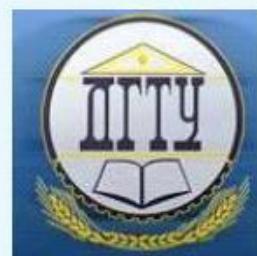


ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ INFORMATION TECHNOLOGY



Check for updates

УДК 004.382.2, 004.7

<https://doi.org/10.23947/2587-8999-2024-8-1-12-28>

Обзорная статья



**Развитие суперкомпьютерных технологий
в ИММ РАН и ИПМ им. М.В. Келдыша РАН**

М.В. Якобовский ☰, М.А. Корнилина

Институт прикладной математики им. М.В. Келдыша Российской академии наук, г. Москва, Российская Федерация

✉ lira@imamod.ru

Аннотация

Представлен обзор работ в области суперкомпьютерных технологий, проводившихся в ИММ РАН и ИПМ им. М.В. Келдыша РАН. Описаны этапы развития вычислительной техники и алгоритмы, разработанные для суперкомпьютерных систем, такие как гиперболизация уравнений, балансировка загрузки, отказоустойчивость, построение динамически адаптивных расчетных сеток, рациональная декомпозиция сеток, а также некоторые прикладные и научные задачи, успешно решаемые с использованием суперкомпьютеров.

Ключевые слова: суперкомпьютеры, центр коллективного пользования, гиперболические уравнения, гиперболизация, балансировка загрузки, отказоустойчивость, динамически адаптивные расчетные сетки, рациональная декомпозиция сеток

Благодарности. Выражаем искреннюю благодарность за ценные обсуждения и предоставленные материалы руководителю ЦКП ИПМ им. М.В. Келдыша РАН С.В. Полякову, а также сотрудникам ЦКП, с использованием оборудования которого выполнены многочисленные расчеты, упомянутые в данной публикации.

Для цитирования. Якобовский М.В., Корнилина М.А. Развитие суперкомпьютерных технологий в ИММ РАН и ИПМ им. М.В. Келдыша РАН. *Computational Mathematics and Information Technologies*. 2024;8(1):12–28.
<https://doi.org/10.23947/2587-8999-2024-8-1-12-28>

Review article

Development of Supercomputer Technologies at the Institute of Mathematical Modelling and Keldysh Institute of Applied Mathematics of Russian Academy of Sciences

Mikhail V. Yakobovskiy ☰, Marina A. Kornilina

Keldysh Institute of Applied Mathematics of Russian Academy of Sciences, Moscow, Russian Federation

✉ lira@imamod.ru

Abstract

A review scientific work in the field of supercomputer technologies at the Institute of Mathematical Modelling and Keldysh Institute of Applied Mathematics is presented. Progress in supercomputer technologies, programming tools and technique (such as hyperbolization, load balancing, fault tolerance, adaptive mesh refinement, rational mesh decomposition) and several supercomputer applications are presented.

Keywords: supercomputers, supercomputing resource center, hyperbolic equations, hyperbolization, load balancing, fault tolerance, dynamically adaptive computational grids, rational mesh decomposition

Acknowledgements. We would like to express our sincere gratitude to S.V. Polyakov the head of Supercomputing Resource Center at the Keldysh Institute for the valuable discussions and materials provided, as well as to the team of the Center for the opportunity to use the facilities of the Center for carrying out numerous computations mentioned in this publication.

For citation. Yakobovskiy M.V., Kornilina M.A. Development of supercomputer technologies at the Institute of Mathematical Modelling and Keldysh Institute of Applied Mathematics of Russian Academy of Sciences. *Computational Mathematics and Information Technologies*. 2024;8(1):12–28. <https://doi.org/10.23947/2587-8999-2024-8-1-12-28>

Статья подготовлена на основе выступления члена-корреспондента РАН М.В. Якововского на научной конференции в ИПМ им. М.В. Келдыша РАН (ИПМ), посвященной 80-летию научного руководителя ИПМ академика Б.Н. Четверушкина.

Современное состояние вычислительной техники в России и в мире. В настоящее время научный прогресс немыслим без использования суперкомпьютерных технологий. Массивные параллельные компьютеры применяются во всех сферах деятельности, а мощность самых передовых из них преодолела заветную отметку 1 Экзафлопс. Производительность 500 наиболее мощных открытых суперкомпьютерных систем в мире, представленная на рис. 1 в логарифмическом масштабе, по своему виду напоминает обратную функцию. В список Top-500 [1] попали всего несколько российских систем и, как мы видим, по своей производительности они уступают наиболее мощным компьютерам на несколько порядков. Но на этом графике интересна его правая ветвь. На уровне около 2 Петафлопс (2×10^{15} операций в секунду) есть почти горизонтальная полочка, которая состоит из сотни систем с очень близкой производительностью. Это практически одинаковые «рабочие лошадки», которые используются в бизнесе и на производстве (телеинформационные системы, облачные провайдеры и др.) для решения рутинных технических и инженерных задач. Российских систем там нет. Так же как их нет и на вертикальной ветви слева (в зоне «Research Academic»), где находятся 20–30 наиболее мощных компьютеров мира, которые ставят качественно новые задачи перед разработчиками и дают принципиально новые возможности для исследователей. В актуальный рейтинговый список входят всего 7 российских суперкомпьютеров, из них три («Ляпунов», «Червоненкис» и «Галушкин») принадлежат «Яндексу», два (Christofari и Christofari Neo) — Сбербанку и по одному МГУ («Ломоносов») и МТС (GROM). В момент своего появления в 2021 году суперкомпьютер «Яндекса» «Червоненкис» занимал 19-ю строчку рейтинга.

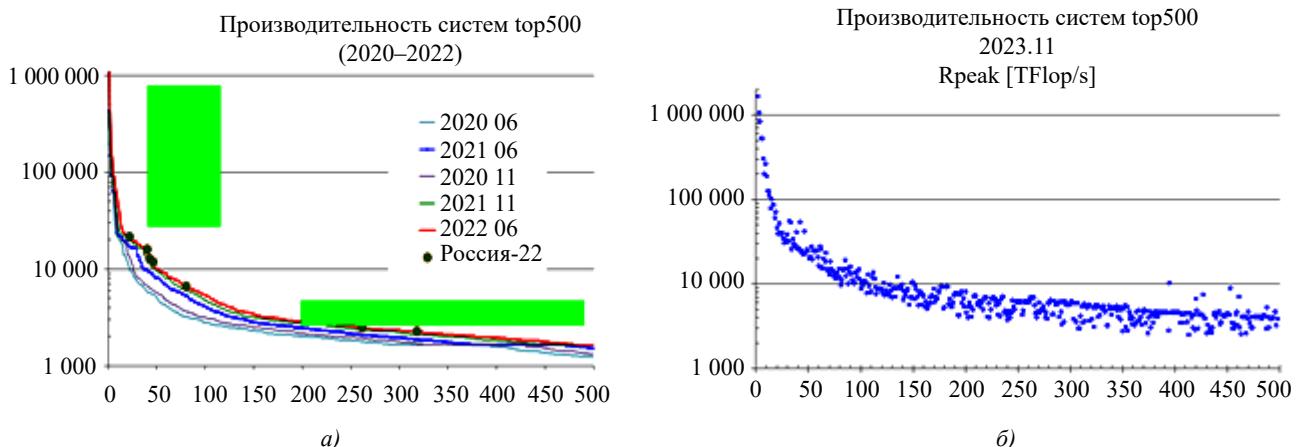


Рис. 1. Производительность суперкомпьютерных систем из списка top-500:
а — в 2020–2022 гг.; б — в 2023 г.

Вычислительные машины ИПМ не входят в Топ-500, но при этом неустанно и верно служат своим целям: используются для разработки алгоритмов и решения практических задач. Современная эпоха развития вычислительной техники в ИПМ, ознаменовавшаяся переходом к массовому использованию суперкомпьютерных технологий, началась после завершения эпохи БЭСМ-6 и серий ЕС ЭВМ почти полвека тому назад. Этапы этого развития, пройденные в Институте математического моделирования РАН (ИММ РАН) и Институте прикладной математики им. М.В. Келдыша РАН (ИПМ), отражены в таблице 1. Начиналось все в 1988 г. с ключевого решения использовать транспьютерные системы для решения научных задач [2]. Каждый транспьютер T800 [3, 4] по своей вычислительной мощности (около 10⁶ операций с плавающей точкой в секунду) соответствовал машине БЭСМ-6 и был в 3 раза производительнее доступных в это время персональных компьютеров серии PC-386. Принципиально важно, что в его состав еще входили 4 дуплексных канала передачи данных по 1 Мбайт/с в каждую сторону. Локальных сетей такого уровня в институте на тот момент еще не было. Из нескольких транспьютеров легко можно было собрать настольный суперкомпьютер.

Их можно было собрать довольно много. Транспьютеры проходили военную приемку, предназначались для космического применения и были очень надежны. Сгоревших транспьютеров в нашей практике не было. Если их подключить сейчас к какой-нибудь подходящей системе, то они благополучно будут работать. На основе этой же техники была используемая 32-процессорная станция APS-48 производительностью 64 Мегафлопса.

Таблица 1

Исторические вехи развития вычислительной техники в ИПМ им. М.В. Келдыша

Даты начала использования	Компьютеры	Направления исследований, некоторые решаемые задачи		
До 90-х. Первые транспьютерные платы в ИММ РАН				
Октябрь 1988 г.	T-800	Расчеты двумерных задач динамики вязкого газа на основе кинетически-согласованных схем		
90-е годы. Кластер ИММ РАН на основе Parsytec				
Август 1991 г.	32-х процессорная станция APS-48, 64 MFLOP	Моделирование течений сжимаемого вязкого газа; Моделирование процессов горения органических топлив; Оптимизация процессов нефтедобычи; Моделирование процессов в электронно-дырочной плазме полупроводников; Решение задач лазерного термоядерного синтеза (неустойчивость Рихтмайера-Мешкова); Параллельные алгоритмы решения задач линейной алгебры; Алгоритмы балансировки нагрузки процессоров; Визуализация результатов многопроцессорных расчетов; Управление потоком задач в распределенных системах		
1994–95 гг.	12-ти процессорная система Parsytec PowerXplorer, 1 GFlops			
1996 г.	12-ти процессорная система Parsytec CC, 3.1 GFLOP			
Май 1998 г.	32-х процессорная система Parsytec CC, 6.5 GFLOP			
Октябрь 1998 г.	2-х процессорный Sun сервер Enterprise 250 (UltraSPARC 300 MHz)			
Октябрь 1998 г.	Объединение в кластер (12 + 32 узла Parsytec CC)			
2000-е годы. ВЦ ИММ РАН				
24 апреля 2001 г.	24-х процессорный кластер (12 двухпроцессорных узлов) 14.4 Gflops Fast Ethernet (100Mbs)	Моделирование трехмерных нестационарных сверхзвуковых течений вязкого газа вокруг тел сложной формы; Решение задачи аэроакустики о генерации и распространении шумов в потоках вязкого газа; Моделирование процессов горения метана в атмосфере; Решение задачи о нелинейном электронном транспорте в квантовом переключателе; Разработка комплекса распределенной визуализации трехмерных скалярных и векторных полей, рассчитанных на МВС; Обучение студентов и аспирантов основам параллельного программирования		
2010 г. и далее. Центр коллективного пользования ИПМ им. М. В. Келдыша РАН				
2010 г. 2012 г. 2017 г. 2017 г. В состав ЦКП входят с 2019 г.	K100 (создан совместно с НИИ «Квант») K10 K60, секция без GPU и секция с GPU	Моделирование прикладных задач аэродинамики и аэроакустики; Расчеты задач физики плазмы; Решение прикладных задач механики сплошных сред; Прогнозирование распространения загрязнений водной среды на примере Азовского моря; Исследование проблем механики космического полета и управления движением; Моделирование двухфазного массообмена в пористых средах; Разработка компьютерных моделей и методов высокотемпературной гидродинамики; Разработка информационных систем принятия решений		
Ресурс	Вычислительные узлы	Ядра CPU на узле/всего	Ядра GPU на узле/всего	Пиковая производительность
K100	64	2×6/768	3×448/86016	107,9 TFLOPS
K10	16	2×8/256	3×512/24576	31,9 TFLOPS
K60, секция без GPU	86	2×14/2408	нет	81,9 TFLOPS
K60, секция с GPU	10	2×16/320	4×5120/204800	300,0 TFLOPS

В дальнейшем начали применяться гибридные системы производства Германии Parsytec PowerXplorer (1 Гфлопс), в которых были одновременно и транспьютеры, и более мощные процессоры; а также Parsytec CC, которые работали уже не на основе транспьютеров и их линков, а на основе IBM-процессоров, высокоскоростных каналов интерконнекта (HS-линков).

Первая система Parsytec CC-12 была получена в рамках проекта Европейского Союза (ЕС) под общим руководством лауреата Нобелевской премии И.Р. Пригожина для решения задач на основе квазигидродинамических систем уравнений. В 1998 г. планировалась, одобренная ЕС, поставка 96-процессорной системы Parsytec CC производительностью 19,5 Гфлопс. Но США ввели эмбарго на эту поставку, забеспокоившись, что Институт получит слишком большую вычислительную мощность. Соответственно, инфраструктуру поставили под все модули, а вот наполнили их только на 1/3. Институт получил только 32-процессорную систему Parsytec CC-32 производительностью 6,5 Гфлопс. По архитектуре она полностью эквивалентна системе Parsytec CC-12, что позволило, после их объединения внешней сетью, выполнять запуск программ на получившейся расширенной системе.

Такими были первые массивные многопроцессорные системы, позволяющие на практике разрабатывать масштабируемые параллельные алгоритмы и программы.

Дополнительное достоинство систем Parsytec CC заключалось в том, что эти суперкомпьютеры функционировали под управлением операционной системы AIX, сконфигурированной для решения задач реального времени. В Германии они использовались на производстве (в металлургической, бумажной промышленности), на транспортных магистралях для отслеживания дорожной ситуации и т. д. Практически полное отсутствие издержек времени выполнения, не связанных непосредственно с выполняемой программой, существенно облегчало отладку и настройку алгоритмов. Современные системы на основе Linux и Windows лишены указанного положительного свойства.

На рис. 2 продемонстрирована историческая фотография, сделанная перед корпусом ИММ РАН (Миусская пл., 4А) в 1998 году, в день получения, разгрузки и начала установки модулей вычислительной системы Parsytec CC-32. Крыльце корпуса А было частично демонтировано для обеспечения возможности вноса оборудования. Разгрузка с автомобильных платформ выполнялась с помощью подъёмных кранов, но транспортировка в корпус и внутри него бережно осуществлялась вручную силами научных и инженерных сотрудников, большей частью запечатлённых на фотографии.



Рис. 2. Сотрудники ИММ РАН после разгрузки модулей Parsytec CC-32. Май 1998 г.

Поскольку вычислительных модулей было поставлено на две трети меньше, чем планировалось, то осталось достаточно много незадействованной «инфраструктуры»: стойки, пустые блоки (rack's) для установки процессорных модулей, оснащенные блоками питания, и другие компоненты. Были отдельно закуплены материнские платы (двухпроцессорные, на процессорах Intel), процессоры, дисковая подсистема, сетевые карты. Из них в ИММ вручную был собран вычислительный кластер, схематично изображенный на рис. 3. Он достаточно долго работал и позволил выполнять исследования и решать различные научные задачи широкого профиля. Описанный вычислительный комплекс был выведен из эксплуатации после появления более мощных вычислительных систем, оставаясь полностью работоспособным.

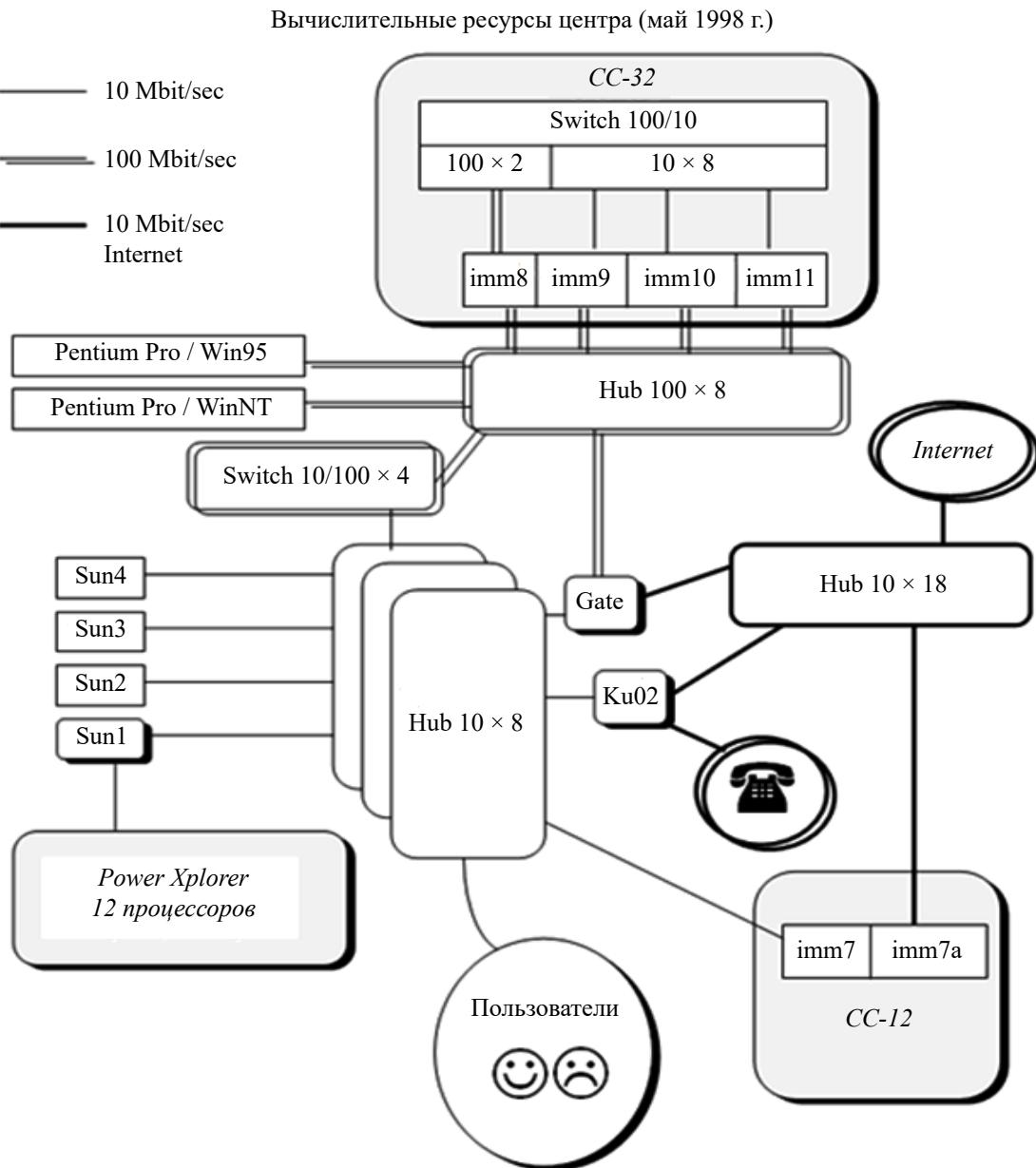


Рис. 3. Вычислительный кластер ИММ РАН
Parsytec PowerXplorer — Parsytec CC-12 — Parsytec CC-32, 1998 г.

В настоящий момент в распоряжении института, созданного для решения с помощью передовой вычислительной техники важных народно-хозяйственных задач (в первую очередь — создания ракетно-ядерного щита нашей Родины) имеется ряд высокопроизводительных систем. Со списком Топ-500 они тягаться не могут, но для решения задач, разработки алгоритмов и программ используются в полной мере. Это система К-100 с пиковой производительностью 107 Терафлопс, система К60, которая в настоящее время имеет производительность порядка 400 Терафлопс, и др. Система К-100 [5, 6] явилась первой крупной вычислительной системой в России, использующей графические платы в качестве ускорителя. Каждый вычислительный узел системы К-100 состоит из двух шестиядерных процессоров Intel Xeon X5670 и трёх ускорителей nVidia Fermi C2050, каждый из которых содержит 448-ядер CUDA. Вычислительные узлы объединены высокоскоростными сетями передачи данных МВС-экспресс и Infiniband. Система К-100 была установлена в 2010 г. по прямому распоряжению Владимира Владимировича Путина, на тот момент премьер-министра России, который выделил на это целевые средства — и в течение года система была разработана, создана, введена в эксплуатацию и до сих пор работает.

Вычислительный центр Института является распределённым, он расположен на двух территориях в разных концах Москвы. Это Центр коллективного пользования (ЦКП) [7], к которому обеспечена возможность удалённого доступа. С одной стороны, его мощности используют сотрудники института, с другой стороны, внешние пользователи — по соглашениям на возмездной или на безвозмездной основе, по коммерческим контрактам и грантам Российского научного фонда. РНФ поддерживает использование центров коллективного пользования и реализацию проектов, в рамках которых предусмотрено финансирование (до 20 % от суммы грантов) на использование мощностей ЦКП.

Суперкомпьютерные алгоритмы и решение сложных актуальных задач. В создании суперкомпьютерного Центра коллективного пользования ИПМ им. М.В. Келдыша РАН Б.Н. Четверушкин сыграл важную роль. При его непосредственном участии была получена техника вычислительного кластера ИММ РАН (Parsytec CC-12, CC-32 PowerXplorer). Но не менее важна его роль в разработке алгоритмов для суперкомпьютерной техники и в поддержке исследований данного направления. При отсутствии адекватных алгоритмов техника стоит мертвым грузом, а разработка алгоритмов для суперкомпьютерной, многопроцессорной вычислительной техники — дело отнюдь не простое. Необходимо эффективно решать широкий спектр проблем. Следует создать алгоритм, допускающий расщепление задачи — декомпозицию на большое количество «независимых» фрагментов, чтобы вычислительные процессы, выполняющие фрагменты, были бы слабо связаны между собой (взаимодействовали редко). Затем из этого алгоритма нужно сделать программу, которая будет эффективно работать. Но вычислительная техника очень разнообразна: многопроцессорная, многоядерная, с ускорителями разных видов, с распределённой оперативной памятью, рабочими станциями и облачными системами. И поэтому возникает требование адекватности и соответствия суперкомпьютерных алгоритмов вычислительной архитектуре.

Хронологически одна из первых работ, посвященных разработке суперкомпьютерных алгоритмов — это работа [8] по адаптации итерационного $(\alpha-\beta)$ -алгоритма решения многомерных параболических и эллиптических уравнений и метод его распараллеливания для систем с распределенной оперативной памятью.

Еще одним важным достижением стала разработка кинетически согласованных схем [9, 10] и квазигазодинамической (КГД) системы уравнений. В отличие от уравнений Навье-Стокса КГД система является гиперболической [11]. Это позволяет при построении численной модели использовать явные схемы, что даёт огромные преимущества при выполнении на высокопроизводительных системах с экстрамассивным параллелизмом.

Хорошо известен тезис Б.Н. Четверушкина, что у суперкомпьютерных алгоритмов должно быть два свойства: они должны быть логически простыми, иначе не получится использовать сложные иерархические вычислительные системы (в том числе ускорители), но при этом эффективными. Эта комбинация далеко не всегда достижима. Как правило, эти два требования как раз вступают друг с другом в противоречие. Известно, что явные разностные схемы очень хорошо адаптируются для суперкомпьютерных вычислений, потому что они обладают важным свойством локальности. При вычислении сеточных переменных в любой ячейке расчётный сетки используются значения сеточных величин только из соседних с ней ячеек. Это свойство локальности позволяет декомпозировать задачу на большое множество независимых процессов, которые только иногда обмениваются данными между собой по границам фрагментов сетки, обрабатываемых процессами. Хорошо известный метод domain decomposition (геометрического параллелизма), используемый при решении задач с помощью явных разностных схем, основан на данном свойстве локальности. Но явные схемы плохи тем, что при их использовании для решения эллиптических или параболических уравнений на шаг по времени накладывается очень жёсткое условие. Если вдвое измельчить шаг по пространству, то нужно в четыре раза уменьшить шаг по времени. Об этом говорит критерий Куранта и фундаментальные свойства соответствующих разностных схем и алгоритмов. И это проблема. При измельчении пространственной сетки вдвое, соответственно, в восемь раз увеличивается количество расчётных узлов. А ещё при этом в четыре раза уменьшается шаг по времени. Значит, число рассматриваемых моментов модельного времени увеличивается в четыре раза. Итого получается, что объём вычислений увеличивается в 32 раза. Это, во-первых, сильно ограничивает возможность расчёта реальных задач большого размера с хорошей точностью. А во-вторых, увеличение в четыре раза числа шагов по времени приводит к увеличению вычислительной ошибки, связанной с накоплением ошибок за счёт конечной машинной точности. Идея гиперболизации параболических систем уравнений позволяет перейти от параболических схем к схемам, похожим на явные, то есть к гиперболическим схемам, у которых более мягкое условие устойчивости. Тогда уменьшая шаг по пространству вдвое, допустимо уменьшить шаг по времени не в 4 раза, а в два раза, или, возможно, несколько больше чем в 2 раза.

Можно привести яркий пример использования этой идеи для расчёта задачи космологии — трёхмерного расчёта гравитационного потенциала [12, 13]. Задача (1) сложная: трехмерные уравнения, гравитационный потенциал, магнитная гидродинамика:

$$\begin{aligned} \frac{\delta^2 \phi}{\delta x_i^2} &= 4\pi G\rho(r), \\ \rho(r) &= \begin{cases} \rho & r \leq R \\ \delta & r > R, \end{cases} \\ \frac{\Phi_i^{j+1} - \Phi_i^j}{\Delta t} &= (\varphi_x)_x^j + F, \\ \frac{\Phi_i^{j+1} - \Phi_i^{j-1}}{2\Delta t} + \tau * \frac{\Phi_i^{j+1} - 2\Phi_i^{j-1}}{\Delta t^2} &= (\varphi_x)_x^j + F. \end{aligned} \quad (1)$$

где ϕ — гравитационный потенциал; F — свободный член; G — гравитационная постоянная.

В уравнении выделено слагаемое, которое преобразует параболическую систему в гиперболическую, и показано, во сколько раз удаётся выиграть в размере шага по времени. Если мы используем гиперболизированную систему уравнений вместо исходной параболической, то при измельчении шага по пространству мы более чем в 40 раз выигрываем в шаге по времени. Скорость вычислений увеличится в 40 раз и, соответственно, накопленная ошибка тоже будет меньше (рис. 4).

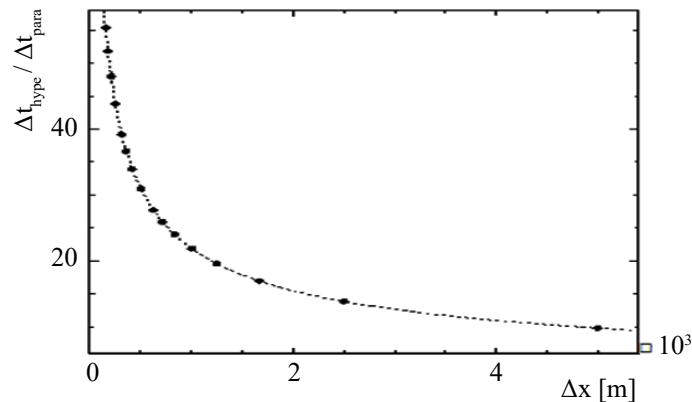


Рис. 4. Отношение шагов временной дискретизации параболического и гиперболического метода как функция дискретизации по пространству при решении уравнения Пуассона

Следует отметить, что предложенная гиперболизация не приводит к изменению качества аппроксимации. Качество аппроксимации остаётся в тех же рамках, что и у уравнений Навье-Стокса. Рис. 5 иллюстрирует картину взаимодействия межзвёздного газа с массивным телом. Представлено два варианта расчёта. На рис. 5 *а* показаны результаты расчёта с низким разрешением. На рис. 5 *б* — результат расчёта на сетке высокого разрешения. Высокое разрешение означает, что для трёхмерной задачи используется порядка 6 млрд узлов. На сетке низкого разрешения (менее миллиарда узлов) мы фактически не видим содержательной картины. А на сетке высокого разрешения можно увидеть процесс формирования джета — релятивистской струи, образующейся в результате взаимодействия гравитационного и магнитного полей. Струи, полученные в расчётах, действительно существуют. Релятивистскую струю из центра галактики M 87 астрономы наблюдают в различных диапазонах волн оптическими и радиотелескопами. Широко известен снимок джета (J.A. Biretta и др., 2000 г.), полученный с орбитально-го телескопа Хаббл [14]. Для решения задач такого рода невозможно использовать лишь фрагмент суперкомпьютера — нужна большая вычислительная мощность, то есть практически весь суперкомпьютер.

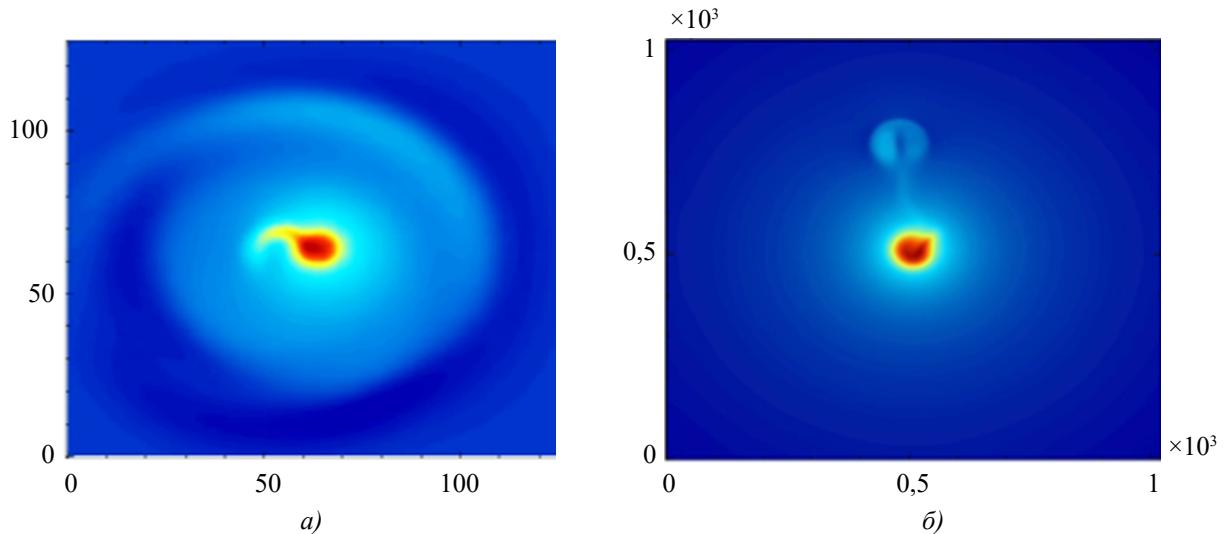


Рис. 5. Гравитационный потенциал:
а — результаты расчета с низким разрешением (нечеткая картина);
б — результаты расчета с высоким разрешением (релятивистская струя)

Заметим, что мощные суперкомпьютеры на практике преимущественно используются в разделляемом многозадачном режиме для решения потоков относительно небольших задач большим количеством научных групп. Каждая группа применяет для конкретного расчёта только небольшой фрагмент суперкомпьютера — одну десятую, одну сотую его части. Но задачи указанного класса могут быть решены только на полной вычислительной мощности всего суперкомпьютера.

При наличии достаточной вычислительной мощности можно промоделировать эти процессы и изучить их цифровую модель. Если такой вычислительной мощности нет, то мы видим только размытую картину, как на рис. 5 *а*.

Что представляет собой система с большой вычислительной мощностью? Такая система может содержать сотни и тысячи ядер. Использование таких масштабных систем требует создания специального программного обеспечения. Это большой вызов, порождающий множество задач, направленных на создание общего программного обеспечения для проведения на экстрамассивных суперкомпьютерных системах масштабных вычислительных экспериментов. Наряду с традиционными задачами, которые для этого необходимо решать (декомпозиция расчетов и расчётов сеток, визуализация результатов, статическая и динамическая балансировка нагрузки процессоров), есть еще одна проблема, обостряющаяся по мере усложнения компьютерных систем. Это проблема, связанная с отказоустойчивостью.

При проведении длительных (несколько часов, суток и более) расчетов, использующих всю суперкомпьютерную систему, содержащую $\sim 10^6$ и более вычислительных узлов и исполнительных устройств, вероятность того, что какая-то ее часть выйдет из строя в ходе выполнения расчета, приближается к единице. Для систем производительности $\sim 1\text{--}10$ Пфлопс время безотказной работы велико и проблема неактуальна. Однако для экзафлопсных систем, по имеющимся оценкам, время безотказной работы составляет $\sim 0,5\text{--}1$ часа [15]. Нет оснований полагать, что развитие аппаратных технологий сможет исправить эту ситуацию в ближайшие десятилетия, так как она связана с принципиальными техническими ограничениями. Более того, с ростом числа суперкомпьютерных узлов проблема только усугубляется. Это означает, что если не использовать резервирование промежуточных данных, сбой почти неизбежно не даст завершить длительный расчет. Таким образом, необходимость заботиться о сохранности данных и возможности возобновления расчета при сбое ложится на плечи разработчика программ. Стандартный подход, связанный с использованием «глобальных контрольных точек», хорош для небольших вычислительных систем, но становится недостаточным при значительном росте числа процессорных узлов [16]. На экзафлопсных системах время сохранения глобальной контрольной точки является величиной того же порядка, что и время безотказной работы. Следовательно, достаточно велика вероятность, что за время её записи или чтения может произойти новый сбой.

Вторым существенным недостатком стандартного подхода является необходимость повторного выполнения всех шагов расчёта, выполненных после сохранения глобальной контрольной точки, во всей области моделирования, несмотря на то, что сбой одного вычислительного устройства приводит лишь к локальной потере данных о небольшой области расчета.

В работах [16, 17] был предложен подход, позволяющий проводить длительные расчеты с использованием локальных контрольных точек. При этом не требуется приостанавливать расчет для их записи и не требуется перезапуск всей программы после сбоя. Данные локальных точек сохраняются непосредственно в памяти вычислительных узлов, участвующих в расчете, и «соседних» с каждым из них, что обеспечивает локальный характер выполнения операций. В круг эффективно выполняемых попадают алгоритмы, отвечающие свойству локальности обработки сеточных элементов, — алгоритмы на основе явных разностных схем, в том числе схемы, полученные с помощью ранее упомянутой процедуры гиперболизации. Схема восстановления данных после сбоя также опирается на идею гиперболизации уравнений. Если какой-то вычислительный узел оказался исключён из процесса вычислений вследствие аппаратного или программного сбоя, то вместо него в работу включаются два или более из заранее зарезервированных запасных узлов. Утерянные в результате сбоя данные восстанавливаются из оперативной или дисковой памяти «соседних», сохранивших работоспособность узлов. Важным свойством гиперболических уравнений является конечность времени распространения возмущений и, как следствие, конечность той зоны, которая оказывает влияние на решение на следующем шаге по времени в каждой локальной точке (рис. 6).

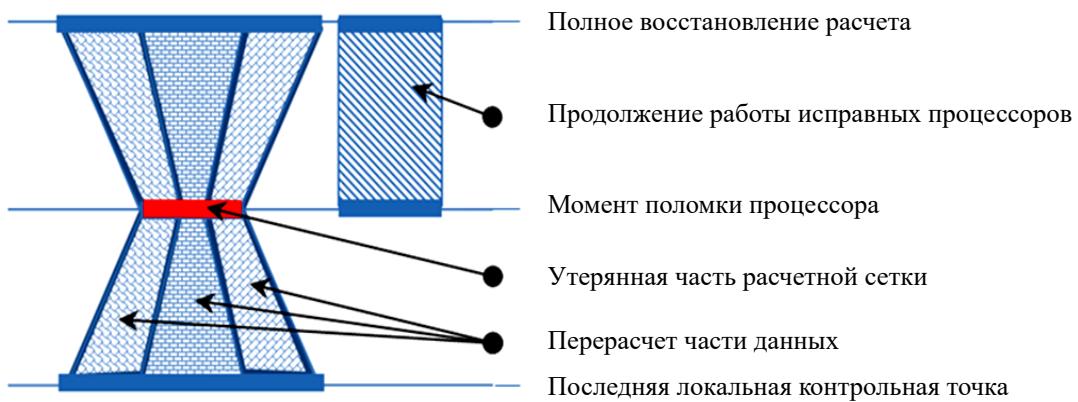


Рис. 6. Схема организации вычислений при восстановлении расчета

Таким образом, для восстановления последствий потери данных понадобится лишь незначительно расширить зону расчета, «отмотать» назад по времени и, взяв данные о предыдущей локальной контрольной точке, с помощью нескольких процессоров ускоренно пересчитать необходимый участок расчетной области заново. Расчет в целом в это время не останавливается — все остальные узлы продолжают расчёт последующих шагов модельного времени.

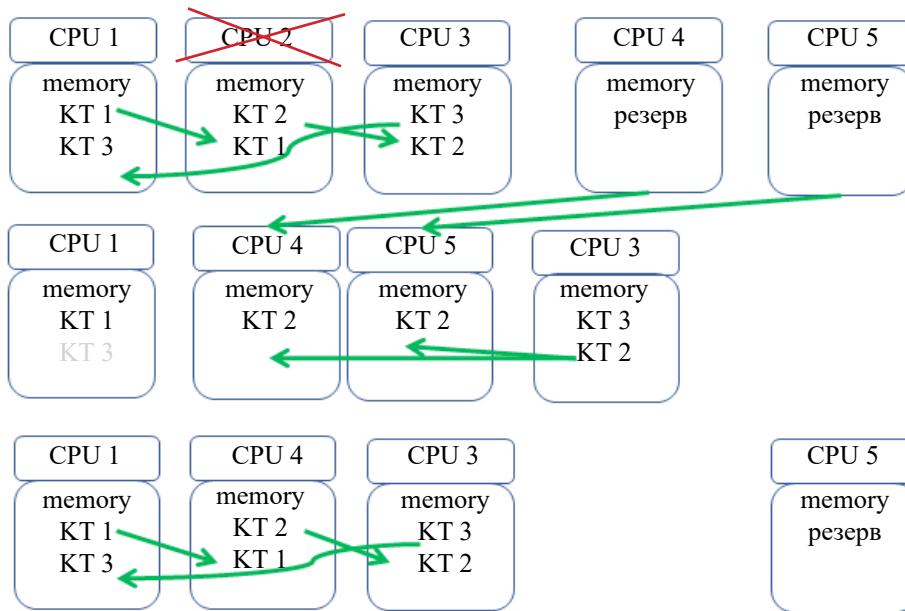


Рис. 7. Общая стратегия обеспечения отказоустойчивости: распределенная запись фрагментов контрольной точки и схема подключения резервных узлов для восстановления расчета

Следовательно, после осуществления восстановления утерянных данных необходимо продолжить ускоренное выполнение расчёта, позволяющее «догнать» расчёт, выполняемый остальными процессорами.

Схема распределенной записи контрольных точек и подключения резервных процессоров представлена на рис. 7. При такой схеме хранения данных и локального пересчета контрольной точки можно восстановить данные и продолжить расчет не только при единичном сбое, но и при нескольких одновременных отказах, независимых в метрике расчетной сетки.

Даже в трехмерном случае для исправления единичного сбоя запасных процессоров требуется всего несколько. Причём все они, кроме одного, возвращаются в число резервных после окончания описанной процедуры. Предложенные алгоритмы успешно опробованы на примере моделирования волнового уравнения (5) с источником $F(x, t)$:

$$\frac{\partial^2 f}{\partial x^2} - \frac{1}{C^2} \frac{\partial^2 f}{\partial t^2} = F(x, t). \quad (2)$$

Приведем полученную в работе [18] оценку числа резервных процессоров p_d (3), необходимых для полного восстановления после единичного отказа на уровне 0 продолжающегося расчета до уровня 2 (текущего уровня, на котором удается «догнать» продолжающийся расчет) с использованием данных контрольной точки с уровня 1 (рис. 8):

$$p_d > \frac{1}{d+1} \frac{1}{k_2} \sum_{j=1}^2 k_j \sum_{i=0}^d \alpha_j^i. \quad (3)$$

Здесь параметр $\alpha_j = 1 + 2\gamma \frac{k_j}{n_0}$; $\gamma = \frac{C\Delta t}{h}$ — число Куранта; c — наклон характеристик (5); t_0 — время сбоя; t_1 — момент модельного времени, в который была записана контрольная точка; t_2 — время полного восстановления расчета; k_1 — число шагов модельного времени, прошедшего с момента записи последней контрольной точки; k_2 — число шагов до полного восстановления расчета, причем $k_2 \leq k_1$; n_0^d — общее число расчетных точек, обрабатываемых процессором; d — размерность пространства.

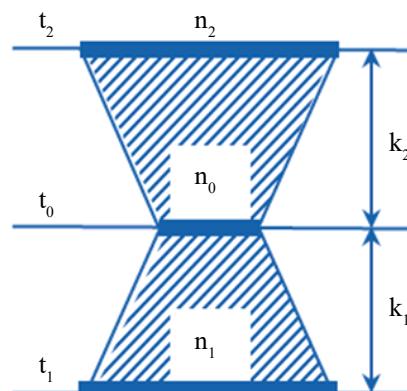


Рис. 8. Схема пространственно-временной области восстановления расчета для оценки числа резервных процессоров

Отметим, что параллельные алгоритмы, в отличие от последовательных, принципиально недетерминированы (аналогично недетерминированным автоматам). Недетерминированность приводит к тому, что для параллельных программ, в отличие от последовательных, невозможно получить полноценный набор тестов. Поэтому особую роль приобретает использование базовых методов, таких как декомпозиция по данным, конвейерный параллелизм, коллективное решение, многократно апробированных многочисленными научными коллективами. Они хорошо работают, но не универсальны. Каждый из них ориентирован на свой класс задач. У метода декомпозиции наиболее слабое ограничение на потенциально возможное при его использовании ускорение. Но этот метод плохо подходит для неявных численных схем. Конвейерный метод имеет довольно широкое поле применения, но более существенные ограничения на ускорение. У коллективного решения свои ограничения на ускорение, диктуемые соотношением характерного времени на выполнение каждого задания и времени на связанные с заданием коммуникации.

Существенно, что все базовые методы ориентированы на однородный вычислительный процесс. На практике при решении сложных задач, например, на динамически адаптивных сетках, он существенно неоднороден. Вычислительная нагрузка может сильно меняться даже на одном шаге по времени, но на разных этапах вычислений (таких как расчет газовой динамики, расчет химической кинетики, принятие оптимизационных решений). При этом на каждом шаге по времени картина распределения вычислительной нагрузки по процессорам будет отличаться для каждого из этих вычислительных этапов. А возможно, потребуется еще и разное распределение данных по процессорам на разных этапах одного шага.

На рис. 9 представлены используемые стратегии балансировки загрузки в зависимости от особенностей вычислительного процесса. В простейшем случае, когда вычислительная нагрузка однородна, при распределении заданий может использоваться статическая балансировка нагрузки.



Рис. 9. Стратегии балансировки загрузки в зависимости от соотношения вычислительной нагрузки W_i^j при расчете i -го фрагмента сетки на j и на $(j-1)$ шаге

Если нагрузка меняется плавно, то используются методы диффузной балансировки нагрузки. Но есть задачи, в которых нагрузка меняется значительно и не прогнозируемо, например, задача горения [19, 20]. Для таких задач стандартные методы балансировки загрузки оказываются неэффективны.

Задача горения (4) описывается следующими уравнениями:

$$\frac{\partial \mathbf{U}}{\partial t} + A\mathbf{U} = f, \quad \mathbf{U} = (p, \rho y^{(i)}, \rho u, \rho v, E)^T, \quad f = (0, \omega_i, 0, 0, 0)^T. \quad (4)$$

Здесь ρ — плотность; $y^{(i)}$ — массовые доли i -х компонент; u, v — скорости; p — давление; E — полная энергия; ω_i — скорости образования компонент.

Задача решается разбиением на вычислительные блоки (5–6) согласно методу суммарной аппроксимации Самарского [21]:

I. Блок Газовой динамики:

$$\frac{\partial \mathbf{U}}{\partial t} + A\mathbf{U} = 0, \quad (5)$$

$$\frac{\mathbf{U}^{j+1} - \mathbf{U}^j}{\Delta t} + \frac{1}{2}(A\mathbf{U}^{j+1} + A\mathbf{U}^j) = 0, \quad (6)$$

II. Блок химической кинетики:

$$\frac{d\mathbf{U}}{dt} = f, \quad f = (0, \omega_i, 0, 0, 0)^T. \quad (7)$$

Практически все вычисления сосредоточены в зоне интенсивного горения, где метан смешивается с кислородом. Именно там происходят интенсивные химические реакции со свето- и энерговыделением и, соответственно, именно в этой зоне необходимо решать множество независимых жестких систем обыкновенных дифференциальных уравнений (ОДУ) (7), что требует больших вычислительных мощностей.

Если всю расчетную область в соответствии с методом геометрического параллелизма равномерно распределить по процессорам, исходя из потребностей этапа (5–6), то получится, что на этапе (7) только несколько процес-

соров будут выполнять основной объём вычислений, а все остальные будут практически пропускать большую часть времени (рис. 10).

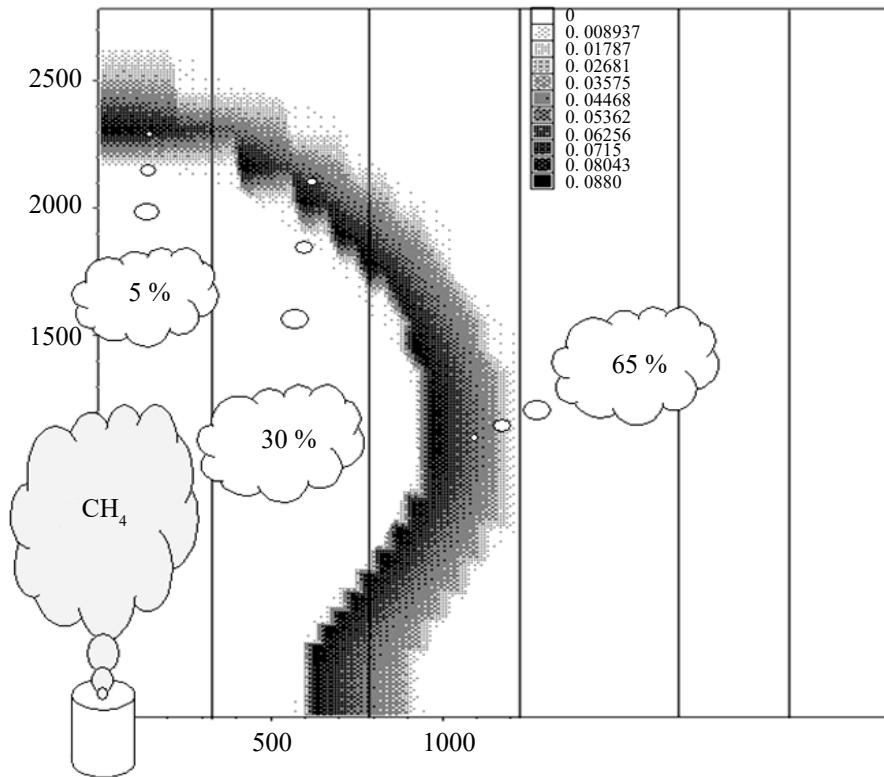


Рис. 10. Метановый факел

Это характерный пример задачи, для решения которой на двух этапах в пределах каждого шага модельного времени требуется разная стратегия распределения вычислений. Более того, если на этапе «газовой динамики» статической балансировки нагрузки достаточно, то при решении множества систем ОДУ — категорически нет. Требуется динамическая балансировка. Стандартные методы динамической балансировки здесь не подходят. Следует использовать специальные варианты метода коллективного решения, в которых много управляемых процессов, что приводит к возникновению сложной задачи обеспечения корректного взаимодействия между ними. На каждом процессоре реализуются вычислительные и управляющие процессы, логика сильно усложняется.

Соответствующий метод динамической балансировки был разработан для задач, подобных задаче горения, и опробован, в том числе, на слабосвязанных кластерах [19, 20]. Примером слабосвязанного кластера может служить кластер ИММ РАН (рис. 3), состоящий из систем Parcytec CC-12 и CC-32, соединенных между собой тонким каналом связи. Внутри каждой подсистемы были 10 мегабайтные каналы, а между ними был 1 мегабайтный канал.

Разработанные алгоритмы позволили эффективно решать задачу горения. Рост ускорения с увеличением числа процессоров был практически линейным. Причем в составе этого кластера процессоры были разные, но, тем не менее, динамическая балансировка эффективно выравнивала загрузку.

Эта же задача успешно считалась и на 1 Терафлопсном компьютере МСЦ РАН МВС 1000М, находившемся на 74 месте в списке TOP-500 в ноябре 2002 года. Задача хорошо масштабируется с эффективностью более 70 % процентов при использовании до 500 процессоров (рис. 11). Далее эффективность снижалась до 62 %, что связано с использованием сравнительно небольшой сетки, содержащей порядка 10 млн узлов. На такой сетке расчёт газодинамической части обуславливает снижение эффективности. Большие сетки (~1 млрд узлов) для этих задач использовать сложно в силу больших требований к вычислительным мощностям, необходимым для проведения на них расчета химической кинетики.

Существенной фундаментальной проблемой, связанной с использованием суперкомпьютеров, является задача рациональной декомпозиции расчетных сеток. Исследования в области создания методов и программного обеспечения для работы с большими сетками [22, 23] ведутся в институте, начиная с 1990-х годов.

Для проведения вычислений требуется распределить сеточные переменные и элементы расчётной сетки по вычислительным узлам, обеспечив равномерную нагрузку процессоров при сохранении низкого уровня накладных расходов, связанных с взаимодействием процессоров между собой. Сложность задачи нелинейно возрастает с увеличением размеров расчётных сеток, особенно неструктурированных. Рис. 12 показывает, как, используя предварительное огрубление расчётной сетки, переходить от вычислительных сеток, содержащих десятки миллиардов узлов, к небольшим макрографам, которые далее разбиваются на домены по числу процессоров.

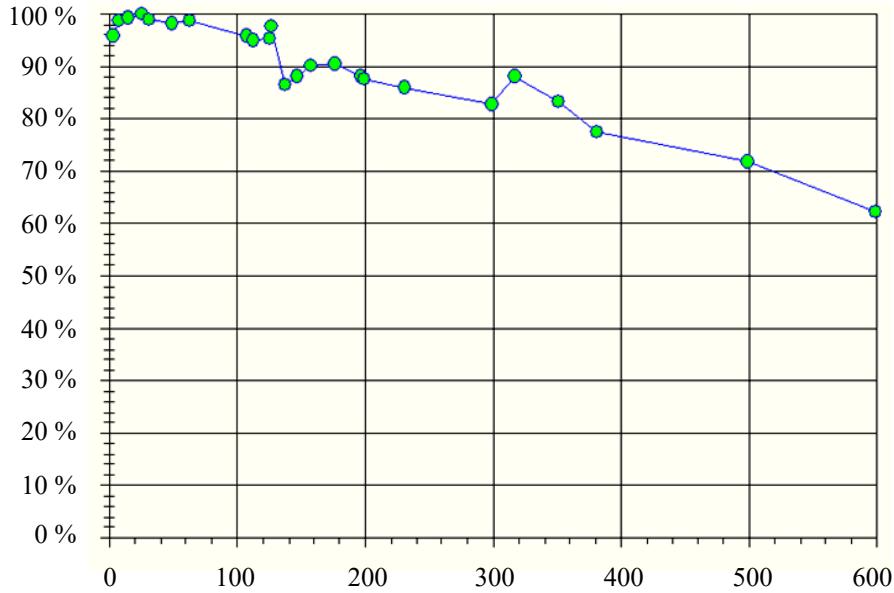


Рис. 11. Эффективность на кластере МСЦ РАН

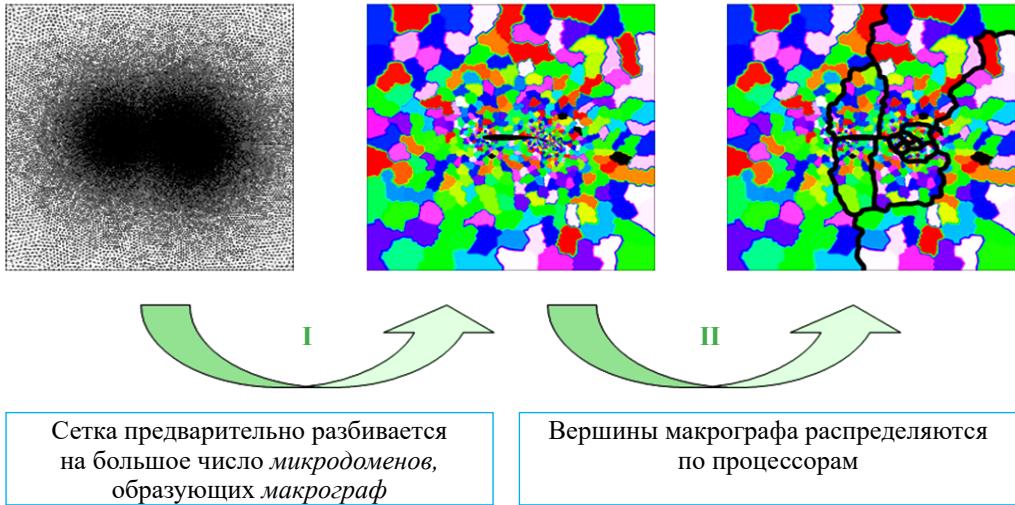


Рис. 12. Двухуровневое разбиение больших сеток

Для обработки макрографов можно использовать стандартные пакеты, такие как *parmetis*, *chaco* и другие [24–27]. Сетки размером в несколько миллионов узлов обрабатываются с их помощью быстро и эффективно. Такой иерархический подход хорошо себя зарекомендовал и при распределении сеток между процессорами во время выполнения расчетов, и при визуализации и хранении данных, и при использовании гибридных вычислительных систем.

Проблема рациональной декомпозиции, непростая даже для статических сеток, становится многократно сложнее при использовании динамически-адаптивных сеток.

Приведем пример использования адаптивных сеток для решения задачи конвективного переноса. В двумерном виде задаче описывается следующим уравнением:

$$\frac{\partial c}{\partial t} + v_x \frac{\partial c}{\partial x} + v_y \frac{\partial c}{\partial y} = 0, \quad (8)$$

где $c = c(x, y, t)$ — концентрация вещества; t — время; $(v_x, v_y) = \text{const}$ — вектор скорости переноса вещества. Моделировалась задача переноса течением капли, отличающейся по своим свойствам от свойств среды [28]. Использовалась расчётная сетка, динамически подстраивающаяся под решение. При этом результат, полученный на адаптивной сетке, был гораздо точнее, чем результат, полученный на равномерной сетке. Аналогичные результаты получены при моделировании процессов переноса примесей в водной среде на примере моделирования экологической катастрофы в Азовском море [29].

При использовании динамически адаптивных сеток необходимо во время расчета также динамически выполнять разбиение сетки на домены и перераспределение ее по процессорам. Эти функции не поддерживаются

стандартными пакетами декомпозиции. Декомпозицию можно делать на основе разных принципов: на основе иерархических алгоритмов декомпозиции (стандартные пакеты используют именно их, так как они наиболее универсальны); используя кривые, заполняющие пространства, фрактальные кривые (например, кривую Гильберта).

Динамическая балансировка во время расчета необходима для использования в расчетах вычислительных систем, в которых априори не известны свойства и топология интерконнекта, либо для задач, для которых вычислительная трудоёмкость обработки фрагментов расчётовой сетки априори неизвестна.

На рис. 13 представлены домены, сформированные в результате выполнения иерархического алгоритма декомпозиции (рис. 13 *a*, *б*) пакетом Metis [26] и при декомпозиции вдоль кривой Гильберта (рис. 13 *в*, *г*) для некоторых двух последовательных этапов адаптации сетки. Второй алгоритм далее будем называть фрактальным. Рис. 13 *а* и 13 *в* соответствуют одному моменту адаптации сетки, а рис. 13 *б* и 13 *г* — другому. В начале расчёта сетка содержала 64 базовые ячейки, каждая из которых была разделена на 16 элементарных ячеек. Общее число элементарных ячеек 1024, согласно методике [28], примерно сохранялось в течение всего расчёта.

Качество декомпозиции приемлемо в обоих случаях, но следует обратить внимание на объёмы данных, перераспределяемых между процессорами в случае иерархического и фрактального алгоритмов. При переходе от декомпозиции (рис. 13 *а*) к декомпозиции (рис. 13 *б*) объёмы перераспределяемых данных больше, чем при аналогичном переходе в случае применения декомпозиции вдоль фрактальной кривой от (рис. 13 *в*) к (рис. 13 *г*). Во втором случае границы доменов меняются более плавно. Перемещение данных при этом затрагивает только пары процессоров, имеющих соседние номера, а в первом случае — произвольные пары процессоров.

Меньшие затраты времени на вычисление доменов при использовании фрактальных кривых обеспечивают возможность более частого вызова процедуры перебалансировки, дополнительно снижая объёмы передаваемых при каждой балансировке данных. Отметим, что кроме кривой Гильберта есть множество других кривых, заполняющих пространство (рис. 14). Например, популярна кривая Мортонса, отличающаяся простотой вычисления топологии, но не сохраняющая, в отличие от кривой Гильберта, близость в физическом пространстве точек, имеющих соседние номера на самой кривой. Тем не менее методы на основе фрактальных кривых не лишены недостатков, поскольку при их использовании затруднено вычисление отношения соседства фрагментов, расположенных в разных частях кривой.

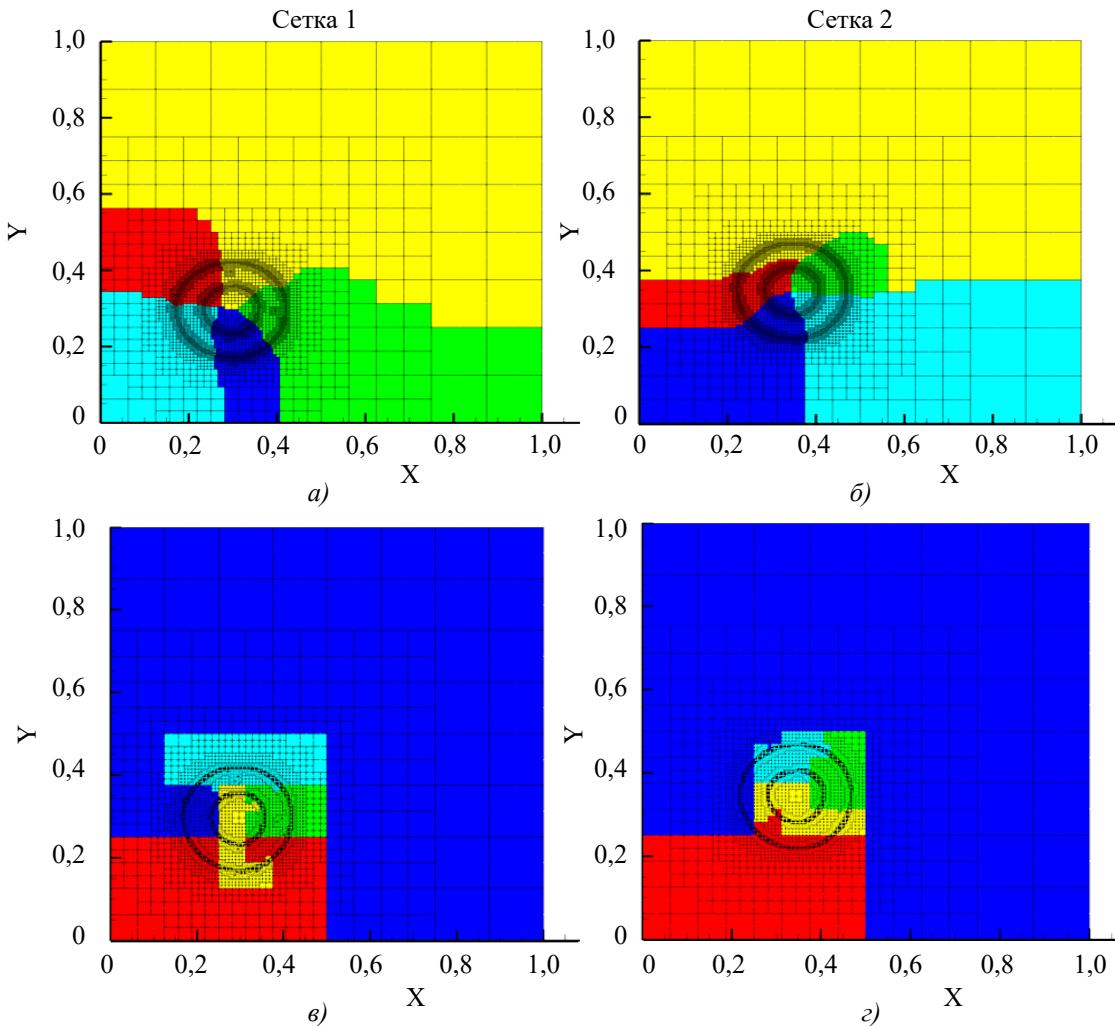


Рис. 13. Домены, сформированные с помощью иерархического метода (*а*, *б*) и вдоль кривой Гильберта (*в*, *г*)

Более подробное изучение вопроса [30, 31] показало, что соблюдение определённой дисциплины работы с сеткой позволяет использовать не фрактальные, а растровые кривые, заполняющие пространство, которые оказываются более эффективны. Они позволяют формировать домены сетки, разбивая сетку сначала на слои, потом каждый слой на полосы, и, наконец, каждую полосу на домены (рис. 15).

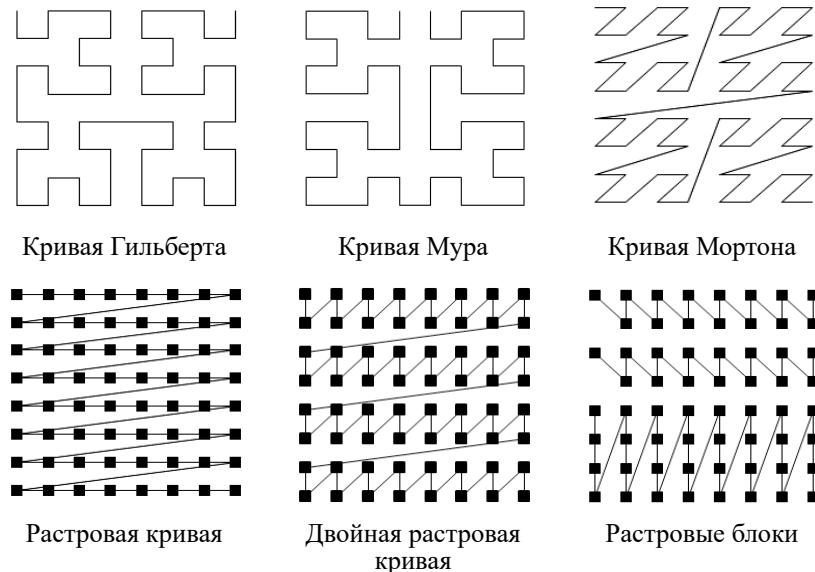


Рис. 14. Кривые, заполняющие пространство

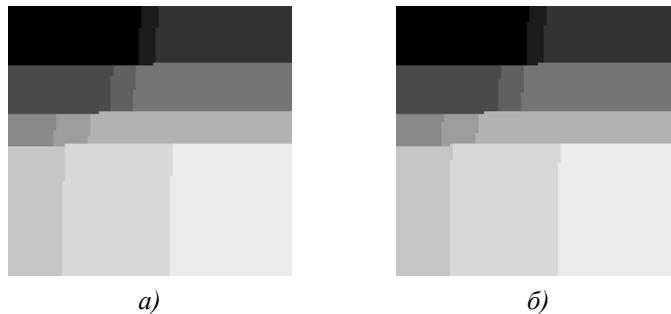


Рис. 15. 2D домены, сформированные с помощью послойного алгоритма

Идея растровых блоков позволяет при использовании локально-измельчаемых расчётов сеток получить плавную динамику изменения границ доменов, а значит — передавать меньший объём данных при перестроении сетки. Дополнительное преимущество данного подхода заключается в возможности создания простых эффективных параллельных алгоритмов самой декомпозиции, требующих низких затрат времени на её выполнение.

В данной публикации описаны многие, но далеко не все научные исследования, которые осуществлялись под руководством и при непосредственном участии академика Б.Н. Четверушкина, в которых (за исключением задачи о релятивистских струях) также участвовали авторы статьи.

Список литературы

1. Рейтинг суперкомпьютеров. URL: <http://top500.org> (дата обращения: 21.02.2024).
2. Елизарова Т.Г., Четверушкин Б.Н. Применение многопроцессорных транспьютерных систем для решения задач математической физики. *Математическое моделирование*. 1992;4(11):75–100.
3. Галушкин А.И., Точенов В.А. Транспьютерные системы — начало становления в России ЭВМ с массовым параллелизмом. *Нейрокомпьютеры: разработка, применение*. 2005;3:17–28.
4. Транспьютерные системы — начало становления в России ЭВМ с массовым параллелизмом. Сборник тезисов докладов НСКФ'2014. Переславль-Залесский: ИПС им. А.К. Айламазяна РАН; 2014. 45 с. URL: https://2014.nscf.ru/TesisAll/0_PostMoore_Plinar/01_008_GalushkinAI.pdf (дата обращения: 21.02.2024).
5. Гибридный вычислительный кластер К-100. URL: <http://www.kiam.ru/MVS/resources/k100.html> (дата обращения: 21.02.2024)
6. Давыдов А.А., Лацис А.О., Луцкий А.Е., Смольянов Ю.П., Четверушкин Б.Н., Шильников Е. В. Много-процессорная вычислительная система гибридной архитектуры «МВС-Экспресс». *Доклады Академии наук*. 2010;434(4):459–463.

7. Центр коллективного пользования ИПМ им. М.В. Келдыша РАН. URL: <http://ckp.kiam.ru> (дата обращения: 26.02.2024).
8. Четверушкин Б.Н., Чурбанова Н.Г. О применении принципа геометрического параллелизма для $(\alpha-\beta)$ -итерационного алгоритма. *Математическое моделирование*. 1991;3(3):123–129.
9. Абалакин И.В., Четверушкин. Б.Н. Кинетически согласованные разностные схемы как модель для описания газодинамических течений. *Математическое моделирование*. 1996;8(8):17–36.
10. Четверушкин Б.Н. *Кинетически-согласованные схемы в газовой динамике: новая модель вязкого газа, алгоритмы, параллельная реализация, приложения*. Москва: Издательство МГУ; 1999. 232 с.
11. Четверушкин Б.Н. Гиперболическая квазигазодинамическая система. *Математическое моделирование*. 2018;30(2):81–98.
12. Четверушкин Б.Н., Д'Ачензо Н., Савельев А.В., Савельев В.И. Кинетическая модель и уравнения магнитной газовой динамики. *Журнал вычислительной математики и математической физики*. 2018;58(5):716–725.
13. Савельев В.И., Четверушкин Б.Н. Моделирование задач магнитной гидродинамики на высокопроизводительных вычислительных системах. *Математическое моделирование*. 2020;32(12):3–13.
14. Снимок космического телескопа Хаббл. URL: <https://apod.nasa.gov/apod/ap000706.html> (дата обращения: 26.02.2024).
15. Cappello F. Fault Tolerance in Petascale/Exascale Systems: Current Knowledge, Challenges and Research Opportunities. *International Journal of High Performance Computing Applications*. 2009;23(3):212–226.
16. Четверушкин Б.Н., Якововский М.В. Вычислительные алгоритмы и отказоустойчивость гиперэкзафлопсовых вычислительных систем. *Доклады Академии наук*. 2017;472(1):1–5.
17. Четверушкин Б.Н., Якововский М.В. Вычислительные алгоритмы и архитектура систем высокой производительности. *Препринты ИПМ им. М.В. Келдыша*. 2018;52:12.
18. Chetverushkin B.N., Yakobovskiy M.V., Kornilina M.A., Semenova A.V. Numerical Algorithms for HPC Systems and Fault Tolerance Communications. *Computer and Information Science*. 2019;1063:34. https://doi.org/10.1007/978-3-030-28163-2_3
19. Корнилина М.А., Якововский М.В. Моделирование эволюции сложных нелинейных систем на многопроцессорных вычислительных комплексах. *Журнал физической химии*. 1995;69(8):1545–1548.
20. Дородницын Л.В., Корнилина М.А., Четверушкин Б.Н., Якововский М.В. Моделирование газовых течений при наличии химически активных компонентов. *Журнал физической химии*. 1997;71(12):2275–2281.
21. Самарский А.А. *Теория разностных схем*. Москва: Наука; 1989. 616 с.
22. Якововский М.В. Обработка сеточных данных на распределенных вычислительных системах. *Вопросы атомной науки и техники. Серия Математическое моделирование физических процессов*. 2004;2:40–53.
23. Якововский М.В. Инкрементный алгоритм декомпозиции графов. *Вестник Нижегородского университета им. Н.И. Лобачевского. Серия Математическое моделирование и оптимальное управление*. 2005;1(28):243–250.
24. Fiedler M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*. 1975;25(100):619–633. URL: http://www3.math.tu-berlin.de/Vorlesungen/SS14/MatricesGraphsPDEs/paper_for_students/CZMathJ-25-1975-Fiedler%20copy.pdf (дата обращения: 27.02.2024).
25. Hendrickson B., Leland R. A Multilevel Algorithm for Partitioning Graphs. *Supercomputing '95 Proceedings*. San Diego, CA, 1995. URL: <http://www.leonidzhukov.net/hse/2016/networks/papers/MultilevelAlgorithmPartitioningGraphs.pdf> (дата обращения: 27.02.2024).
26. Karypis G. Family of Graph and Hypergraph Partitioning Software. URL: <http://glaros.dtc.umn.edu/gkhome/views/metis/> (дата обращения: 27.02.2024).
27. Pothen A., Simon H.D., Kang-Pu P.L. Partitioning Sparse Matrices with Eigenvectors of Graphs. *Report RNR-89-009*. July 1989. URL: <http://snap.stanford.edu/class/cs224w-readings/Pothen89Partition.pdf> (дата обращения: 27.02.2024).
28. Сухинов А.А. *Математическое моделирование процессов переноса примесей в жидкостях и пористых средах*. Дис. канд. физ.-мат. наук. Москва; 2009. 24 с.
29. Сухинов А.А. Реконструкция экологической катастрофы в Азовском море на основе математических моделей. *Математическое моделирование*. 2008;20(6):15–22.
30. Корнилина М.А., Якововский М.В. Оценка накладных расходов при выполнении расчетов на локально измельчаемых сетках. Препринты ИПМ им. М.В. Келдыша. 2022;102:36. <https://doi.org/10.20948/prepr-2022-102>
31. Григорьев С.К., Захаров Д.А., Корнилина М.А., Якововский М.В. Динамическая балансировка нагрузки при использовании адаптивных локально измельчаемых сеток. *Математическое моделирование*. 2023;35(12):69–88. <https://doi.org/10.20948/mm-2023-12-05>

References

- Top500 Supercomputer Sites. (In Russ.). URL: <http://top500.org> (accessed: 21.02.2024).
- Elizarova T.G., Chetverushkin B.N. Application of multiprocessor transputer systems to solve problems of mathematical physics. *Mathematical Modeling*. 1992;4(11):75–100. (In Russ.).
- Galushkin A.I., Tochenov V.A. Transputer systems — the beginning of the formation of mass parallel computers in Russia. *Neurocomputers: development, application*. 2005;3:17–28. (In Russ.).

4. Transputer Systems – the Beginning of the Formation of Mass Parallel Computers in Russia. *Collection of abstracts of reports NSCF'2014*. Pereslavl-Zalessky: IPS named after A.K. Aylamazyan RAS; 2014. 45 p. (In Russ.). URL: https://2014.ncsf.ru/TesisAll/0_PostMoore_Plenar/01_008_GalushkinAI.pdf (accessed: 21.02.2024).
5. Hybrid Computing Cluster K-100. (In Russ.). URL: <http://www.kiam.ru/MVS/resources/k100.html> (accessed: 21.02.2024).
6. Davydov A.A., Latsis A.O., Lutsky A.E., Smolyanov Yu.P., Chetverushkin B.N., Shilnikov E.V. Multiprocessor computing system of hybrid architecture “MVS-Express”. *Reports of the Academy of Sciences*. 2010;434(4):459–463. (In Russ.).
7. Center for Collective Use of IPM named after M.V. Keldysh RAS. (In Russ.). URL: <http://ckp.kiam.ru> (accessed: 26.02.2024).
8. Chetverushkin B.N., Churbanova N.G. On the application of the principle of geometric parallelism for the $(\alpha-\beta)$ -iterative algorithm. *Mathematical Modeling*. 1991;3(3):123–129. (In Russ.).
9. Abalakin I.V., Chetverushkin. B.N. Kinetic consistent difference schemes as a model for describing gas dynamic flows. *Mathematical Modeling*. 1996;8(8):17–36. (In Russ.).
10. Chetverushkin B.N. *Kinetically-consistent schemes in gas dynamics: a new model of viscous gas, algorithms, parallel implementation, applications*. Moscow: Moscow State University Publishing House; 1999. 232 p. (In Russ.).
11. Chetverushkin B.N. Hyperbolic quasi-gas dynamics system. *Mathematical Modeling*. 2018;30(2):81–98. (In Russ.).
12. Chetverushkin B.N., D'Aschenzo N., Saveliev A.V., Saveliev V.I. Kinetic model and equations of magnetic gas dynamics. *Computational Mathematics and Mathematical Physics*. 2018;58(5):716–725. (In Russ.).
13. Saveliev V.I., Chetverushkin B.N. Modeling problems of magnetohydrodynamics on high-performance computing systems. *Mathematical Modeling*. 2020;32(12):3–13. (In Russ.).
14. Hubble Space Telescope Snapshot. (In Russ.). URL: <https://apod.nasa.gov/apod/ap000706.html> (accessed: 26.02.2024).
15. Cappello F. Fault Tolerance in Petascale/ Exascale Systems: Current Knowledge, Challenges and Research Opportunities. *International Journal of High Performance Computing Applications*. 2009;23(3):212–226.
16. Chetverushkin B.N., Yakobovsky M.V. Computational algorithms and fault tolerance of hyperexascale computing systems. *Reports of the Academy of Sciences*. 2017;472(1):1–5. (In Russ.).
17. Chetverushkin B.N., Yakobovsky M.V. Computational algorithms and architecture of high-performance systems. *Preprints of the M.V. Keldysh Institute of Applied Mathematics*. 2018;52:12. (In Russ.).
18. Chetverushkin B.N., Yakobovsky M.V., Kornilina M.A., Semenova A.V. Numerical Algorithms for HPC Systems and Fault Tolerance Communications. *Computer and Information Science*. 2019;1063:34. (In Russ.). URL: https://doi.org/10.1007/978-3-030-28163-2_3
19. Kornilina M.A., Yakobovsky M.V. Modeling the evolution of complex nonlinear systems on multiprocessor computing complexes. *Journal of Physical Chemistry*. 1995;69(8):1545–1548. (In Russ.).
20. Dorodnitsyn L.V., Kornilina M.A., Chetverushkin B.N., Yakobovsky M.V. Modeling gas flows in the presence of chemically active components. *Journal of Physical Chemistry*. 1997;71(12):2275–2281. (In Russ.).
21. Samarsky A.A. *Theory of Difference Schemes*. Moscow: Nauka; 1989. 616 p.
22. Yakobovsky M.V. Processing grid data on distributed computing systems. *Issues of atomic science and technology. Series Mathematical modeling of physical processes*. 2004;2:40–53. (In Russ.).
23. Yakobovsky M.V. Incremental graph decomposition algorithm. *Bulletin of Lobachevsky University of Nizhny Novgorod. Series Mathematical Modeling and Optimal Control*. 2005;1(28):243–250. (In Russ.).
24. Fiedler M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*. 1975;25(100):619–633. URL: http://www3.math.tu-berlin.de/Vorlesungen/SS14/MatricesGraphsPDEs/paper_for_students/CZMathJ-25-1975-Fiedler%20copy.pdf (accessed: 27.02.2024).
25. Hendrickson B., Leland R. A Multilevel Algorithm for Partitioning Graphs. *Supercomputing '95 Proceedings*. San Diego, CA, 1995. URL: <http://www.leonidzhukov.net/hse/2016/networks/papers/MultilevelAlgorithmPartitioningGraphs.pdf> (accessed: 27.02.2024).
26. Karypis G. Family of Graph and Hypergraph Partitioning Software URL: <http://glaros.dtc.umn.edu/gkhome/views/metis/> (accessed: 27.02.2024).
27. Pothen A., Simon H.D., and Kang-Pu P.L. Partitioning Sparse Matrices with Eigenvectors of Graphs. Report RNR-89-009, July 1989. URL: <http://snap.stanford.edu/class/cs224w-readings/Pothen89Partition.pdf> (accessed: 27.02.2024).
28. Sukhinov A.A. *Mathematical Modeling of Impurity Transport Processes in Liquids and Porous Media*. Candidate of Physical and Mathematical Sciences dissertation. Moscow, 2009. 24 p. (In Russ.).
29. Sukhinov A.A. Reconstruction of an Ecological Disaster in the Azov Sea Based on Mathematical Models. *Mathematical Modeling*. 2008;20(6):15–22. (In Russ.).
30. Kornilina M.A., Yakobovsky M.V. Overhead Costs Assessment for Calculations on Locally Refined Grids. *Preprints of the M.V. Keldysh Institute of Applied Mathematics*. 2022;102:36 p. (In Russ.). URL: <https://doi.org/10.20948/prepr-2022-102>
31. Grigoriev S.K., Zakharov D.A., Kornilina M.A., Yakobovsky M.V. Dynamic Load Balancing Using Adaptive Locally Refined Grids. *Mathematical Modeling*. 2023;35(12):69–88. (In Russ.). URL: <https://doi.org/10.20948/mm-2023-12-05>

Поступила в редакцию 19.02.2024

Поступила после рецензирования 07.03.2024

Принята к публикации 11.03.2024

Об авторах:

Якобовский Михаил Владимирович, заместитель директора по научной работе, Институт прикладной математики им. М.В. Келдыша Российской академии наук (РФ, 125047, Москва, Миусская пл., 4), [ORCID](#), [MathNet](#), [ScopusID](#), lira@imamod.ru

Корнилина Марина Андреевна, научный сотрудник, Институт прикладной математики им. М.В. Келдыша Российской академии наук (125047, Москва, Миусская пл., 4), [ORCID](#), [MathNet](#), [ScopusID](#), mary@imamod.ru

Заявленный вклад соавторов:

М.В. Якобовский — личное участие в проведении всех описываемых исследований за исключением моделирования гравитационного потенциала.

М.А. Корнилина — участие в работах по отказоустойчивости и моделированию задачи горения, анализ материалов деятельности диссертационного совета, подготовка материалов статьи.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.

Received 19.02.2024

Received 07.03.2024

Accepted 11.03.2024

About the Authors:

Mikhail V. Yakobovskiy, Deputy Director for Scientific Work, Institute of Applied Mathematics named after M.V. Keldysh of the Russian Academy of Sciences (4, Miusskaya Sq., Moscow, 125047, RF), [ORCID](#), [MathNet](#), [ScopusID](#), lira@imamod.ru

Marina A. Kornilina, Researcher, Institute of Applied Mathematics named after M.V. Keldysh of the Russian Academy of Sciences (4, Miusskaya Sq., Moscow, 125047, RF), [ORCID](#), [MathNet](#), [ScopusID](#), mary@imamod.ru

Contributions of the co-authors:

М.В. Якобовский — direct participation in all described research activities except for modeling gravitational potential.

М.А. Корнилина — involvement in work on fault tolerance and combustion modeling, analysis of materials of the dissertation council, preparation of manuscript materials.

Conflict of interest statement

The authors do not have any conflict of interest.

All authors have read and approved the final manuscript.