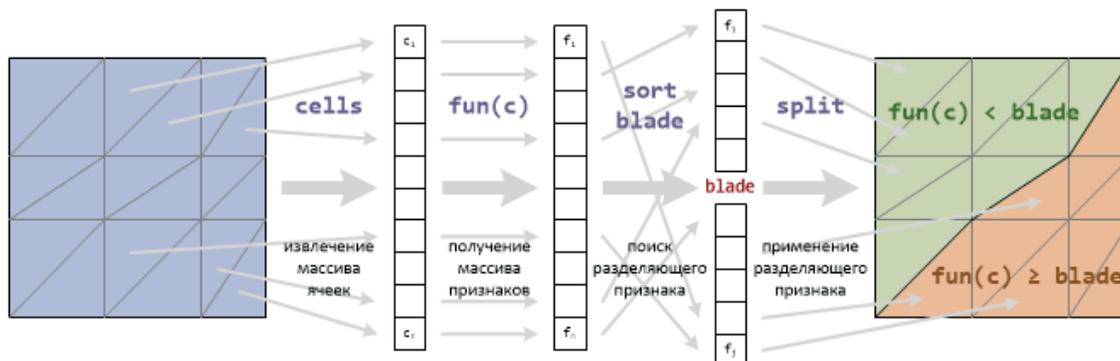
Иерархический метод декомпозиции с выбором наилучшего разбиения

В работе [22] описан параллельный алгоритм геометрической декомпозиции сеточных данных. Во время работы данного алгоритма происходит последовательное деление текущего домена пополам с помощью сечения плоскостью. На рис. 6. продемонстрирована схема, по которой изначальный головной домен h делится на пару доменов hl (left), hr (right), каждый из которых делится далее пополам и так далее на любое количество доменов, равное степени двойки.



Р и с. 6. Иллюстрация иерархического трехуровневого разделения головного домена

F i g. 6. Illustration of a hierarchical three-level division of the head domain



Р и с. 7. Схема выполнения разделения домена пополам по заданному признаку fun

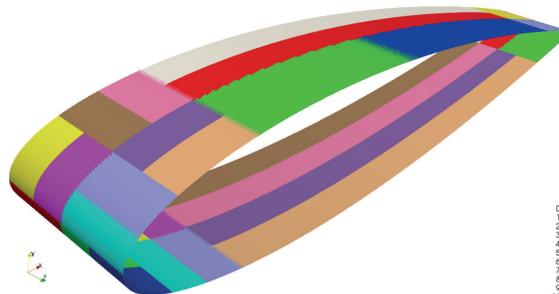
F i g. 7. Scheme for dividing a domain in half according to a given attribute fun

Данный алгоритм предлагается расширить, введя в него произвольные критерии разбиения текущего домена на пару более мелких доменов. Вначале рассмотрим схему простого деления домена пополам с использованием произвольного признака, по которому производится деление (см. рис. 7).

Будем описывать алгоритм разделения домена на две части в нотации языка Python. Пусть задан массив ячеек домена (h — head) и произвольная функция извлечения признака из ячейки (fun). Первым шагом является вычисление массива признаков для всех ячеек (s — signs): $s = \text{list}(\text{map}(\text{fun}, h))$. После получения массива признаков его нужно отсортировать: $s.\text{sort}()$. В отсортированном массиве признаков следует выбрать среднее значение (b — blade): $b = s[\text{len}(s) // 2]$. Данное значение будет использоваться для разделения домена на два мелких домена (hl — head left, hr — head right) с помощью применения двух простых фильтров: $hl, hr = \text{list}(\text{filter}(h, \text{lambda } x: \text{fun}(x) < b))$, $\text{list}(\text{filter}(h, \text{lambda } x: \text{fun}(x) >= b))$.

После разделения домена на два более мелких домена можно вычислить параметр, отражающий эффективность разбиения. В качестве такого параметра предлагается использовать длину границы между двумя образованными новыми доменами. Таким образом, критерий разбиения зависит от функции вычисления признака (fun). В свою очередь это означает, что при выполнении разбиения не обязательно ограничиваться одной функцией вычисления признака, вместо этого можно подать список функций, для каждой функции вычислитель показатель качества разбиения и в результате остановиться на той функции вычисления признака, которая в конечном итоге

приводит к наиболее эффективному разбиению. Если в качестве функций вычисления признака ячейки использовать просто извлечение трех координат центров ячеек, то мы получим в чистом виде алгоритм геометрической декомпозиции сетки с выбором для дробления наиболее протяженного размера по одной из координат. Результат применения данного алгоритма показан на рис. 8.



Р и с. 8. Результат декомпозиции расчетной сетки на 32 домена с помощью иерархического алгоритма

F i g. 8. The result of decomposition of the computational grid into 32 domains using a hierarchical algorithm

Варьируя набор функций вычисления признаков, по которым можно выполнять разбиение домена, возможно выполнять геометрическую декомпозицию вдоль направления любой кривой, для которой вычисляется проекция ячейки. Декомпозиция с помощью данного метода не ограничивается только ге-

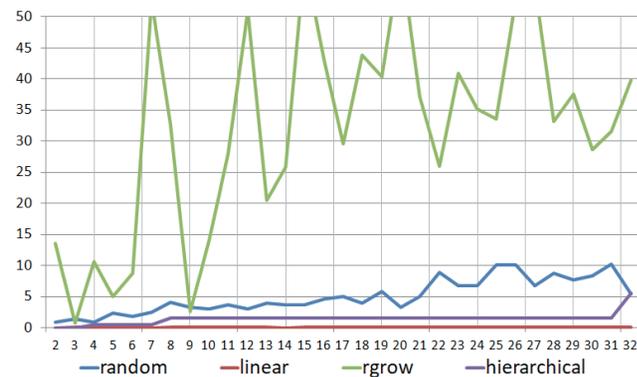


ометрическими признаками. Функции вычисления признаков могут использоваться для анализа физических данных ячеек, например, для локализации и выделения в отдельные домены областей с повышенным давлением.

В рамках данной работы описанный алгоритм декомпозиции поверхностной неструктурированной расчетной сетки применялся к поверхностным сеткам, используемым для расчета обледенения поверхности летательного аппарата⁴ [24], [25]. При расчете обледенения летательного аппарата основной объем вычислений относится к обработке поверхностных ячеек, общие данные между соседними доменами собраны на междоменных ребрах, что обеспечивает небольшой обмен данными, которыми требуется обмениваться в ходе синхронизации вычислений. Характерный размер таких сеток составил около 10^5 ячеек, рассматривались как односвязные, так и многосвязные поверхности, а также поверхности, состоящие из нескольких изолированных друг от друга зон.

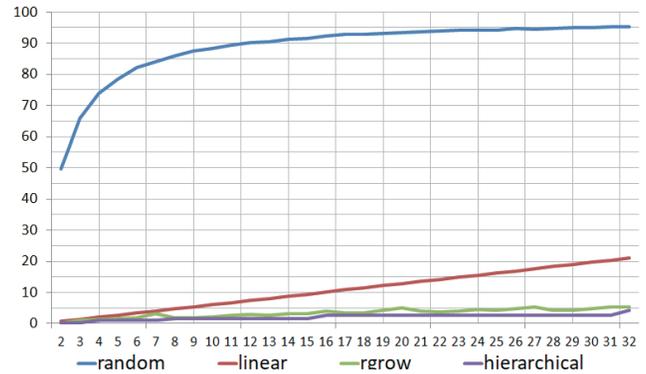
Результаты экспериментов⁵

На рисунках 9-11 представлены графики параметров D , I , L , вычисленные во время применения к тестовой сетке wing различных алгоритмов декомпозиции поверхностной расчетной сетки на разное количество доменов от 2 до 32. На данных графиках введены следующие краткие названия алгоритмов: random — случайное распределение ячеек по доменам, linear — линейное разделение ячеек между доменами по индексу, rgrow — алгоритм наращивания доменов от случайных ячеек, hierarchical — иерархическое деление доменов пополам по одной из трех координат (в качестве функций вычисления признаков ячеек брались просто функции извлечения каждой из трех координат центра ячейки).



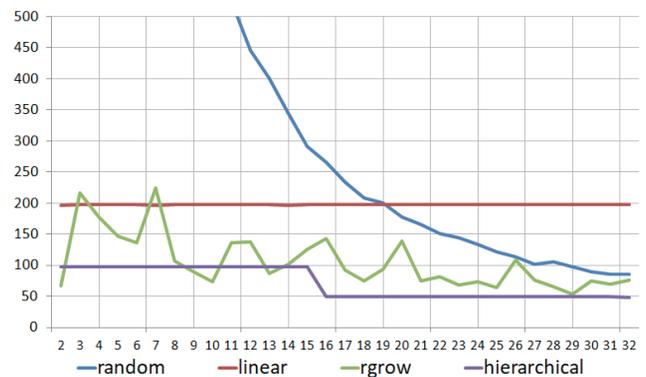
Р и с. 9. Графики отклонения максимального размера домена от оптимального значения (в процентах) для расчетной сетки wing при разных алгоритмах декомпозиции

Fig. 9. Graphs of the deviation of the maximum domain size from the optimal value (in percent) for the wing computational mesh with different decomposition



Р и с. 10. Графики доли (в процентах) количества ребер, являющихся границей между разными доменами, от общего количества ребер расчетной сетки wing при разных алгоритмах декомпозиции

Fig. 10. Graphs of the proportion (in percent) of the number of edges that are the boundary between different domains, of the total number of edges of the wing computational mesh for different decomposition algorithms



Р и с. 11. Графики максимальной длины границы между доменами при декомпозиции расчетной сетки wing с помощью разных алгоритмов декомпозиции

Fig. 11. Graphs of the maximum length of the boundary between domains when decomposing the computational mesh wing using different decomposition algorithms

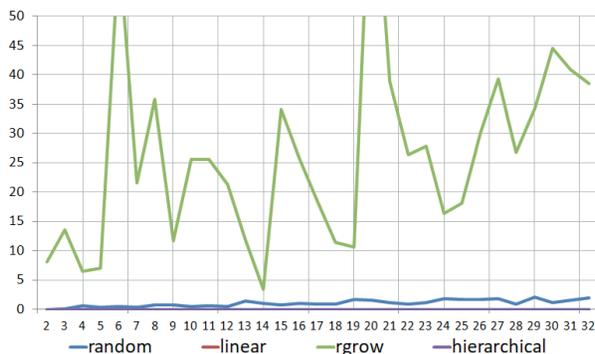
При этом отметим, что при использовании алгоритма rgrow умышленно использовалась только одна итерация наращивания, для демонстрации того, насколько неравномерным может быть распределение размеров доменов при случайном выборе стартовых ячеек (из рис. 9 видно, что отклонение может достигать 50% и более).

На рисунках 12-14 приведены аналогичные графики параметров D , I , L , полученные на сетке, более приближенной к реальной геометрии летательного аппарата (сетка ref). Данная сетка обладает следующими характеристиками: количество узлов — 228722, количество ребер — 685192, количество поверхностных ячеек — 456469.

⁴ Wright W. B., Struk P., Bartkus T., Addy G. Recent Advances in the LEWICE Icing Model [Электронный ресурс] // SAE 2015 International Conference on Icing of Aircraft, Engines, and Structures. SAE Technical Paper 2015-01-2094, 2015. DOI: <https://doi.org/10.4271/2015-01-2094>

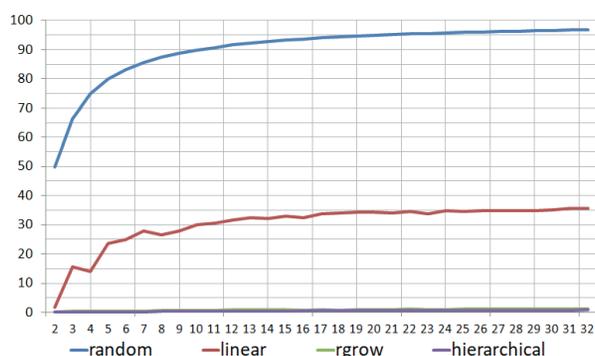
⁵ Реализация описанных в статье методов декомпозиции и тестовая расчетная сетка доступны в сети Интернет в репозитории <https://github.com/r-aax/crus-gsu/releases/tag/v1.0> (дата обращения 16.09.2020). Для повторения экспериментов, описанных в данном разделе, достаточно запустить на исполнение головной модуль репозитория src/main.py.





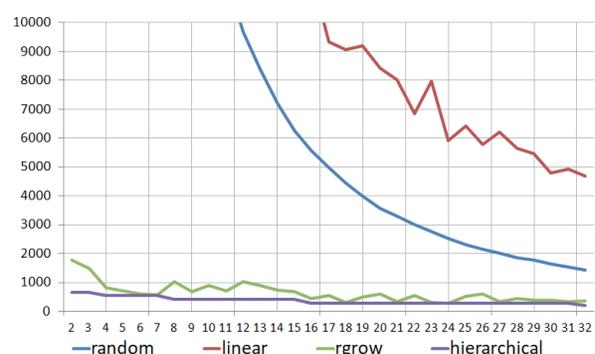
Р и с. 12. Графики отклонения максимального размера домена от оптимального значения (в процентах) для расчетной сетки ref при разных алгоритмах декомпозиции

Fig. 12. Graphs of deviation of the maximum domain size from the optimal value (in percent) for the computational grid ref for different decomposition algorithms



Р и с. 13. Графики доли (в процентах) количества ребер, являющихся границей между разными доменами, от общего количества ребер расчетной сетки ref при разных алгоритмах декомпозиции

Fig. 13. Graphs of the proportion (in percent) of the number of edges that are the boundary between different domains, of the total number of edges of the computational grid ref for different decomposition algorithms



Р и с. 14. Графики максимальной длины границы между доменами при декомпозиции расчетной сетки ref с помощью разных алгоритмов декомпозиции

Fig. 14. Graphs of the maximum length of the boundary between domains when decomposing the computational grid ref using different decomposition algorithms

Из приведенных графиков можно сделать выводы, что из представленных алгоритмов иерархическое деление доменов с выбором оптимального критерия деления из списка функций вычисления признаков ячеек является наиболее приемлемым по параметрам D , I , L , то есть с помощью данного алгоритма генерируется достаточно равномерное распределение ячеек по доменам при низкой общей доле междоменных ребер и малой протяженности границ между доменами. Отметим также, что наиболее затратной операцией с точки зрения асимптотики при разделении доменов является сортировка признаков ячеек, что с учетом иерархии дробления доменов приводит к сложности всего алгоритма не хуже $n \cdot \log^2 n$.

Заключение

В ходе выполнения работы рассмотрен ряд алгоритмов декомпозиции поверхностной неструктурированной расчетной сетки и проведен анализ эффективности этих алгоритмов по следующим параметрам: максимальное отклонение размера получившегося домена от идеального значения, доля междоменных ребер в итоговом разбиении, протяженность наиболее длинной границы между доменами. Предложен алгоритм иерархического разбиения доменов расчетной сетки с помощью выбора оптимального критерия разбиения на основе списка функций получения признаков ячеек. Полученные характеристики предложенного алгоритма, простота реализации и скорость его работы позволили использовать данный алгоритм в промышленных программных кодах для декомпозиции поверхностных расчетных сеток с целью улучшения масштабируемости суперкомпьютерных расчетов.

Список использованных источников

- [1] Minami, K. Supercomputers and Application Performance / K. Minami. – DOI 10.1007/978-981-13-9802-5_1 // The Art of High Performance Computing for Computational Science; M. Geshi (ed.). Springer, Singapore, 2019. – Vol. 2. – Pp. 1-9. – URL: https://link.springer.com/chapter/10.1007/978-981-13-9802-5_1 (дата обращения: 16.09.2020).
- [2] Schadt, E. Computational solutions to large-scale data management and analysis / E. Schadt, M. Linderman, J. Sorenson, L. Lee, G. P. Nolan. – DOI 10.1038/nrg2857 // Nature Reviews Genetics. – 2010. – Vol. 11. – Pp. 647-657.
- [3] Wright, N. J. Cori: A Pre-Exascale Supercomputer for Big Data and HPC Applications / N. J. Wright, S. S. Dosanjh, A. K. Andrews [et al.]. – DOI 10.3233/978-1-61499-583-8-82 // Advances in Parallel Computing. Big Data and High Performance Computing. – 2015. – Vol. 26. – Pp. 82-100. – URL: <https://ebooks.iospress.nl/publication/41174> (дата обращения: 16.09.2020).
- [4] Golovchenko, E. Numerical Experiments with New Algorithms for Parallel Decomposition of Large Computational Meshes / E. Golovchenko, E. Dorofeeva, I. Gasilova, A. Boldareva. – DOI 10.3233/978-1-61499-381-0-441 // Advances in Parallel Computing. Parallel Computing: Accelerating Computational Science and Engineering (CSE). – 2014. – Vol. 25. – Pp. 441-450. – URL: <https://ebooks.iospress.nl/publication/35910> (дата обращения: 16.09.2020).



- [5] Savin, G. I. RANS/ILES Method Optimization for Effective Calculations on Supercomputer / G. I. Savin, L. A. Bend-erskiy, D. A. Lyubimov, A. A. Rybakov. — DOI 10.1134/S1995080219050172 // Lobachevskii Journal of Math-ematics. — 2019. — Vol. 40, Issue 5. — Pp. 566-573. — URL: <https://link.springer.com/article/10.1134/S1995080219050172> (дата обращения: 16.09.2020).
- [6] Dorris, J. Task-Based Cholesky Decomposition on Knights Corner Using OpenMP / J. Dorris, J. Kurzak, P. Luszczek, A. YarKhan, J. Dongarra. — DOI 10.1007/978-3-319-46079-6_37 // High Performance Computing. ISC High Performance 2016. Lecture Notes in Computer Science; M. Tauber, V. Mohr, J. Kunkel (ed.). Springer, Cham. — 2016. — Vol. 9945. — Pp. 544-562. — URL: https://link.springer.com/chapter/10.1007/978-3-319-46079-6_37 (дата обращения: 16.09.2020).
- [7] Петров, М. Н. Многопоточная OpenMP-реализация ме-тода LU-SGS с использованием многоуровневой деком-позиции неструктурированной расчетной сетки / М. Н. Петров, В. А. Титарев, С. В. Утюжников, А. В. Чикиткин. — DOI 10.7868/S0044466917110138 // Журнал вы-числительной математики и математической физики. — 2017. — Т. 57, № 11. — С. 1895-1905. — URL: <https://www.elibrary.ru/item.asp?id=30480189> (дата обраще-ния: 16.09.2020). — Рез. англ.
- [8] Shabanov, V. M. Vectorization of High-performance Sci-entific Calculations Using AVX-512 Instruction Set / V. M. Shabanov, A. A. Rybakov, S. S. Shumilin. — DOI 10.1134/S1995080219050196 // Lobachevskii Journal of Mathemat-ics. — 2019. — Vol. 40, Issue 5. — Pp. 580-598. — URL: <https://link.springer.com/article/10.1134/S1995080219050196> (дата обращения: 16.09.2020).
- [9] Krzikalla, O. Dynamic SIMD Vector Lane Scheduling / O. Krzikalla, F. Wende, M. Höhnerbach. — DOI 10.1007/978-3-319-46079-6_25 // High Performance Computing. ISC High Performance 2016. Lecture Notes in Computer Science; M. Tauber, V. Mohr, J. Kunkel (ed.). Springer, Cham. — 2016. — Vol. 9945. — Pp. 354-365. — URL: https://link.springer.com/chapter/10.1007/978-3-319-46079-6_25 (дата обращения: 16.09.2020).
- [10] Chunduri, S. Characterization of MPI Usage on a Production Supercomputer / S. Chunduri, S. Parker, P. Balaji, K. Harms, K. Kumaran. — DOI 10.1109/SC.2018.00033 // SC18: International Conference for High Performance Computing, Networking, Storage and Analysis. — Dallas, TX, USA, 2018. — Pp. 386-400. — URL: [https://ieeexplore.ieee.org/docu-ment/8665758](https://ieeexplore.ieee.org/document/8665758) (дата обращения: 16.09.2020).
- [11] Ross, J. A. Parallel Programming Model for the Epiph-any Many-Core Coprocessor Using Threaded MPI / J. A. Ross, D. A. Richie, S. J. Park, D. R. Shires. — DOI 10.1145/2768177.2768183 // Proceedings of the 3rd Inter-national Workshop on Many-core Embedded Systems (MES '15). — Association for Computing Machinery, New York, NY, USA, 2015. — Pp. 41-47.
- [12] Klenk, B. An Overview of MPI Characteristics of Exas-cale Proxy Applications / B. Klenk, H. Fröning. — DOI 10.1007/978-3-319-58667-0_12 // High Performance Computing. ISC 2017. Lecture Notes in Computer Science; J. Kunkel, R. Yokota, P. Balaji, D. Keyes (ed.). Springer, Cham. — 2017. — Vol. 10266. — Pp. 217-236. — URL: https://link.springer.com/chapter/10.1007/978-3-319-58667-0_12 (дата обращения: 16.09.2020).
- [13] Бендерский, Л. А. Анализ эффективности масштабиро-вания при расчетах высокоскоростных турбулентных течений на суперкомпьютере RANS/ILES методом вы-сокого разрешения / Л. А. Бендерский, Д. А. Любимов, А. А. Рыбаков. — DOI 10.25682/NIISI.2018.4.9975 // Труды НИИСИ РАН. — 2017. — Т. 7, № 4. — С. 32-40. — URL: <https://www.elibrary.ru/item.asp?id=32294100> (дата обраще-ния: 16.09.2020). — Рез. англ.
- [14] Рыбаков, А. А. Распределение вычислительной нагруз-ки между узлами гетерогенного вычислительного кла-стера / А. А. Рыбаков. — DOI 10.15827/2311-6749.26.300 // Программные продукты, системы и алгоритмы. — 2018. — № 1. — С. 26-32. — URL: <https://www.elibrary.ru/item.asp?id=35560918> (дата обращения: 16.09.2020). — Рез. англ.
- [15] Рыбаков, А. А. Внутреннее представление и механизм межпроцессного обмена для блочно-структуриро-ванной сетки при выполнении расчетов на суперком-пьютере / А. А. Рыбаков. — DOI 10.25209/2079-3316-2017-8-1-121-134 // Программные системы: теория и приложения. — 2017. — Т. 8, № 1. — С. 121-134. — URL: <https://www.elibrary.ru/item.asp?id=28848236> (дата об-ращения: 16.09.2020). — Рез. англ.
- [16] Farhat, C. A simple and efficient automatic fem do-main decomposer / C. Farhat. — DOI 10.1016/0045-7949(88)90004-1 // Computers & Structures. — 1988. — Vol. 28, issue 5. — Pp. 579-602. — URL: <https://www.sciencedirect.com/science/article/pii/0045794988900041> (дата обращения: 16.09.2020).
- [17] Preis, R. PARTY - a Software Library for Graph Partitioning / R. Preis, R. Diekmann. — DOI 10.4203/ccp.45.3.1 // Ad-vances in Computational Mechanics for Parallel and Distrib-uted Processing; ed. by B.H.V. Topping. — Civil-Comp Press, Edinburgh, UK, 1997. — Pp. 63-71.
- [18] Якововский, М. В. Инкрементальный алгоритм деком-позиции графов / М. В. Якововский // Вестник Ниже-городского университета им. Н. И. Лобачевского. Се-рия «Математическое моделирование и оптимальное управление». — 2005. — № 1(28). — С. 243-250.
- [19] Urschel, J. C. Spectral bisection of graphs and connectedness / J. C. Urschel, L. T. Zikanov. — DOI 10.1016/j.laa.2014.02.007 // Linear Algebra and its Applications. — 2014. — Vol. 449. — Pp. 1-16. — URL: <https://www.sciencedirect.com/science/article/pii/S0024379514000743> (дата обращения: 16.09.2020).
- [20] Zhao, L. Automatic optimal block decomposition for structured mesh generation using genetic algorithm / L. Zhao, Y. Liu, C. Zhang, X. Zhang. — DOI 10.1007/s40430-018-1510-0 // Journal of the Brazilian Society of Mechanical Sciences and Engineering. — 2019. — Vol. 41, Article 10. — URL: <https://link.springer.com/article/10.1007%2Fs40430-018-1510-0> (дата обращения: 16.09.2020).
- [21] Karypis, G. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs / G. Karypis, V. Kumar. —



- DOI 10.1137/S1064827595287997 // *SIAM Journal on Scientific Computing*. — 1998. — Vol. 20, issue 1. — Pp. 359-392. — URL: <https://epubs.siam.org/doi/abs/10.1137/S1064827595287997> (дата обращения: 16.09.2020).
- [22] Головченко, Е. Н. Обзор алгоритмов декомпозиции графов / Е. Н. Головченко. — DOI 10.20948/prepr-2020-2 // Препринты ИПМ им. М. В. Келдыша. — 2020. — № 2. — С. 1-38. — URL: <https://www.elibrary.ru/item.asp?id=41670303> (дата обращения: 16.09.2020). — Рез. англ.
- [23] Железнякова, А. Л. Эффективные методы декомпозиции неструктурированных адаптивных сеток для высокопроизводительных расчетов при решении задач вычислительной аэродинамики / А. Л. Железнякова // Физико-химическая кинетика в газовой динамике. — 2017. — Т. 18, № 1. — С. 1-20. — URL: <https://www.elibrary.ru/item.asp?id=35136792> (дата обращения: 16.09.2020). — Рез. англ.
- [24] Li, S. Fast Evaluation of Aircraft Icing Severity Using Machine Learning Based on XGBoost / S. Li, J. Qin, M. He, R. Paoli. — DOI 10.3390/aerospace7040036 // *Aerospace*. — 2020. — Vol. 7, issue 4. — Pp. 36. — URL: <https://www.mdpi.com/2226-4310/7/4/36> (дата обращения: 16.09.2020).
- [25] Bourgault-Côté, S. Multi-Layer Icing Methodologies for Conservative Ice Growth / S. Bourgault-Côté, K. Hasanzadeh, P. Lavoie, E. Laurendeau. — DOI 10.13009/EUCASS2017-258 // 7th European Conference for Aeronautics and Aerospace Sciences (EUCASS). — 2017. — URL: <https://www.eucass.eu/doi/EUCASS2017-258.pdf> (дата обращения: 16.09.2020).

Поступила 16.09.2020; одобрена после рецензирования
23.10.2020; принята к публикации 06.12.2020.

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» (119334, Российская Федерация, г. Москва, Ленинский пр., д. 32а), кандидат физико-математических наук, ORCID: <http://orcid.org/0000-0002-9755-8830>, rybakov@jssc.ru

Чопорняк Антон Дмитриевич, старший научный сотрудник Межведомственного суперкомпьютерного центра Российской академии наук – филиала Федерального государственного учреждения «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук» (119334, Российская Федерация, г. Москва, Ленинский пр., д. 32а), ORCID: <http://orcid.org/0000-0001-6617-5303>, adc@jssc.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

References

- [1] Minami K. Supercomputers and Application Performance. In: Geshi M. (ed.) *The Art of High Performance Computing for Computational Science*, Vol. 2. Springer, Singapore; 2019. p. 1-9. (In Eng.) DOI: https://doi.org/10.1007/978-981-13-9802-5_1
- [2] Schadt E., Linderman M., Sorenson J., Lee L., Nolan G.P. Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics*. 2010; 11:647-657. (In Eng.) DOI: <https://doi.org/10.1038/nrg2857>
- [3] Wright N.J., Dosanjh S.S., Andrews A.K. et al. Cori: A Pre-Exascale Supercomputer for Big Data and HPC Applications. In: *Advances in Parallel Computing. Big Data and High Performance Computing*. 2015; 26:82-100. (In Eng.) DOI: <https://doi.org/10.3233/978-1-61499-583-8-82>
- [4] Golovchenko E., Dorofeeva E., Gasilova I., Boldareva A. Numerical Experiments with New Algorithms for Parallel Decomposition of Large Computational Meshes. In: *Advances in Parallel Computing. Parallel Computing: Accelerating Computational Science and Engineering (CSE)*. 2014; 25:441-450. (In Eng.) DOI: <https://doi.org/10.3233/978-1-61499-381-0-441>
- [5] Savin G.I., Benderskiy L.A., Lyubimov D.A., Rybakov A.A. RANS/ILES Method Optimization for Effective Calculations on Supercomputer. *Lobachevskii Journal of Mathematics*. 2019; 40(5):566-573. (In Eng.) DOI: <https://doi.org/10.1134/S1995080219050172>
- [6] Dorris J., Kurzak J., Luszczek P., YarKhan A., Dongarra J. Task-Based Cholesky Decomposition on Knights Corner Using OpenMP. In: Taufer M., Mohr B., Kunkel J. (ed.) *High Performance Computing. ISC High Performance 2016. Lecture Notes in Computer Science*. 2016; 9945:544-562. Springer, Cham. (In Eng.) DOI: https://doi.org/10.1007/978-3-319-46079-6_37
- [7] Petrov M.N., Titarev V.A., Utyuzhnikov S.V., Chikitkin A.V. A multithreaded OpenMP implementation of the LU-SGS method using the multilevel decomposition of the unstructured computational mesh. *Computational Mathematics and Mathematical Physics*. 2017; 57(11):1895-1905. (In Eng.) DOI: <https://doi.org/10.1134/S0965542517110124>
- [8] Shabanov B.M., Rybakov A.A., Shumilin S.S. Vectorization of High-performance Scientific Calculations Using AVX-512 Instruction Set. *Lobachevskii Journal of Mathematics*. 2019; 40(5):580-598. (In Eng.) DOI: <https://doi.org/10.1134/S1995080219050196>
- [9] Krzikalla O., Wende F., Höhnerbach M. Dynamic SIMD Vector Lane Scheduling. In: Taufer M., Mohr B., Kunkel J. (eds) *High Performance Computing. ISC High Performance 2016. Lecture Notes in Computer Science*. 2016; 9945:354-365. Springer, Cham. (In Eng.) DOI: https://doi.org/10.1007/978-3-319-46079-6_25
- [10] Chunduri S., Parker S., Balaji P., Harms K., Kumaran K. Characterization of MPI Usage on a Production Supercomputer. In: *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*. Dallas, TX, USA; 2018. p. 386-400. (In Eng.) DOI: <https://doi.org/10.1109/SC.2018.00033>



- [11] Ross J.A., Richie D.A., Park S.J., Shires D.R. Parallel Programming Model for the Epiphany Many-Core Coprocessor Using Threaded MPI. In: *Proceedings of the 3rd International Workshop on Many-core Embedded Systems (MES '15)*. Association for Computing Machinery, New York, NY, USA; 2015. p. 41-47. (In Eng.) DOI: <https://doi.org/10.1145/2768177.2768183>
- [12] Klenk B., Fröning H. An Overview of MPI Characteristics of Exascale Proxy Applications. In: Kunkel J., Yokota R., Balaji P., Keyes D. (ed.) *High Performance Computing. ISC 2017. Lecture Notes in Computer Science*. 2017; 10266:217-236. Springer, Cham. (In Eng.) DOI: https://doi.org/10.1007/978-3-319-58667-0_12
- [13] Benderskiy L.A., Lyubimov D.A., Rybakov A.A. Scaling of fluid dynamic calculations using the RANS/ILES method on supercomputer. *Proceedings of NIISI RAS*. 2017; 7(4):32-40. (In Russ., abstract in Eng.) DOI: <https://doi.org/10.25682/NIISI.2018.4.9975>
- [14] Rybakov A.A. *Raspredelenie vychislitel'noj nagruzki mezhdu uzlami heterogennogo vychislitel'nogo klastera* [Distribution of computational load between nodes of a heterogeneous computational cluster]. *Software Journal: Theory and Applications*. 2018; (1):26-32. (In Russ., abstract in Eng.) DOI: <https://doi.org/10.15827/2311-6749.26.300>
- [15] Rybakov A. A. Inner representation and crossprocess exchange mechanism for block-structured Grid for supercomputer calculations. *Program Systems: Theory and Applications*. 2017; 8(1):121-134. (In Russ., abstract in Eng.) DOI: <https://doi.org/10.25209/2079-3316-2017-8-1-121-134>
- [16] Farhat C. A simple and efficient automatic fem domain decomposer. *Computers & Structures*. 1988; 28(5):579-602. (In Eng.) DOI: [https://doi.org/10.1016/0045-7949\(88\)90004-1](https://doi.org/10.1016/0045-7949(88)90004-1)
- [17] Preis R., Diekmann R. PARTY - a Software Library for Graph Partitioning. In: Topping B.H.V. (ed.) *Advances in Computational Mechanics for Parallel and Distributed Processing*. Civil-Comp Press, Edinburgh, UK; 1997. p. 63-71. (In Eng.) DOI: <https://doi.org/10.4203/ccp.45.3.1>
- [18] Yakobovskii M.V. *Inkremental'nyj algoritm dekompozicii grafov* [An incremental algorithm for graph decomposition]. *Vestn. Nizhegorod. Univ. im. N.I. Lobachevskogo. Ser. Mat. Model. Optim. Upr.* 2005; (1):243-250. (In Russ.)
- [19] Urschel J.C., Zikatanov L.T. Spectral bisection of graphs and connectedness. *Linear Algebra and its Applications*. 2014; 449:1-16. (In Eng.) DOI: <https://doi.org/10.1016/j.laa.2014.02.007>
- [20] Zhao L., Liu Y., Zhang C., Zhang X. Automatic optimal block decomposition for structured mesh generation using genetic algorithm. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*. 2019; 41(1):10. (In Eng.) DOI: <https://doi.org/10.1007/s40430-018-1510-0>
- [21] Karypis G., Kumar V. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*. 1998; 20(1):359-392. (In Eng.) DOI: <https://doi.org/10.1137/S1064827595287997>
- [22] Golovchenko E.N. *Obzor algoritmov dekompozicii grafov* [Survey of graph partitioning algorithms]. *Keldysh Institute PREPRINTS*. 2020; (2):1-38. (In Russ., abstract in Eng.) DOI: <https://doi.org/10.20948/prepr-2020-2>
- [23] Zheleznyakova A.L. Load Balancing Methods for Molecular Dynamics Applications. *Physical-Chemical Kinetics in Gas Dynamics*. 2017; 18(1):1-20. Available at: <https://www.elibrary.ru/item.asp?id=35136792> (accessed 16.09.2020). (In Russ., abstract in Eng.)
- [24] Li S., Qin J., He M., Paoli R. Fast Evaluation of Aircraft Icing Severity Using Machine Learning Based on XGBoost. *Aerospace*. 2020; 7(4):36. (In Eng.) DOI: <https://doi.org/10.3390/aerospace7040036>
- [25] Bourgault-Côté S., Hasanzadeh K., Lavoie P., Laurendeau E. Multi-Layer Icing Methodologies for Conservative Ice Growth. In: *7th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*. 2017. (In Eng.) DOI: <https://doi.org/10.13009/EUCASS2017-258>

Submitted 16.09.2020; approved after reviewing 23.10.2020;
accepted for publication 06.12.2020.

About the authors:

Alexey A. Rybakov, Senior Research Fellow of the Joint Supercomputer Center of the Russian Academy of Sciences - Branch of Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (32a Leninsky prospect, Moscow 119334, Russian Federation), Ph.D. (Phys.-Math.), ORCID: <http://orcid.org/0000-0002-9755-8830>, rybakov@jssc.ru

Anton D. Chopornyak, Senior Researcher of the Joint Supercomputer Center of the Russian Academy of Sciences - Branch of Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (32a Leninsky prospect, Moscow 119334, Russian Federation), ORCID: <http://orcid.org/0000-0001-6617-5303>, adc@jssc.ru

All authors have read and approved the final manuscript.

