

Сглаживание границ между доменами поверхностной расчетной сетки

А. Д. Багров, А. А. Рыбаков*

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

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

* rybakov@jssc.ru

Аннотация

Для эффективного использования суперкомпьютерных вычислительных ресурсов при решении задач имитационного моделирования важное значение имеет качественная декомпозиция расчетной сетки. Современные расчетные задачи работают с входными данными большого объема, из-за чего обработка их на одном вычислительном узле за разумное время невозможна. В частности, для проведения расчетов на больших расчетных сетках требуется выполнение их декомпозиции на отдельные домены. При этом каждый из доменов обрабатывается на своем вычислительном узле, а для поддержания целостности задачи вычислительные узлы обмениваются между собой расчетными данными, пересекающими границы доменов расчетной сетки. Качество декомпозиции расчетной сетки характеризуется такими параметрами, как равномерность распределения ячеек по доменам, максимальная длина границы между двумя доменами, суммарная длина всех границ между всеми парами доменов и многими другими. Важнейшим индикатором качества декомпозиции сетки является равномерность распределения ячеек по доменам, так как скорость обсчета всей задачи характеризуется прежде всего скоростью обработки самого крупного домена. Однако, также высокую значимость имеет максимальная длина границы между парами доменов, так как значение данного параметра определяет скорость выполняемых обменов данными между вычислительными узлами. При уменьшении шага по времени в расчетных задачах и увеличении количества вычислительных узлов (высокая степень распараллеливания), доля межпроцессных обменов существенно возрастает, и уменьшение длины границы между доменами существенным образом влияет на общее время работы расчетной задачи. В данной статье рассматривается алгоритм сглаживания границ между доменами, в процессе которого длины границ уменьшаются, при этом не ухудшая равномерность распределения ячеек сетки между доменами. Численные эксперименты показали, что применение предложенного алгоритма приводит к сокращению длин границ между доменами примерно 10%, что приводит к ускорению обмена данными в процессе проведения расчетов.

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

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.

Авторы заявляют об отсутствии конфликта интересов.



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



Для цитирования: Багров, А. Д. Сглаживание границ между доменами поверхностной расчетной сетки / А. Д. Багров, А. А. Рыбаков. — DOI 10.25559/SITITO.17.202102.265-274 // Современные информационные технологии и ИТ-образование. — 2021. — Т. 17, № 2. — С. 265-274.

© Багров А. Д., Рыбаков А. А., 2021



Smoothing the Boundaries between the Domains of the Surface Computational Mesh

A. D. Bagrov, A. A. Rybakov*

Scientific Research Institute for System Analysis of the Russian Academy of Sciences, Moscow, Russian Federation

Nakhimovsky prospect, 36, building 1, Moscow 117218, Russian Federation

* rybakov@jssc.ru

Abstract

For the efficient use of supercomputer computing resources in solving problems of simulation, a high-quality decomposition of the computational meshes is of great importance. Modern computational problems work with huge input data, which makes it impossible to process them on one computational node in a reasonable time. In particular, to carry out calculations on large computational meshes, it is necessary to decompose them into separate domains. In this case, each of the domains is processed at its own computational node, and to maintain the integrity of the problem, computational nodes exchange between themselves computational data that cross the boundaries of the computational mesh domains. The quality of the computational mesh decomposition is characterized by such parameters as the uniformity of the distribution of cells over domains, the maximum length of the boundary between two domains, the total length of all boundaries between all pairs of domains, and many others. The most important indicator of the mesh decomposition quality is the uniformity of the distribution of cells across domains, since the speed of computation of the entire problem is characterized primarily by the speed of processing the largest domain. However, the maximum length of the boundary between pairs of domains is also of high importance, since the value of this parameter determines the speed of data exchanges between computational nodes. With a decrease in the time step in computational problems and an increase in the number of computational nodes (a high degree of parallelization), the proportion of interprocess communications increases significantly, and a decrease in the length of the boundary between domains significantly affects the total time of the computational problem. This article discusses an algorithm for smoothing the boundaries between domains, in the process of which the lengths of the boundaries are reduced, while not impairing the uniformity of the distribution of mesh cells between domains. Numerical experiments have shown that the application of the proposed algorithm leads to a reduction in the lengths of the boundaries between domains by about 10%, which leads to acceleration of data exchange in the course of calculations.

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

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 OИ supercomputer located in the Interdepartmental Super-computer Center of the Russian Academy of Sciences.

The authors declare no conflict of interest.

For citation: Bagrov A.D., Rybakov A.A. Smoothing the Boundaries between the Domains of the Surface Computational Mesh. *Sovremennye informacionnye tehnologii i IT-obrazovanie = Modern Information Technologies and IT-Education*. 2021; 17(2):265-274. DOI: <https://doi.org/10.25559/SITI-TO.17.202102.265-274>



Введение

Для повышения эффективности суперкомпьютерных вычислений используют различные техники, направленные, в конечном итоге, на повышение степени параллельности проведения расчетов [1-3]. Это распределение вычислений между узлами суперкомпьютерного кластера с помощью декомпозиции расчетных сеток на отдельные домены [4-6], организация многопоточных вычислений при работе в рамках одного узла с общей памятью [7-9], повышение эффективности расчетного кода путем выполнения низкоуровневых оптимизаций [10-12], удаления избыточных зависимостей и применения векторизации кода [13-15]. В данной статье рассматривается только проблема декомпозиции расчетной сетки для эффективного распределения вычислительной нагрузки между узлами гомогенного кластера.

Рассмотрим неструктурированную расчетную сетку, состоящую из S ячеек, пусть также суперкомпьютер состоит из n вычислительных узлов с одинаковыми характеристиками. Если представить скорость обработки расчетных ячеек на одном вычислительном узле как a^{-1} , то время выполнения одной итерации расчетов на одном узле будет равно $T_1 = aS$. Пусть 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 . Предположим, что каждый домен обрабатывается в своем вычислительном узле. Таким образом, все домены обрабатываются параллельно и независимо друг от друга, и время обработки всех доменов определяется временем обработки самого крупного из них, то есть $a \max_{i=1, n} S_i$. Далее предположим, что скорость обмена данными между любой парой узлов определяется величиной b^{-1} . После обчета всех доменов выполняется параллельный обмен данными между теми парами, которые граничат между собой. Исходя из этого, можно предположить что время выполнения всех обменов определяется длиной самой протяженной границы между двумя доменами и равняется $b \max_{i, j=1, n} L_{ij}$. Таким образом, суммарное время расчета одной итерации на n вычислительных узлах можно выразить формулой $T_n = a \max_{i=1, n} S_i + b \max_{i, j=1, n} L_{ij}$. Из приведенной формулы видно, что время обработки одной итерации на n вычислительных узлах тем ниже, чем меньше значения величин $\max_{i=1, n} S_i$ (размер максимального домена) и $\max_{i, j=1, n} L_{ij}$ (длина максимальной границы между доменами). Вместо абсолютного размера максимального домена удобнее рассматривать характеристику, отражающую равномерность распределения ячеек между доменами, которая вычисляется по следующей формуле

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

Из приведенной формулы видно, что распределение ячеек между доменами является идеальным если выполняется условие $D = 0$. Вторую же характеристику оставим без изменений и будем использовать в абсолютном значении $L = \max_{i, j=1, n} L_{ij}$. Как правило, основное время расчетов занимает именно обработка данных внутри домена, тогда как время обмена данными имеет второстепенное значение. Таким образом, параметр D более важен, чем L . Поэтому при проведении расчетов будем пользоваться методами декомпозиции

расчетной сетки, при которых выполняется $D = 0$ (возможно с погрешностью в одну ячейку).

Существует достаточно большое количество алгоритмов декомпозиции неструктурированных расчетных сеток [16-19]. Отметим из них только те, которые характеризуются показателем $D = 0$. Во-первых, к таким алгоритмам относится простейший алгоритм линейного распределения ячеек по доменам. В нем фигурируют только индексы ячеек и полностью игнорируются геометрические особенности расположения данных ячеек. В данном алгоритме массив ячеек разбивается на равные подмассивы длины S/n , каждый из которых соотносится со своим доменом. Конечно, при таком подходе получается идеально равномерное распределение ячеек по доменам, однако границы между доменами получаются абсолютно непредсказуемыми, что приводит к непригодности использования данного алгоритма.

Большую группу алгоритмов представляют так называемые алгоритмы наращивания доменов. В основе данных алгоритмов лежит следующий принцип: вначале выбирается ячейка (или несколько ячеек), от которой производится наращивание домена путем последовательного добавления соседних ячеек. В результате формируются зачастую связанные домены (связность не гарантируется, но некоторые домены могут обладать этим свойством). Одним из самых привычных алгоритмов наращивания доменов является простой алгоритм Фархата [20]. Для применения алгоритма Фархата используется граф смежности ячеек расчетной сетки (узлами данного графа являются ячейки сетки, а ребрами — факт смежности двух ячеек сетки по линейной границе), далее строится путь обхода данного графа в ширину от произвольной ячейки сетки, после чего список обхода, подобно применению алгоритму линейного распределения, разбивается на равные подписки длиной S/n , каждый из которых порождает связный домен. Существуют другие алгоритмы наращивания доменов, в частности алгоритм пузырькового роста доменов [21] или инкрементальный алгоритм декомпозиции графа [22], однако для них не гарантируется выполнение условия $D = 0$.

Отдельно отметим иерархический алгоритм декомпозиции графа, который предназначен для разбиения расчетной сетки на количество доменов, представленное степени двойки. На первом шаге алгоритма вся сетка представлена только одним доменом, содержащим все ячейки расчетной сетки. На каждом последующем шаге декомпозиции каждый домен делится пополам согласно определенному правилу. Наиболее простым способом разделения домена на два является сечение некоторой плоскостью (в простейшем варианте в качестве секущих плоскостей могут выступать плоскости OXY, OXZ, OYZ) [23]. Данный алгоритм также порождает декомпозицию, удовлетворяющую условию $D = 0$, однако он обладает существенным недостатком — не может быть использован для декомпозиции сетки на произвольное количество доменов (допустимы только значения, являющиеся степенью двойки).

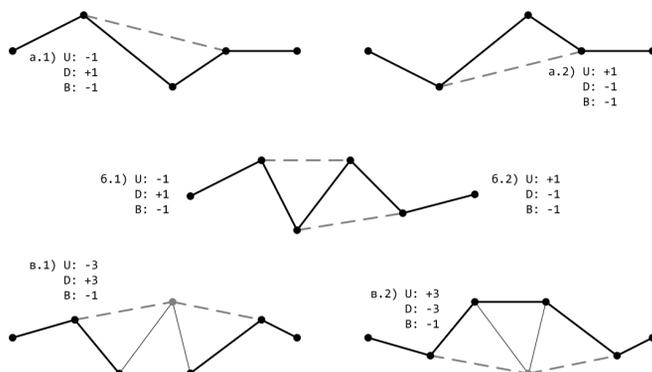
Исходя из написанного выше для изучения алгоритма сглаживания границ между доменами будем рассматривать только два алгоритма декомпозиции: алгоритм Фархата и иерархический алгоритм пространственной декомпозиции.



Алгоритм сглаживания границ между доменами

Алгоритм сглаживания носит локальный характер, он применяется последовательно к каждой паре доменов и направлен на уменьшение длины границы между ними с сохранением баланса количества ячеек в этих доменах. Граница между двумя доменами может быть представлена в виде набора простых циклов и простых цепей. При этом простой цикл может быть обработан таким же образом, как и простая цепь, с учетом совпадения первого и последнего узла этой цепи (для такого виртуального размыкания простого цикла может быть выбран произвольный узел этого цикла). В процессе применения алгоритма сглаживания независимо обрабатывается каждая цепь.

Вначале одним линейным проходом по цепи выполняется поиск всех пригодных для оптимизации границы шаблонов, которые представлены на рис. 1.



Р и с. 1. Шаблоны элементарных действий сглаживания границ между доменами

Fig. 1. Basic Domain Border Smoothing Templates

На рис. 1 представлены 5 шаблонов, которые могут быть использованы для оптимизации длины границы между двумя доменами. Рассмотрим шаблон а.1. На нем обозначена часть границы между двумя доменами (домен сверху от ломаной обозначен буквой U, домен снизу от ломаной обозначен буквой D). Черной сплошной линией прочерчена текущая граница между доменами. Пунктирной линией показано возможное улучшение, которое в данном случае приведет к следующим последствиям: уменьшение количества ячеек верхнего домена на 1 (U: -1), увеличение количества ячеек нижнего домена на 1 (D: +1), уменьшение длины границы между двумя доменами на 1 (B: -1). Таким образом, шаблон а.1 направлен на втягивание одной ячейки из верхнего домена в нижний домен с сокращением длины границы на единицу. Шаблон а.2 выполняет симметричное действие по втягиванию одной ячейки из нижнего домена в верхний домен, также сокращая при этом длину границы на единицу. Шаблоны в.1 и в.2 выполняют аналогичные действия, однако вместо одной ячейки происходит втягивание сразу трех соседних ячеек с одновременным уменьшением длины границы между доменами на единицу. Отдельно стоит отметить шаблоны б.1 и б.2. Они представляют собой один шаблон в рамках которого может быть осу-

ществлено втягивание либо ячейки из нижнего домена в верхний, либо наоборот, но не одновременно (в противном случае длина границы останется неизменной). Можно рассматривать и более сложные шаблоны, однако существенного прироста производительности они уже не дают, но приводят к усложнению алгоритма.

После того, как внутри цепи найдены все шаблоны, потенциально пригодные для оптимизации границы, выполняется разметка того, как данные шаблоны могут влиять на другие шаблоны. С учетом того, что любой шаблон может повлиять только на своих непосредственных соседей, данное действие также выполняется с линейной сложностью относительно длины границы.

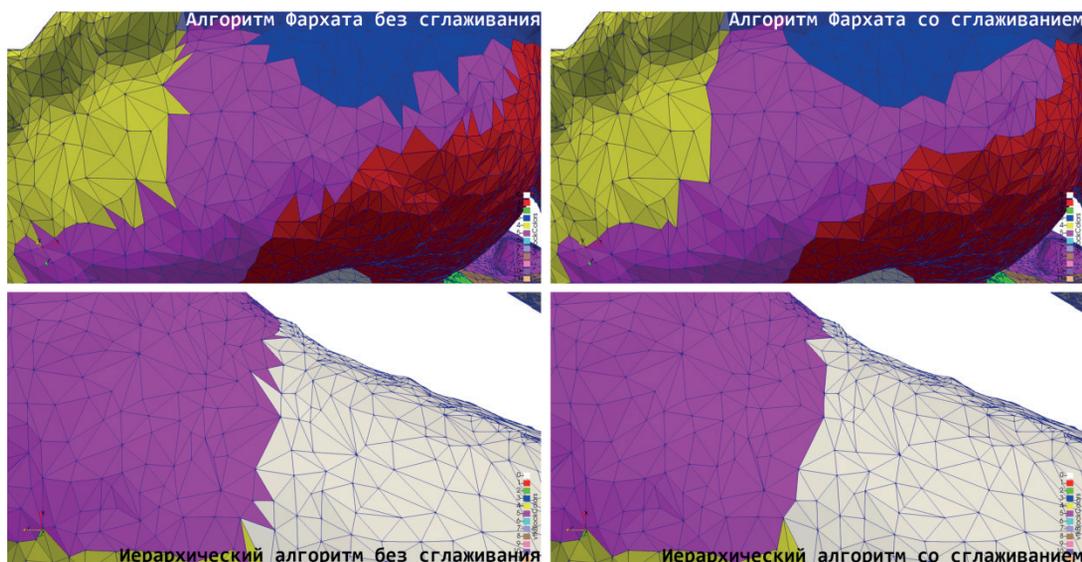
Последним шагом применения алгоритма является выбор такого множества шаблонов, которые не влияют друг на друга (то есть могут быть применены все одновременно) и не нарушают суммарного баланса ячеек (так как для нас важнейшим показателем эффективности декомпозиции расчетной сетки является равномерность распределения ячеек по доменам). После выбора наибольшего возможного набора шаблонов они применяются, и на этом обработка данной цепи считается завершенной.



Р и с. 2. Визуализация декомпозиции тестовой расчетной сетки dragon на 32 домена с помощью алгоритма Фархата (сверху) и иерархического алгоритма (снизу)

Fig. 2. Visualization of the decomposition of the dragon test computational grid into 32 domains using the Farhat algorithm (top) and the hierarchical algorithm (bottom)





Р и с. 3. Визуализация применения сглаживания границ между доменами после работы алгоритма Фархата (сверху) и иерархического алгоритма (снизу)
Fig. 3. Visualization of applying smoothing of boundaries between domains after running the Farhat algorithm (top) and the hierarchical algorithm (bottom)

На рис. 2 представлена визуализация декомпозиции тестовой расчетной сетки dragon с помощью алгоритма Фархата и иерархического алгоритма бинарной декомпозиции. В обоих случаях декомпозиция выполняется на 32 домена. При этом на рис. 2 представлены результаты работы с уже примененным сглаживанием границ между доменами.

На рис. 3 крупным планом продемонстрированы отдельные части тестовой расчетной сетки dragon с отображением ребер ячеек. На данном рисунке виден эффект от применения алгоритма сглаживания границ между доменами, прежде всего от заключается в устранении одиноких ячеек, которые вторгаются в соседний домен одной своей вершиной. После применения алгоритма границы между доменами визуально выглядят более гладко, их длина уменьшается.

Результаты экспериментов

Для тестирования эффективности алгоритма сглаживания границ между доменами использовались тестовые неструктурированные поверхностные сетки bunny, dragon, luse, к которым были применены алгоритмы декомпозиции с показателем $D = 0$ и сравнены показатели качества декомпозиции до и после сглаживания границ. Результаты проведения экспериментов представлены на рис. 4.

Неструктурированные поверхностные расчетные сетки bunny (количество ячеек $5 \cdot 10^3$), dragon (количество ячеек 10^5), luse (количество ячеек 10^5) были взяты из открытых источников¹. Из приведенных на рис. 4 данных видно, что в целом проиллюстрированные показатели эффективности декомпозиции расчетных сеток (процентная доля ребер сетки, являющихся граничными ребрами между доменами, и максимальная

длина границы между доменами) лучше для иерархического пространственного алгоритма декомпозиции. Применение же алгоритма сглаживания границ приводит к сокращению как общего количества граничных ребер (кросс-ребер), так и длины максимальной границы примерно на 10%. Данный эффект приводит к снижению времени межпроцессных обменов для задач, выполняющих расчеты на поверхностных расчетных сетках² [24], [25].

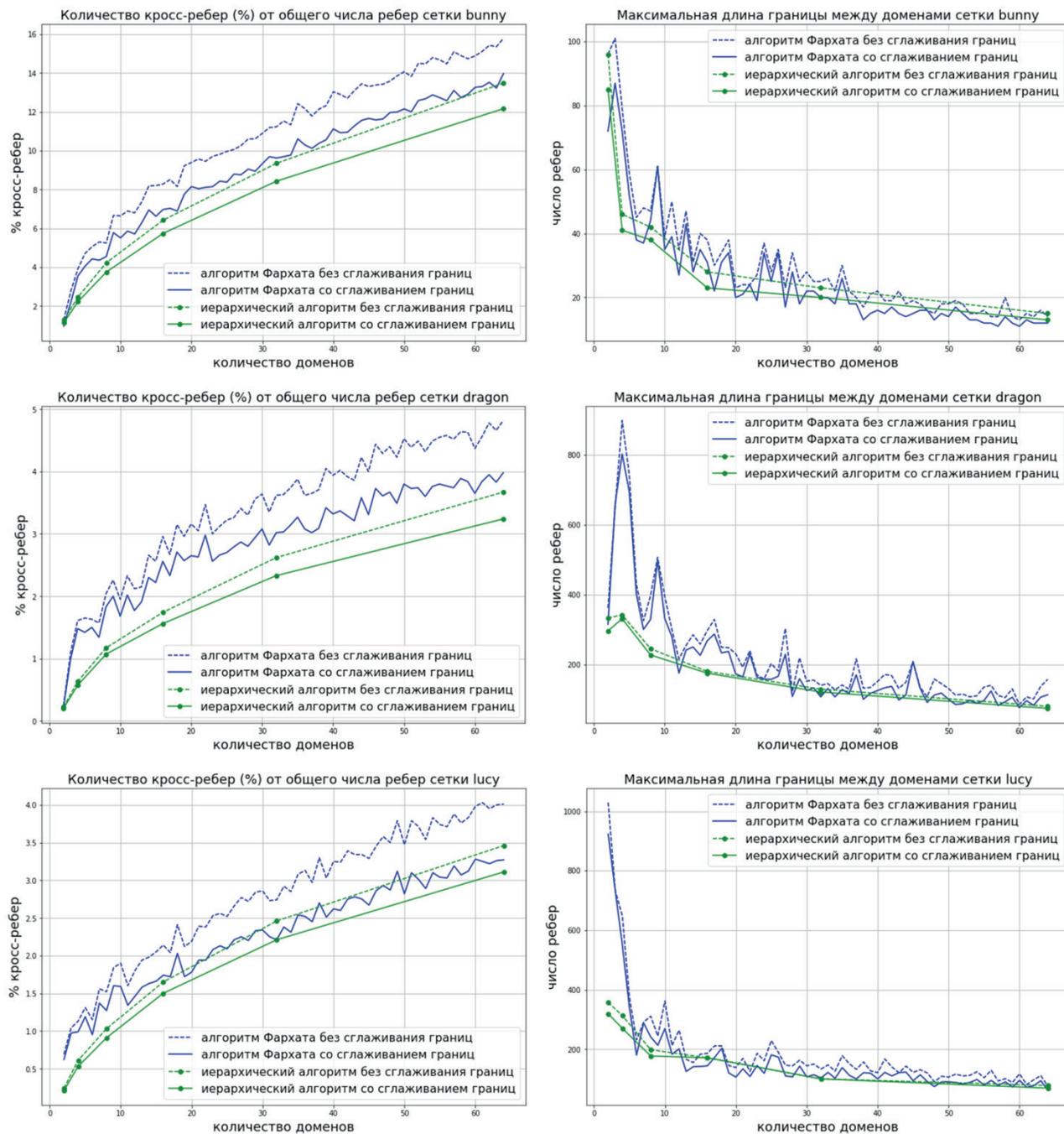
Заключение

Задача качественной декомпозиции расчетной сетки имеет важное значение для повышения эффективности суперкомпьютерных расчетов. При этом двумя наиболее важными показателями качества декомпозиции являются равномерность распределения ячеек по доменам и длина максимальной границы между доменами. В рамках работ по исследованию подходов к повышению качества декомпозиции расчетных сеток для использования в практических задачах в качестве ведущего показателя эффективности был выбран параметр равномерности распределения ячеек по доменам. Рассматривались только такие алгоритмы декомпозиции, которые приводят к идеальному распределению ячеек: алгоритм Фархата и иерархический пространственный алгоритм декомпозиции. Для выбранных алгоритмов встала задача сокращения длин границ между доменами без потери равномерности распределения. Для этого был реализован алгоритм сглаживания границ между доменами, применение которого привело к 10% сокращению общего количества кросс-ребер в расчетной сетке, максимальной длины границы и общего времени межпроцессных обменов при решении практических задач.

¹ The Stanford 3D Scanning Repository : офиц. сайт [Электронный ресурс]. 2021. URL: <http://graphics.stanford.edu/data/3Dscanrep> (дата обращения: 12.03.2021).

² Bartkus T., Wright W. B., Struk P., 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. SAE International, 2015. doi: <https://doi.org/10.4271/2015-01-2094>





Р и с. 4. Графики доли кросс-ребер (%) от общего числа ребер сетки и максимальной длины границы между доменами при декомпозиции тестовых сеток bunny, dragon и lucy на количество доменов от 2 до 64

Fig. 4. Cross-edges fraction graphs (%) of the total edges numbers of the grid and the maximum length of the boundary between domains in the decomposition of bunny, dragon and lucy test grids by the number of domains from 2 to 64



Список использованных источников

- [1] Aldinucci, M. Practical parallelization of scientific applications with OpenMP, OpenACC, and MPI / M. Aldinucci, V. Cesare, I. Colonnelli [и др.]. — DOI 10.1016/j.jpdc.2021.05.017 // *Journal of Parallel and Distributed Computing*. — 2021. — Vol. 157. — Pp. 13-29.
- [2] 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*; ed. by M. Geshi. — Springer, Singapore, 2019. — Vol. 2. — Pp. 1-9.
- [3] 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 Mathematics*. — 2019. — Vol. 40, issue 5. — Pp. 566-573.
- [4] Utkin, A. V. Parallel molecular dynamics for silicon and silicon carbide: MPI, CUDA and CUDA-MPI Implementation / A. V. Utkin, V. M. Fomin, E. I. Golovneva. — DOI 10.1063/5.0028297 // *AIP Conference Proceedings*. — 2020. — Vol. 2288, issue 1. — Pp. 030083.
- [5] Porting VASP from MPI to MPI+OpenMP [SIMD] / F. Wende, M. Marsman [и др.]. — DOI 10.1007/978-3-319-65578-9_8 // *Scaling OpenMP for Exascale Performance and Portability. IWOMP 2017. Lecture Notes in Computer Science*; ed. by B. de Supinski, et al. — Springer, Cham, 2017. — Vol. 10468. — Pp. 107-122.
- [6] 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.
- [7] AbdulRazzaq, A. A. Parallel implementation of maximum-shift algorithm using OpenMP / A. A. AbdulRazzaq, Q. S. Hamad, A. M. Taha. — DOI 10.11591/ijeecs.v22.i3.pp1529-1539 // *Indonesian Journal of Electrical Engineering and Computer Science*. — 2021. — Vol. 22, No. 3. — Pp. 1529-1539.
- [8] Aznavah, M. Parallel GraphBLAS with OpenMP / M. Aznavah, J. Chen, T. Davis [и др.]. — DOI 10.1137/1.9781611976229.14 // *2020 Proceedings of the SIAM Workshop on Combinatorial Scientific Computing*. — Society for Industrial and Applied Mathematics, 2020. — Pp. 138-148.
- [9] Suci, A. Parallel implementation of a PIC simulation algorithm using OpenMP / A. Suci, A. Hangan, A. Marginean [и др.]. — DOI 10.15439/2020F130 // *Proceedings of the Federal Conference on Computer Science and Information Systems*. — Sofia, Bulgaria: IEEE Press, 2020. — Pp. 381-385.
- [10] McDoniel, W. LAMMPS' PPPM Long-Range Solver for the Second Generation Xeon Phi / W. McDoniel, M. Höhnerbach [и др.]. — DOI 10.1007/978-3-319-58667-0_4 // *High Performance Computing. ISC High Performance 2017. Lecture Notes in Computer Science*; ed. by J. M. Kunkel, R. Yokota, P. Balaji, D. Keyes. — Springer, Cham, 2017. — Vol. 10266. — Pp. 61-78.
- [11] Malas, T. Optimization of the Sparse Matrix-Vector Products of an IDR Krylov Iterative Solver in EMGeo for the Intel KNL Manycore Processor / T. Malas, T. Kurth, J. Deslippe. — DOI 10.1007/978-3-319-46079-6_27 // *High Performance Computing. ISC High Performance 2016. Lecture Notes in Computer Science*; ed. by M. Taufer, B. Mohr, J. Kunkel. — Springer, Cham, 2016. — Vol. 9945. — Pp. 378-389.
- [12] Cook, B. High Performance Optimizations for Nuclear Physics Code MFDn on KNL / B. Cook, P. Maris, M. Shao. — DOI 10.1007/978-3-319-46079-6_26 // *High Performance Computing. ISC High Performance 2016. Lecture Notes in Computer Science*; ed. by M. Taufer, B. Mohr, J. Kunkel. — Springer, Cham, 2016. — Vol. 9945. — Pp. 366-377.
- [13] Ka Sing, J.T. Hadamard transform improvement for HEVC using Intel AVX-512 / J. T. Ka Sing, U. Ullah Sheikh, M. Mokji [и др.]. — DOI 10.1109/ISCAIE.2019.8743904 // *2019 IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. — IEEE Press, Malaysia, 2019. — Pp. 310-315.
- [14] Shabanov, B. M. Vectorization of High-performance Scientific Calculations Using AVX-512 Instruction Set / B. M. Shabanov, A. A. Rybakov, S. S. Shumilin. — DOI 10.1134/S1995080219050196 // *Lobachevskii Journal of Mathematics*. — 2019. — Vol. 40, issue 5. — Pp. 580-598.
- [15] Rybakov, A. A. Vectorization of the Riemann solver using the AVX-512 instruction set / A. A. Rybakov, S. S. Shumilin. — DOI 10.25209/2079-3316-2019-10-3-41-58 // *Program Systems: Theory and Applications*. — 2019. — Vol. 10, issue 3(42). — Pp. 41-58.
- [16] Urschel, J. C. Spectral bisection of graphs and connectedness / J. C. Urschel, L. T. Zikatanov. — DOI 10.1016/j.laa.2014.02.007 // *Linear Algebra and its Applications*. — 2014. — Vol. 449. — Pp. 1-16.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] Farhat, C. A simple and efficient automatic fem domain decomposer / C. Farhat. — DOI 10.1016/0045-7949(88)90004-1 // *Computers & Structures*. — 1988. — Vol. 28, issue 5. — Pp. 579-602.
- [21] Preis, R. PARTY — a Software Library for Graph Partitioning / R. Preis, R. Diekmann. — DOI 10.4203/ccp.45.3.1 // *Advances in Computational Mechanics for Parallel and Distributed Processing*; ed. by B. H. V. Topping. — Civil-Comp Press, Edinburgh, UK, 1997. — Pp. 63-71.
- [22] Якобовский, М. В. Инкрементальный алгоритм декомпозиции графов / М. В. Якобовский // *Вестник Нижегородского университета им. Н. И. Лобачевского. Се-*



- рия «Математическое моделирование и оптимальное управление». — 2005. — № 1(28). — С. 243-250.
- [23] 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.
- [24] 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. — IOS Press, 2015. — Vol. 26. — Pp. 82-100.
- [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 association, 2017. — Pp. 1-15.
- Поступила 12.03.2021; одобрена после рецензирования 20.04.2021; принята к публикации 15.05.2021.
- Об авторах:**
- Багров Андрей Дмитриевич**, инженер-программист 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: <https://orcid.org/0000-0002-9473-1925>**, andrey.bagrov@yandex.ru
- 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: <https://orcid.org/0000-0002-9755-8830>**, rybakov@jssc.ru
- Все авторы прочитали и одобрили окончательный вариант рукописи.
- ## References
- [1] Aldinucci M., Cesare V., Colonnelli I., et al. Practical parallelization of scientific applications with OpenMP, OpenACC, and MPI. *Journal of Parallel and Distributed Computing*. 2021; 157:13-29. (In Eng.) DOI: <https://doi.org/10.1016/j.jpdc.2021.05.017>
- [2] Minami K. Supercomputers and Application Performance. In: Ed. by M. Geshi. *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
- [3] 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>
- [4] Utkin A.V., Fomin V.M., Golovneva E.I. Parallel molecular dynamics for silicon and silicon carbide: MPI, CUDA and CUDA-MPI Implementation. *AIP Conference Proceedings*. 2020; 2288(1):030083. (In Eng.) DOI: <https://doi.org/10.1063/5.0028297>
- [5] Wende F., Marsman M., Zhao Z., Kim J. Porting VASP from MPI to MPI+OpenMP [SIMD]. In: Ed. by B. de Supinski, S. Olivier, C. Terboven, B. Chapman, M. Müller. *Scaling OpenMP for Exascale Performance and Portability. IWOMP 2017. Lecture Notes in Computer Science*, vol. 10468. Springer, Cham; 2017. p. 107-122. (In Eng.) DOI: https://doi.org/10.1007/978-3-319-65578-9_8
- [6] 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>
- [7] AbdulRazzaq A.A., Hamad Q.S., Taha A.M. Parallel implementation of maximum-shift algorithm using OpenMP. *Indonesian Journal of Electrical Engineering and Computer Science*. 2021; 22(3):1529-1539. (In Eng.) DOI: <https://doi.org/10.11591/ijeecs.v22.i3.pp1529-1539>
- [8] Aznavah M., Chen J., Davis T.A., et al. Parallel GraphBLAS with OpenMP. *2020 Proceedings of the SIAM Workshop on Combinatorial Scientific Computing*. Society for Industrial and Applied Mathematics; 2020. p. 138-148. (In Eng.) DOI: <https://doi.org/10.1137/1.9781611976229.14>
- [9] Suci A., Hangan A., Marginean A., et al. Parallel implementation of a PIC simulation algorithm using OpenMP. *Proceedings of the Federal Conference on Computer Science and Information Systems (FedCSIS)*. IEEE Press, Sofia, Bulgaria; 2020. p. 381-385. (In Eng.) DOI: <https://doi.org/10.15439/2020F130>
- [10] McDoniel W., Höhnerbach M., Canales R., Ismail A.E., Bientinesi P. LAMMPS' PPPM Long-Range Solver for the Second Generation Xeon Phi. In: Ed. by J. M. Kunkel, R. Yokota, P. Balaji, D. Keyes. *High Performance Computing. ISC High Performance 2017. Lecture Notes in Computer Science*, vol. 10266. Springer, Cham; 2017. p. 61-78. (In Eng.) DOI: https://doi.org/10.1007/978-3-319-58667-0_4
- [11] Malas T., Kurth T., Deslippe J. Optimization of the Sparse Matrix-Vector Products of an IDR Krylov Iterative Solver in EMGeo for the Intel KNL Manycore Processor. In: Ed. by M. Taufer, B. Mohr, J. Kunkel. *High Performance Computing. ISC High Performance 2016. Lecture Notes in Computer Science*, vol. 9945. Springer, Cham; 2016. p. 378-389. (In Eng.) DOI: https://doi.org/10.1007/978-3-319-46079-6_27
- [12] Cook B., Maris P., Shao M. High Performance Optimizations for Nuclear Physics Code MFDn on KNL. In: Ed. by M. Taufer, B. Mohr, J. Kunkel. *High Performance Computing. ISC High Performance 2016. Lecture Notes in Computer Science*, vol. 9945. Springer, Cham; 2016. p. 366-377. (In Eng.) DOI:



- https://doi.org/10.1007/978-3-319-46079-6_26
- [13] Ka Sing J.T., Ullah Sheikh U., Mokji M., et al. Hadamard Transform Improvement for HEVC using Intel AVX-512. *2019 IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. IEEE Press, Malaysia; 2019. p. 310-315. (In Eng.) DOI: <https://doi.org/10.1109/ISCAIE.2019.8743904>
- [14] 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>
- [15] Rybakov A.A., Shumilin S.S. Vectorization of the Riemann solver using the AVX-512 instruction set. *Program Systems: Theory and Applications*. 2019; 10(3):41-58. (In Eng.) DOI: <https://doi.org/10.25209/2079-3316-2019-10-3-41-58>
- [16] 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>
- [17] 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>
- [18] 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>
- [19] 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>
- [20] 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)
- [21] Preis R., Diekmann R. PARTY — a Software Library for Graph Partitioning. In: Ed. by B. H. V. Topping. *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>
- [22] 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.)
- [23] 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. (In Russ., abstract in Eng.) DOI: <https://doi.org/10.25559/SITI-TO.16.202004.851-861>
- [24] Wright N.J., Dosanjh S.S., Andrews A.K., et al. Cori: A Pre-Exascale Supercomputer for Big Data and HPC Applications. *Advances in Parallel Computing, Big Data and High Performance Computing*, vol. 26. IOS Press; 2015. p. 82-100. (In Eng.) DOI: <https://doi.org/10.3233/978-1-61499-583-8-82>
- [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)*. EUCASS association; 2017. p. 1-15. (In Eng.) DOI: <https://doi.org/10.13009/EUCASS2017-258>

Submitted 12.03.2021; approved after reviewing 20.04.2021;
accepted for publication 15.05.2021.

About the authors:

Andrey D. Bagrov, Software Engineer in Joint Supercomputer Center of the Russian Academy of Sciences — Branch of the Federal State Institution “Scientific Research Institute for System Analysis of the Russian Academy of Sciences” (32a Leninsky prospect, Moscow 119334, Russian Federation), **ORCID:** <https://orcid.org/0000-0002-9473-1925>, andrey.bagrov@yandex.ru

Alexey A. Rybakov, Lead Researcher in Joint Supercomputer Center of the Russian Academy of Sciences — Branch of the 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:** <https://orcid.org/0000-0002-9755-8830>, rybakov@jssc.ru

All authors have read and approved the final manuscript.

