



doi 10.26089/NumMet.v26r439

УДК 004.272.2;
004.942

Исследование масштабируемости параллельного алгоритма численного моделирования удержания плазмы в открытых магнитных ловушках методами имитационного моделирования

И. Г. Черных

Институт вычислительной математики и математической геофизики СО РАН,
Новосибирск, Российская Федерация

ORCID: 0000-0001-9564-1553, e-mail: chernykh@ssd.ssc.ru

Д. В. Винс

Институт вычислительной математики и математической геофизики СО РАН,
Новосибирск, Российская Федерация
ORCID: 0000-0003-3909-5249, e-mail: vins@sscc.ru

В. А. Вшивков

Институт вычислительной математики и математической геофизики СО РАН,
Новосибирск, Российская Федерация
ORCID: 0000-0002-8598-6926, e-mail: vsh@ssd.ssc.ru

М. А. Боронина

Институт вычислительной математики и математической геофизики СО РАН,
Новосибирск, Российская Федерация
ORCID: 0000-0001-7090-4054, e-mail: boronina@ssd.ssc.ru

Аннотация: В статье рассматривается методика оценки масштабируемости параллельных алгоритмов. Вместо проведения дорогостоящих экспериментов на суперкомпьютерах предлагаются использовать специальную имитационную модель. Методика заключается в изучении и разработке схемы коммуникаций и вычислений исследуемого алгоритма, создании на ее основе имитационной модели с использованием модели акторов, настройке под конкретную архитектуру суперкомпьютера и исследовании масштабируемости. В статье демонстрируется применение этой методики к задаче численного моделирования удержания плазмы в открытых магнитных ловушках. Результаты имитационного моделирования свидетельствуют о 85% эффективности масштабирования алгоритма при его развертывании на тысячах вычислительных ядер.

Ключевые слова: физика плазмы, параллельные алгоритмы, имитационное моделирование.

Благодарности: Исследование выполнено при финансовой поддержке РНФ (грант № 19-71-20026).

Для цитирования: Черных И.Г., Винс Д.В., Вшивков В.А., Боронина М.А. Исследование масштабируемости параллельного алгоритма численного моделирования удержания плазмы в открытых магнитных ловушках методами имитационного моделирования // Вычислительные методы и программирование. 2025. **26**, № 4. 601–614. doi 10.26089/NumMet.v26r439.

Scalability study of parallel algorithm for numerical modeling of plasma confinement in open magnetic traps using simulation modeling methods

Igor G. Chernykh

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
Novosibirsk, Russia

ORCID: 0000-0001-9564-1553, e-mail: chernykh@ssd.sscc.ru

Dmitry V. Vins

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
Novosibirsk, Russia

ORCID: 0000-0003-3909-5249, e-mail: vins@sscc.ru

Vitaly A. Vshivkov

The Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
Novosibirsk, Russia

ORCID: 0000-0002-8598-6926, e-mail: vsh@ssd.sscc.ru

Marina A. Boronina

The Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
Novosibirsk, Russia

ORCID: 0000-0001-7090-4054, e-mail: boronina@ssd.sscc.ru

Abstract: This article presents a methodology for evaluating the scalability of parallel algorithms. Instead of conducting costly experiments on supercomputers, we propose using a specialized simulation model. The methodology involves studying and developing the communication and computation scheme of the tested algorithm, creating a simulation model based on the actor model, configuring for a specific supercomputer architecture, and investigating its scalability. The article demonstrates the application of this methodology to a numerical modeling problem for plasma confinement in open magnetic traps. The results indicate 85% scaling efficiency of the algorithm for thousands of cores.

Keywords: plasma physics, parallel algorithms, simulation modeling.

Acknowledgements: The work was supported by the Russian Science Foundation (grant No. 19–71–20026).

For citation: I. G. Chernykh, D. V. Vins, V. A. Vshivkov, M. A. Boronina, “Scalability study of parallel algorithm for numerical modeling of plasma confinement in open magnetic traps using simulation modeling methods,” Numerical Methods and Programming. **26** (4), 601–614 (2025). doi 10.26089/NumMet.v26r439.

1. Введение. Современные суперкомпьютерные системы демонстрируют высокий уровень параллельной обработки данных, гетерогенность вычислительных компонентов и иерархически распределенную организацию коммуникационных подсистем, что обуславливает значительную сложность разработки специализированного программного обеспечения и представляет собой актуальную научно-техническую проблему.

После реализации программного комплекса для решения вычислительно сложных задач ключевой проблемой становится определение точки насыщения масштабируемости, т.е. установление оптимального количества вычислительных узлов, при котором достигается максимальное ускорение вычислений (speedup) для заданной архитектуры и входных параметров, причем дальнейшее добавление вычислительных ресурсов не приводит к значимому росту производительности (закон Амдала).

Экспериментальное решение данной проблемы требует проведения серии вычислительных экспериментов с вариацией количества задействованных узлов. Однако высокая себестоимость эксплуатации современных суперкомпьютерных систем, а также лимитированная доступность машинного времени в большинстве суперкомпьютерных центров существенно ограничивают возможность применения данного подхода на практике.

В контексте интегрального подхода к разработке параллельных программ, разработанного в ИВМиМГ СО РАН [1], предлагается решение указанной проблемы посредством применения методов мультиагентного моделирования. Данный подход позволяет имитировать серию вычислительных экспериментов, существенно сокращая требования к реальным ресурсам и временными затратам, при этом обеспечивая оценку масштабируемости исследуемых алгоритмов для современных и перспективных суперкомпьютерных архитектур.

Можно выделить несколько основных направлений исследований в данной области:

- Экстрамасштабное моделирование (Extreme-Scale simulation) [2]. Основной фокус направлен на моделирование систем с миллионами ядер и экзабайтами данных. Разрабатываются легковесные модели, способные предсказывать поведение алгоритмов на таких масштабах, не требуя фактического развертывания.
- Исследование энергоэффективности [3]. Наряду с оценкой производительности, имитационные модели используются для анализа энергопотребления параллельных алгоритмов. Это позволяет оптимизировать алгоритмы не только по скорости, но и по стоимости вычислений, что критически важно для дата-центров и суперкомпьютеров.
- Гетерогенные и гибридные архитектуры [4]. Современные модели должны учитывать работу алгоритмов в средах, где сочетаются CPU, GPU, Phi и другие ускорители.
- Интеграция с машинным обучением [5]. Появляются работы, где методы ML используются внутри симуляторов для предсказания поведения системы в неопределенных условиях или для автоматической оптимизации параметров алгоритма в ходе моделирования. Особое распространение получили специализированные платформы для НРС, такие как SimGrid [6] и SST (Structural Simulation Toolkit)[7]. Эти инструменты предоставляют абстракции для моделирования вычислительных узлов и сетевой инфраструктуры, позволяя исследователям сосредоточиться только на логике алгоритма.

В данной работе представлен подход к исследованию параллельных алгоритмов на основе модели акторов.

Первоначально моделирование осуществлялось с использованием системы AGNES [8], однако выявленные ограничения данной платформы обусловили необходимость перехода к модели акторов для имитации вычислений (рис. 1) [9]. По аналогии с парадигмой объектно-ориентированного программирования, где каждый примитив рассматривается как объект, модель акторов выделяет в качестве универсальной сущности понятие “актора”. Актор представляет собой автономный вычислительный компонент, который, получив сообщение, может одновременно: отправить конечное число сообщений другим акторам, создать конечное число новых акторов, выбрать поведение, которое будет использоваться при обработке следующего полученного сообщения. Фактически, каждая вычислительная функция программы становится “черным ящиком”, принимающим входные данные в виде сообщений и выдающим результат также в виде сообщений. На рис. 1 схематично представлен такой подход. Время обработки данных разными “черными ящиками” и времена пересылок сообщений могут быть как синтетическими, так и извлекаемыми из данных по реальным запускам исследуемой программы на системах с различным количеством вычислительных ядер.

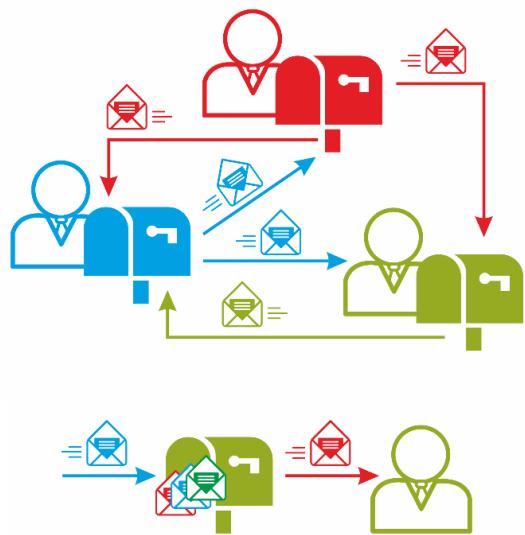


Рис. 1. Модель акторов для имитации вычислений

Fig. 1. Actor model for the computation simulation

Как показывает сравнительный анализ, данная модель обеспечивает:

- сокращение коммуникационных накладных расходов между агентами;
- повышенную эффективность при масштабировании;
- улучшенные показатели отказоустойчивости.

Результаты экспериментальной верификации подтверждают, что функциональный язык программирования Erlang демонстрирует оптимальные показатели по следующим критериям:

- простота реализации моделей;
- эффективность масштабирования;
- устойчивость к отказам.

Благодаря этому он представляет собой предпочтительный инструмент для задач имитационного моделирования масштабируемости алгоритмов [10, 11].

В данной работе будет проиллюстрировано применение такого подхода к исследованию масштабируемости алгоритма численного моделирования удержания плазмы в магнитной ловушке [12, 13].

2. Общая схема исследования параллельного алгоритма

параллельного алгоритма. В параллельных вычислениях программа представляет собой набор одновременно работающих потоков (нитей), распределенных по отдельным вычислительным узлам. Эти потоки обмениваются данными через систему сообщений. Основными параметрами, определяющими работу каждой нити, являются длительность выполнения вычислений и время, затрачиваемое на передачу данных другим узлам. Для прогнозирования работы программы на суперкомпьютере строится специальная модель, которая включает создание схемы коммуникации между всеми нитями процесса (как показано на рис. 2) и разделение работы каждой нити на фазы вычислений и фазы обмена данными. Каждая нить представлена как последовательность вычислительных и коммуникационных операций. Все взаимодействия между нитями formalизованы и могут быть проанализированы. Модель позволяет оценить временные характеристики выполнения программы до ее реального запуска на суперкомпьютере. Такой подход дает возможность предсказать поведение параллельной программы и оптимизировать ее работу для конкретной вычислительной архитектуры.

Ключевой характеристикой эффективности вычислительного процесса при его исполнении на высокопроизводительных суперкомпьютерах является масштабируемость. Здесь следует отметить, что для разных вычислительных алгоритмов может быть важен разный показатель масштабируемости. Например, методы статистического моделирования [14, 15], как правило, демонстрируют слабую масштабируемость, т.е. уменьшение времени исполнения при увеличении вычислительных узлов, а сеточные методы численного моделирования [16–18] — сильную масштабируемость, т.е. неизменность времени расчета алгоритма при увеличении расчетной области пропорционально количеству вычислительных узлов. В исследовании рассматривается исключительно межузловая масштабируемость, поскольку количество ядер вычислительных элементов зафиксировано и намного меньше, чем потенци-

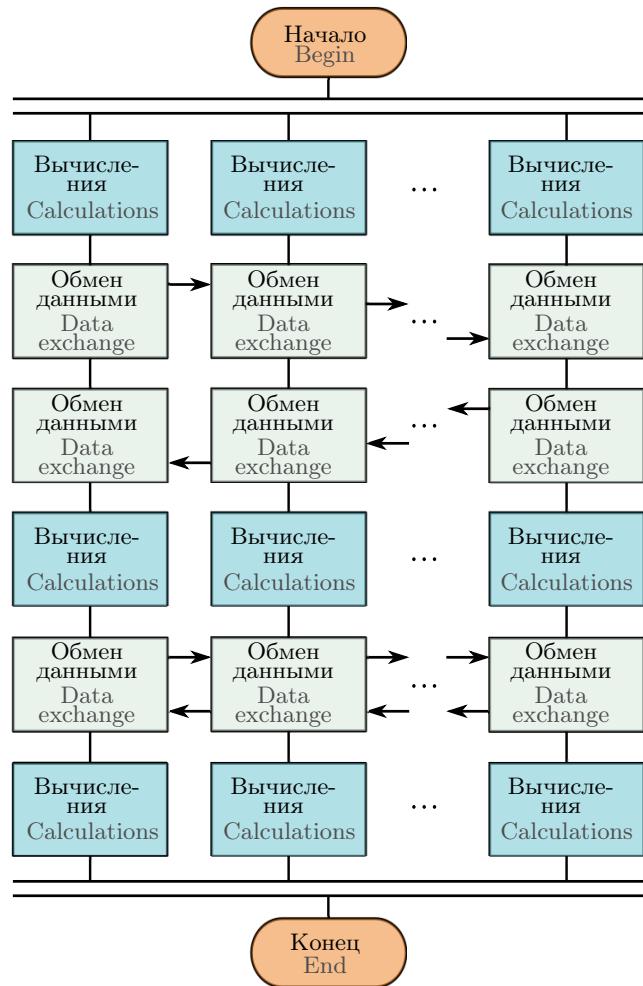


Рис. 2. Базовая схема коммуникаций параллельного вычислительного процесса

Fig. 2. Basic communication scheme of the parallel computational process

альное число вычислительных узлов высокопроизводительных суперкомпьютеров. Таким образом, все локальные операции внутри отдельного вычислительного узла считаются фиксированной величиной, не оказывающей существенного влияния на общую структуру расчета. Процесс исследования масштабируемости параллельных алгоритмов можно условно разбить на несколько этапов:

- Создание упрощенной модели программы. На данном этапе происходит анализ того, как разные части программы (потоки) работают и взаимодействуют между собой. Затем группируются похожие потоки, которые выполняют одинаковые вычисления. Для каждой группы создается виртуальный “шаблон” (актор), который будет имитировать ее поведение.
- Настройка виртуальных копий потоков. Каждый актор получает следующие параметры: время вычислений, объем передаваемых данных и скорость обмена сообщениями.
- Запуск имитации. Акторы определяют и устанавливают адресатов для взаимодействия, а также выполняют расчеты, обмениваясь сообщениями в соответствии с заданной схемой вычислений. Этот процесс продолжается до тех пор, пока не будет достигнуто условие останова (например, выполнится заданное число циклов вычислений).
- Анализ результатов. На данном этапе происходит исследование полученных показателей, таких как общее время работы вычислительного процесса, время выполнения каждого актора, количество итераций цикла вычислений, совершенных каждым актором, задержки при передаче данных между акторами. Описанный метод позволяет заранее узнать, как программа поведет себя на реальном суперкомпьютере, помогает найти “узкие места” в работе программы, сокращает временные и финансовые издержки (не нужно арендовать реальный суперкомпьютер для тестов), а также особенно полезен для разработки алгоритмов для суперкомпьютеров следующего поколения, которые еще находятся в стадии проектирования. Предлагаемый метод исследования вычислительных процессов уже был успешно использован при изучении возможности масштабирования нескольких вычислительных алгоритмов на большое число вычислительных ядер [19, 20].

3. Исследование масштабируемости параллельного алгоритма динамики плазмы.

3.1. Описание схемы вычислений для исследуемого алгоритма. Исходная физическая задача по исследованию динамики плазмы в магнитной ловушке описывается трехмерной гибридной моделью, где ионная компонента представляет собой набор частиц, а электронная — жидкость. В область ловушки с постоянным магнитным полем \mathbf{B}_0 и фоновой плазмой плотности n_0 инжектируются частицы/ Самосогласованное движение частиц в электромагнитных полях ловушки описывается кинетическим уравнением Власова для ионов:

$$\frac{\partial f}{\partial t} + \mathbf{v} \frac{\partial f}{\partial \mathbf{r}} + \frac{\mathbf{F}}{m} \frac{\partial f}{\partial \mathbf{v}} = 0,$$

где сила \mathbf{F} учитывает трение между электронами и ионами. При этом плотности и скорости ионов определяются как интегралы

$$n_i(\mathbf{r}, t) = \int f(\mathbf{r}, \mathbf{v}, t) d\mathbf{v}, \quad \mathbf{V}_i(\mathbf{r}, t) = \frac{1}{n_i} \int \mathbf{v} f(\mathbf{r}, \mathbf{v}, t) d\mathbf{v}.$$

Плазма считается квазинейтральной $n_e = n_i = n$, токи рассчитываются как $\mathbf{j} = e(n_i \mathbf{V}_i - n_e \mathbf{V}_e)$. Для описания электронов используются уравнения магнитной гидродинамики в безмассовом приближении:

$$e\mathbf{E} - \frac{e}{c} \mathbf{V}_e \times \mathbf{B} - \frac{\nabla p_e}{n_e} + \frac{m_e}{\tau_{ei}} (\mathbf{V}_i - \mathbf{V}_e) = 0,$$

где $p_e = n_e T_e$ — давление электронной компоненты, T_e — ее температура. При этом температура T_e удовлетворяет уравнению:

$$n_e \left(\frac{\partial T_e}{\partial t} + \mathbf{V}_e \nabla T_e \right) = (\gamma - 1) (Q_e - \operatorname{div} \mathbf{q}_e - p_e \operatorname{div} \mathbf{V}_e),$$

где Q_e — джоулев нагрев, \mathbf{q}_e — плотность теплового потока, $\gamma = 5/3$ — показатель адиабаты. Электромагнитные поля описываются уравнениями Максвелла с малыми токами смещения:

$$\nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{j}, \quad \nabla \times \mathbf{E} = -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t}.$$

В начальный момент времени $\mathbf{B} = (0, 0, B_{0z})$, $\mathbf{E} = \mathbf{0}$, $T_i = T_0$. Задача решается с помощью метода “частиц-в-ячейке” (PIC-метод). На рис. 3 представлено общее описание PIC-метода, которое подробно изложено в [21–23].

В параллельной реализации численной модели удержания плазмы в открытых магнитных ловушках используется гибридная декомпозиция расчетной области [24]. Расчетная область делится равномерно на подобласти, причем за каждую подобласть отвечает отдельная группа процессов. Частицы в подобласти распределяются равномерно между процессами группы. Передача граничных узлов для подобласти узлов сетки происходит между процессами основной группы (рис. 4). Частицы могут быть перенаправлены в любой процесс из соседней группы.

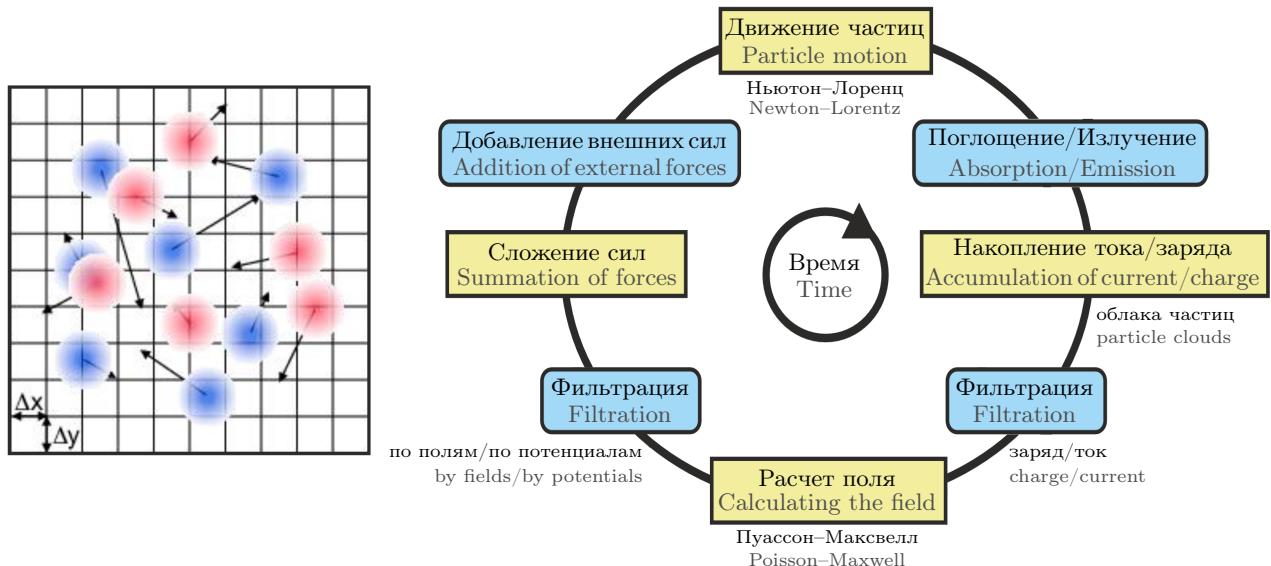


Рис. 3. Общее описание метода “частиц-в-ячейках”: а) пример перелета частиц между ячейками сетки;
б) общая схема метода

Fig. 3. General description of the particle-in-cell method: a) example of particle movement between grid cells;
b) general scheme of the method

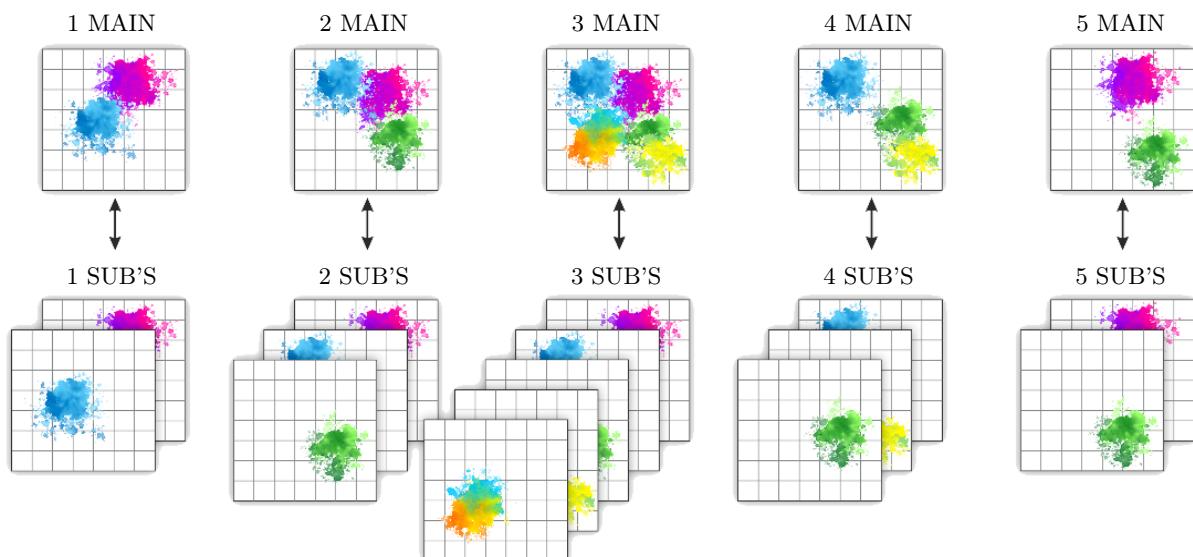


Рис. 4. Декомпозиция расчетной области в исследуемом алгоритме
удержания плазмы в открытых магнитных ловушках

Fig. 4. Domain decomposition in the studied algorithm of the plasma confinement
in open magnetic traps



Каждый этап вычислительного цикла включает в себя лагранжеву стадию для расчета скоростей и координат частиц и эйлерову стадию для вычисления на пространственной сетке. В начале каждого шага вычислительного цикла основной процесс распределяет по группе значения электрического и магнитного поля в подобласти, используя процедуру MPI BCAST. Чтобы перейти к лагранжевой стадии, мы применяем билинейную интерполяцию, находим в каждом процессе силу Лоренца, действующую на частицу в ее местоположении, и вычисляем скорости и координаты частиц. Если частица вылетает за пределы подобласти, то ее физические характеристики отправляются в один из процессов соседней группы. Использование локального ранжирования позволяет равномерно распределить частицы по процессам при отправке. После этого каждый процесс вычисляет средние скорости ионов в ячейке и плотность, и мы переходим к эйлерову этапу. Основные процессы группы собирают данные о группе, используя MPI REDUCE, а затем обмениваются граничными данными. Далее вычисляются скорости электронов, параметры электрического поля, магнитного поля и температуры. После решения каждого уравнения на эйлеровом этапе происходит обмен граничными узлами сетки между соседними процессорами основной группы. Исходя из описания параллельной реализации исследуемого алгоритма была составлена схема вычислений (рис. 5), на основе которой подготовлена имитационная модель для исследования его масштабируемости.

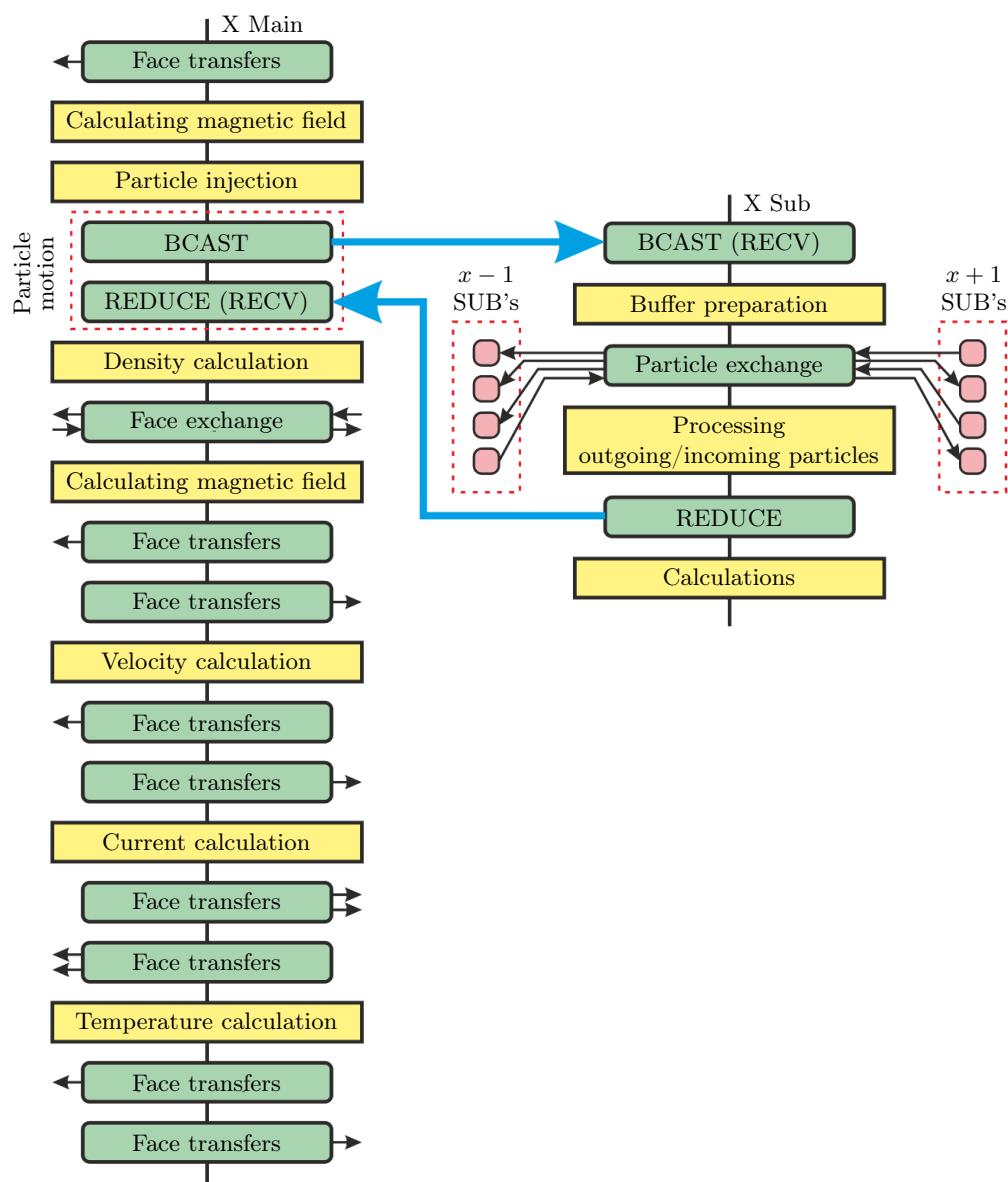


Рис. 5. Схема коммуникаций между вычислительными нитями исследуемого параллельного алгоритма

Fig. 5. Communication scheme between computational threads of the studied parallel algorithm

3.2. Исследование производительности алгоритма и оптимизация. Для того, чтобы провести исследование масштабируемости с использованием имитационного моделирования, необходимо набрать статистические данные по работе программы для различного количества MPI потоков. В табл. 1 показаны данные, собираемые для имитационного моделирования, где N_x , N_y , N_z , $N_{z_{loc}}$ — параметры

Таблица 1. Собираемые данные для имитационного моделирования

Table 1. Gathered data for the simulation modeling

Процедура Procedure	Пересылки или другие процедуры Transfers or other subprocedures	Размер пересылки Transfer size
1. Вычисление магнитного поля (часть 1) 1. Magnetic field calculation (part 1)	Пересылки граней MPI SEND, MPI RECV Face transfers MPI SEND, MPI RECV	$2 \times N_x \times N_y$ влево $2 \times N_x \times N_y$ to the left
2. Инжекция 2. Injection		
3. Движение частиц 3. Particle motion	Передача данных полей MPI BCAST Field data transfer MPI BCAST Подготовка пересылок: частицы, вылетающие из текущей области, попадают в буфер Transfer preparation: particles flying out of the current domain fall into the buffer	$3 \times N_x \times N_y \times N_{z_{loc}}$ $3 \times N_x \times N_y \times N_{z_{loc}}$ $3 \times N_x \times N_y \times N_{z_{loc}}$
	Пересылки частиц MPI SEND, MPI RECV Particle transfers MPI SEND, MPI RECV	Разный Different size
	Перераспределение частиц по ячейкам: включает перевыделение памяти, заполнение пустых элементов массива частиц, вылетевших из текущей области, и добавление прилетевших частиц в эту область Redistributing particles among cells: includes memory reallocation, filling empty elements of the particle array that flew out of the current domain and adding incoming particles to this domain	
4. Плотности 4. Densities	Сбор плотностей MPI REDUCE Density gathering MPI REDUCE	$3 \times N_x \times N_y \times N_{z_{loc}}$ $3 \times N_x \times N_y \times N_{z_{loc}}$ $3 \times N_x \times N_y \times N_{z_{loc}}$
	Пересылка граней MPI SEND, MPI RECV Face transfers MPI SEND, MPI RECV	$6 \times N_x \times N_y \times N_{z_{loc}}$ to the left $6 \times N_x \times N_y \times N_{z_{loc}}$ to the right $6 \times N_x \times N_y \times N_{z_{loc}}$ to the left $6 \times N_x \times N_y \times N_{z_{loc}}$ to the right $2 \times N_x \times N_y \times N_{z_{loc}}$ to the left $2 \times N_x \times N_y \times N_{z_{loc}}$ to the right
5. Вычисление магнитного поля (часть 2) 5. Magnetic field calculation (part 2)		
6. Токи 6. Currents	Пересылки граней MPI SEND, MPI RECV Face transfers MPI SEND, MPI RECV	$2 \times N_x \times N_y$ to the left $2 \times N_x \times N_y$ to the right
7. Скорости электронов 7. Electron velocities	Пересылки граней E_z MPI SEND, MPI RECV Face transfers E_z MPI SEND, MPI RECV	$2 \times N_x \times N_y$ to the left $2 \times N_x \times N_y$ to the right
8. Электрическое поле 8. Electric field	Пересылки граней E_x , E_y MPI SEND, MPI RECV Face transfers E_x , E_y MPI SEND, MPI RECV Пересылки граней E_z MPI SEND, MPI RECV Face transfers E_z MPI SEND, MPI RECV	$2 \times N_x \times N_y$ to the right $2 \times N_x \times N_y$ to the right $2 \times N_x \times N_y$ to the left $2 \times N_x \times N_y$ to the left $N_x \times N_y$ to the left $N_x \times N_y$ to the right
9. Температура 9. Temperature	Пересылки граней MPI SEND, MPI RECV Face transfers MPI SEND, MPI RECV	$N_x \times N_y$ to the left $N_x \times N_y$ to the right



групп данных, соответствующие размерам сетки всей области в направлениях x , y , z и локальному размеру сетки по направлению z для каждого процессора. Пересылки от i -ого процессора к $(i + 1)$ -му (то есть соседу справа) будем называть пересылками вправо, от i -го процессора к $(i - 1)$ -му — пересылками влево.

Чтобы выявить источник снижения производительности исследуемого алгоритма, мы анализируем различные показатели производительности программного кода, связанные с конкретной аппаратной архитектурой вычислительного комплекса. В нашем случае мы собирали данные с 8 до 32 MPI потоков для двухпроцессорного узла с 16 ядерными Intel Xeon 2697Av4 кластера НКС-1П ЦКП ССКЦ СО РАН. Значения задержек для разных типов интерконнектов известны из предыдущих расчетов. Конкретно для нашего исследования мы использовали их усредненные значения. Анализ производительности и последующая оптимизация вычислительного кода проводилась с использованием программы Intel VTune [25] из пакета IntelOneAPI 2024. Это свободно распространяемый пакет программ, который включает в себя как полный набор самых современных компиляторов, так и широкий круг средств отладки и оптимизации программ. Intel VTune анализирует производительность программы в части эффективности использования различных блоков процессора, подсистемы памяти и параллельной реализации кода.

На рис. 6 показана метрика использования микроархитектуры процессора. Показатель использования микроархитектуры — это ключевая метрика, позволяющая оценить (в %), насколько эффективно код выполняется на текущей микроархитектуре. На использование микроархитектуры могут повлиять операции с памятью, характеризующиеся большой задержкой, операции с плавающей запятой или SIMD, неисполненные инструкции из-за неправильного прогнозирования ветвей. На рис. 7 показана метрика ограничения производительности кода, вызванная работой подсистемы оперативной памяти. Эта метрика показывает, как проблемы подсистемы памяти влияют на производительность. Memory Bound определяет долю слотов, где конвейер может быть остановлен из-за инструкций по загрузке или сохранения по требованию. В нашем случае этот показатель превышает 20%. Это означает, что из-за обращений к памяти, процессор простояивает в ожидании загрузки или сохранения данных значительную часть времени. При этом нам удалось достичь неплохих показателей по использованию кэшей всех трех уровней за счет выравнивания данных и эффективного размера блоков данных, обрабатываемых за один такт процессора

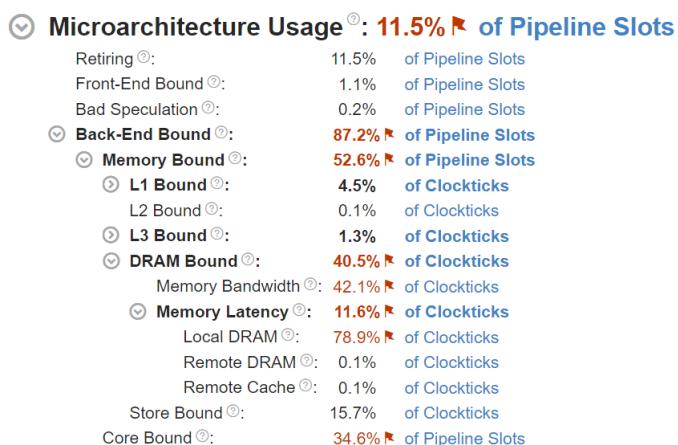


Рис. 6. Использование микроархитектуры

Fig. 6. Microarchitecture usage



Рис. 7. Ограничения, связанные с подсистемой памяти исследуемого параллельного алгоритма

Fig. 7. Memory subsystem limitations of the studied parallel algorithm

в случае применения SIMD инструкций. Что касается использования микроархитектуры, то анализ показывает, что задержки, связанные с использованием памяти, дают в целом невысокую эффективность использования вычислительных ядер. Это типичные результаты для кодов, связанных с решением данного класса задач. Доля ошибок, обусловленных неэффективным использованием инструкций вычислительных ядер, составляет 34.6% при целевом показателе 10%. Это указывает на значительный потенциал для оптимизации кода.

Результаты первоначальных тестов производительности были на 30% ниже, чем представленные на рис. 6, 7. Мы подобрали оптимальные коэффициенты, приведенные в табл. 1, чтобы данные для вычислений эффективно использовали кэш-память процессора, а также разделили арифметические операции, чтобы облегчить векторизацию кода компилятору Intel Fortran Compiler. К сожалению, в нашем распоряжении отсутствуют высокопроизводительные вычислительные системы, насчитывающие больше 50000 вычислительных ядер. В этом случае для исследования масштабируемости программы можно использовать имитационное моделирование, позволяющее представить программный код в виде “черных ящиков”. Опираясь на данные о задержках выполнения различных частей программы, вызванных работой памяти и сетевых устройств, мы можем построить модель ее исполнения на вычислительных системах с потенциально “бесконечным” количеством ядер.

3.3. Исследование масштабируемости параллельного алгоритма

Для имитации исполнения алгоритма численного моделирования динамики плазмы в открытых магнитных ловушках реализованы классы акторов PMain и PSub. Акторы класса PMain имитируют выполнение главного процесса группы: они осуществляют рассылку и сбор данных со всех процессов группы, а также обмениваются данными с соседними акторами того же класса. Акторы класса PSub имитируют движение частиц между группами и проводят остальные вычисления в группе. Задержки при передаче сообщений между акторами в итоге имитируют соответствующие задержки при передаче сообщений в реальной вычислительной системе. Модель исполнения алгоритма выполняет заранее заданное количество вычислительных циклов. Временные характеристики выполнения всех этапов вычислительных циклов получены на основе результатов запуска реального вычислительного алгоритма на кластере НКС-1П ЦКП ССКЦ СО РАН. Для демонстрации масштабируемости исследуемого алгоритма были произведены расчеты для различного числа вычислительных узлов (более 10⁶ вычислительных ядер). Исследована слабая масштабируемость вычислительного алгоритма — изменение времени выполнения задачи при увеличении параллельных процессов (вычислительных ядер) при условии сохранения фиксированной вычислительной нагрузки в пересчете на один процесс. Результаты данных исследований приведены на рис. 8. Для идеального параллельного алгоритма время исполнения при этом должно меняться незначительно. Однако, стремительный рост нагрузки на систему обмена сообщениями приводит к тому, что после достижения определенного количества вычислительных ядер их дальнейшее увеличение приводит к неоправданно высоким затратам вычислительных и сетевых ресурсов по сравнению с достигаемым выигрышем во времени. Поиск такого количества вычислительных ресурсов с помощью имитационного моделирования существенно снижает нагрузку как на разработчика вычислительных алгоритмов, так и на вычислительную инфраструктуру.

Исходя из рис. 8 можно сделать вывод, что использование в схеме вычислений главного процесса группы, который занимается рассылкой и агрегацией данных группы и обменом данными с соседними областями, существенно снижает издержки на коммуникации. Также можно отметить, что алгоритм при исполнении на большем числе вычислительных ядер демонстрирует высокую масштабируемость, то есть способность эффективно решать задачи значительно большего размера.

4. Заключение. В рамках интегрального подхода к параллельному программированию разработана инновационная методика анализа масштабируемости, основанная на мультиагентной имитацион-

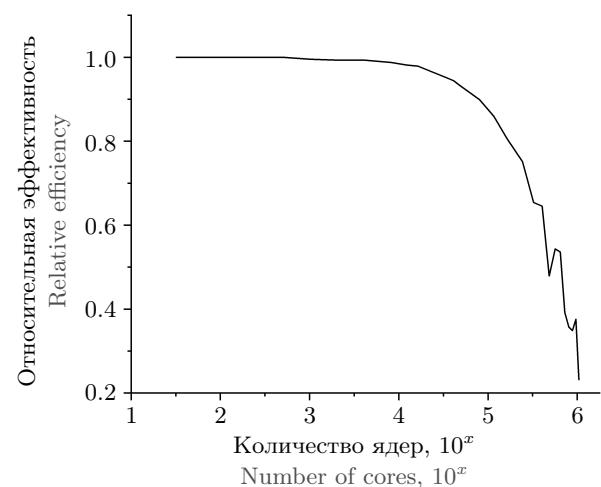


Рис. 8. Масштабируемость исследуемого алгоритма

Fig. 8. Scalability of the studied algorithm



ной модели, принципах дискретного событийного моделирования и технологиях прогнозной аналитики. Предлагаемая методика позволяет осуществлять предиктивный анализ масштабируемости алгоритмов, минимизировать потребности в реальных вычислительных ресурсах, т.е. снижать материальные затраты на этапе проектирования и оценивать перспективы использования алгоритмов в экзафлопсных системах. Данная методика расширяет инструментарий НРС-аналитики и позволяет избегать узких мест на этапе проектирования. Исследован алгоритм численного моделирования динамики плазмы в аксиально-симметричных открытых магнитных ловушках. Особенностью данного алгоритма является локальность вычислительных операций, отсутствие глобальных коммуникационных операций и оптимизированная схема обмена данными. Исследована сильная масштабируемость данного алгоритма на конфигурациях до 1 млн ядер. При решении задач физики плазмы, которые эффективно загружают данными до 100 тысяч ядер, достигается сохранение эффективности на уровне 85%.

Список литературы

1. Glinsky B., Kulikov I., Chernykh I., Snytnikov A., Sapetina A., Weins D. The Integrated Approach to Solving Large-Size Physical Problems on Supercomputers // Proc. *Supercomputing. RuSCDays 2017*. Eds. Voevodin V., Sobolev S. (CCIS, Springer, Cham, 2017). Vol. 793, 278–289. doi [10.1007/978-3-319-71255-0_22](https://doi.org/10.1007/978-3-319-71255-0_22).
2. Dosanjh S.S., Barrett R.F., Doerfler D.W., et al. Exascale design space exploration and co-design // Future Generation Computer Systems. 2014. **30** (1). 46–58. doi [10.1016/j.future.2013.04.018](https://doi.org/10.1016/j.future.2013.04.018).
3. Czarnul P., Kuchta J., Rosciszewski P., Proficz J. Modeling energy consumption of parallel applications // Proc. in Federated Conference on Computer Science and Information Systems, Gdańsk, Poland, September 11–14, 2016. 855–864. doi [10.15439/2016F308](https://doi.org/10.15439/2016F308).
4. Shubhangi D., Bushra S. Deploying Efficiently MapReduce Applications in Heterogeneous Computing Environments Using Novel Scheduling Algorithms // International Journal of Scientific Engineering and Research (IJSER). 2015. **5**, N 7. 61–71. doi [10.70729/IJSER171622](https://doi.org/10.70729/IJSER171622).
5. Coelho M., Ocaña K., Pereira A. et al. Machine Learning Regression-Based Prediction for Improving Performance and Energy Consumption in HPC Platforms // In: High Performance Computing. CARLA 2024. Eds. Guerrero G., San Martín J., Meneses E., Barrios Hernández C.J., Osthoff C., Monsalve Diaz J.M. (CCIS, Springer, Cham, 2025). Vol. 2270, 186–200. doi [10.1007/978-3-031-80084-9_13](https://doi.org/10.1007/978-3-031-80084-9_13).
6. Casanova H., Legrand A., Quinson M. SimGrid: A Generic Framework for Large-Scale Distributed Experiments // Proc. in Tenth International Conference on Computer Modeling and Simulation (uksim 2008), April 1–3, 2008. (Cambridge, UK, 2008). doi [10.1109/UKSIM.2008.28](https://doi.org/10.1109/UKSIM.2008.28).
7. Hsieh M., Riesen R., Thompson K., Song W., Rodrigues A. SST: A Scalable Parallel Framework for Architecture-Level Performance, Power, Area and Thermal Simulation // Comput. J. 2011. **55**, N 2. 181–191. doi [10.1093/comjnl/bxr069](https://doi.org/10.1093/comjnl/bxr069).
8. Podkorytov D., Rodionov A., Choo H. Agent-based Simulation System AGNES for Networks Modeling: Review and Researching // Proc. of the 6th Int. Conference on Ubiquitous Information Management and Communication (ACM ICUIMC 2012), Malaysia, Kuala Lumpur, February 20–22, 2012. Article No. 115. 1–4. doi [10.1145/2184751.2184883](https://doi.org/10.1145/2184751.2184883).
9. Weins D.V., Glinsky B.M., Chernykh I.G. Analysis of Means of Simulation Modeling of Parallel Algorithms // Proc. in Supercomputing, RuSCDays 2018, Eds. Voevodin V., Sobolev S. Moscow, Russia, September 24–25, 2018. CCIS. 2019. **965**. 29–39. doi [10.1007/978-3-030-05807-4_3](https://doi.org/10.1007/978-3-030-05807-4_3).
10. Cesarini F., Thompson S. Erlang Programming. O'Reilly, 2009.
11. Index - Erlang/OTP. <http://www.erlang.org/>. (Дата обращения: 18 декабря 2025).
12. Ромов Д.Д. Открытые ловушки // УФН. 1988. **154**, № 4. 565–614. doi [10.3367/UFNr.0154.198804b.0565](https://doi.org/10.3367/UFNr.0154.198804b.0565).
13. Chernykh I., Kulikov I., Vshivkov V., Genrikh E., Weins D., Dudnikova G., Chernoshtanov I., Boronina M. Energy Efficiency of a New Parallel PIC Code for Numerical Simulation of Plasma Dynamics in Open Trap // Mathematics. 2022. **10**, N 19. 3684. doi [10.3390/math10193684](https://doi.org/10.3390/math10193684).
14. Glinsky B., Rodionov A., Marchenko M., Podkorytov D., Weins D. Scaling the Distributed Stochastic Simulation to Exaflop Supercomputers // Proceedings of the 14th International Conference on High Performance Computing and Communication & IEEE 9th International Conference on Embedded Software and Systems (HPCC–2012), Liverpool, UK. 2012. 1131–1136. doi [10.1109/HPCC.2012.166](https://doi.org/10.1109/HPCC.2012.166).

15. Мигов Д.А., Винс Д.В. Параллельная реализация и имитационное моделирование оценки надежности сети методом Монте-Карло // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2019. № 47. 66–74. http://vital.lib.tsu.ru/vital/access/manager/Repository/vts_s:000661496. (Дата обращения: 18 декабря 2025).
16. Chernoshtanov I.S., Chernykh I.G., Dudnikova G.I., Boronina M.A., Liseykina T.V., Vshivkov V.A. Effects Observed in Numerical Simulation of High-Beta Plasma with Hot Ions in an Axisymmetric Mirror Machine // Journal of Plasma Physics. 2024. **90**, N 2. Article No. 905900211. doi [10.1017/S0022377824000333](https://doi.org/10.1017/S0022377824000333).
17. Glinsky B., Sapetina A., Martynov V., Weins D., Chernykh I. The Hybrid-Cluster Multilevel Approach to Solving the Elastic Wave Propagation Problem // Parallel Computational Technologies (PCT 2017). Eds. Sokolinsky L., Zymbler M. Cham: Springer, 2017. **753**. 261–274. doi [10.1007/978-3-319-67035-5_19](https://doi.org/10.1007/978-3-319-67035-5_19)
18. Kulikov I., Chernykh I., Glinsky B., Weins D., Shmelev A. Astrophysics simulation on RSC massively parallel architecture // Proceedings of 2015 IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2015). 2015. 1131–1134. doi [10.1109/CCGrid.2015.102](https://doi.org/10.1109/CCGrid.2015.102).
19. Боронина М.А., Куликов И.М., Черных И.Г., Винс Д.В. Использование комбинации схем Рое и Русанова для численного решения уравнений гидродинамики в задачах космической плазмы // Сибирский журнал индустриальной математики. 2022. **25**, № 4. 14–26. [http://old.math.nsc.ru/publishing/SIBJIM/92\(4\)/402.pdf](http://old.math.nsc.ru/publishing/SIBJIM/92(4)/402.pdf). (Дата обращения: 18 декабря 2025).
20. Weins D., Vorobyev V., Chernykh I., Logashenko I. Development of simulation model of HPC system for Super Charm-Tau factory // Journal of Physics: Conference Series. 2019. **1336**. Article No. 012025. doi [10.1088/1742-6596/1336/1/012025](https://doi.org/10.1088/1742-6596/1336/1/012025).
21. Particle-in-Cell Method — WarpX 22.11 documentation. https://warpx.readthedocs.io/en/22.11/theory/picsar_theory.html. (Дата обращения: 18 декабря 2025).
22. Boronina M.A., Chernoshtanov I.S., Chernykh I.G., et al. Three-Dimensional Model for Numerical Simulation of Beam-Plasma Dynamics in Open Magnetic Trap // Lobachevskii J. Math. 2024. **45**, 1–11. doi [10.1134/S1995080224010074](https://doi.org/10.1134/S1995080224010074).
23. Hockney R.W., Eastwood J.W. Computer Simulation Using Particles. New York: Taylor & Francis Group, 1988. doi [10.1201/9780367806934](https://doi.org/10.1201/9780367806934).
24. Киреев С.Е. Параллельная реализация метода частиц в ячейках для моделирования задач гравитационной космодинамики // Автометрия. 2006. **42**, № 3. 32–39. <https://sibran.ru/upload/iblock/d40/d4055c89a69fe12026b16920879c8ba6.pdf>. (Дата обращения: 18 декабря 2025).
25. Fix Performance Bottlenecks with Intel® VTune™ Profiler. <https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html>. (Дата обращения: 18 декабря 2025).

Получена
25 сентября 2025 г.

Принята
1 декабря 2025 г.

Опубликована
22 декабря 2025 г.

Информация об авторах

Игорь Геннадьевич Черных — к.ф.-м.н., заведующий лабораторией; Институт вычислительной математики и математической геофизики СО РАН, пр-кт Академика Лаврентьева, 6, 630090, Новосибирск, Российская Федерация.

Дмитрий Владимирович Винс — к.ф.-м.н., Институт вычислительной математики и математической геофизики СО РАН, пр-кт Академика Лаврентьева, 6, 630090, Новосибирск, Российская Федерация.

Виталий Андреевич Вшивков — д.ф.-м.н., гл. науч. сотр.; Институт вычислительной математики и математической геофизики СО РАН, пр-кт Академика Лаврентьева, 6, 630090, Новосибирск, Российская Федерация.

Марина Андреевна Боронина — к.ф.-м.н., Институт вычислительной математики и математической геофизики СО РАН, пр-кт Академика Лаврентьева, 6, 630090, Новосибирск, Российская Федерация.



References

1. B. Glinsky, I. Kulikov, I. Chernykh, A. Snytnikov, A. Sapetina, D. Weins, “The Integrated Approach to Solving Large-Size Physical Problems on Supercomputers,” in *V. Voevodin, S. Sobolev (eds) Supercomputing, RuSCDays 2017, Moscow, Russia, September 25–26, 2017.* (CCIS, Springer, Cham, 2017). Vol. 793, 278–289. doi [10.1007/978-3-319-71255-0_22](https://doi.org/10.1007/978-3-319-71255-0_22).
2. S. S. Dosanjh, R. F. Barrett, D. W. Doerfler, et al., “Exascale design space exploration and co-design,” *Future Generation Computer Systems.* **30** (1), 46–58 (2014). doi [10.1016/j.future.2013.04.018](https://doi.org/10.1016/j.future.2013.04.018).
3. P. Czarnul, J. Kuchta, P. Rosciszewski, J. Proficz, “Modeling energy consumption of parallel applications,” Proc. in *M. Ganzha, L. Maciaszek, M. Paprzycki (eds) Federated Conference on Computer Science and Information Systems, Gdańsk, Poland, September 11–14, 2016.* Annals of Computer Science and Information Systems, Vol. 8, 855–864. doi [10.15439/2016F308](https://doi.org/10.15439/2016F308).
4. D. Shubhangi, S. Bushra, “Deploying Efficiently MapReduce Applications in Heterogeneous Computing Environments Using Novel Scheduling Algorithms,” *International Journal of Scientific Engineering and Research (IJSER).* **5** (7), 61–71 (2017). doi [10.70729/IJSER171622](https://doi.org/10.70729/IJSER171622).
5. M. Coelho, K. Ocaña, A. Pereira, et al., “Machine Learning Regression-Based Prediction for Improving Performance and Energy Consumption in HPC Platforms,” in *Guerrero G., San Martín J., Meneses E., Barrios Hernández C.J., Osthoff C., Monsalve Diaz J.M. (eds) High Performance Computing, CARLA 2024.* (CCIS, Springer, Cham, 2025). Vol. 2270, 186–200. doi [10.1007/978-3-031-80084-9_13](https://doi.org/10.1007/978-3-031-80084-9_13).
6. H. Casanova, A. Legrand, and M. Quinson, “SimGrid: A Generic Framework for Large-Scale Distributed Experiments,” Proc. in *Tenth International Conference on Computer Modeling and Simulation (UKSim 2008), April 1–3, 2008.* (Cambridge, UK, 2008). 126–131. doi [10.1109/UKSIM.2008.28](https://doi.org/10.1109/UKSIM.2008.28)
7. M. Hsieh, R. Riesen, K. Thompson, W. Song, A. Rodrigues, “SST: A Scalable Parallel Framework for Architecture-Level Performance, Power, Area and Thermal Simulation,” *Comput. J.* **55** (2), 181–191 (2012). doi [10.1093/comjnl/bxr069](https://doi.org/10.1093/comjnl/bxr069).
8. D. Podkorytov, A. Rodionov, H. Choo, “Agent-based simulation system AGNES* for networks modeling: review and researching,” *Proc. of the 6th Int. Conference on Ubiquitous Information Management and Communication (ACM ICUIMC 2012), Kuala Lumpur, Malaysia, February 20–22, 2012.* (ACM, New York, NY, USA). Article No. 115, 1–4. doi [10.1145/2184751.2184883](https://doi.org/10.1145/2184751.2184883).
9. D. V. Weins, B. M. Glinsky, I. G. Chernykh, “Analysis of Means of Simulation Modeling of Parallel Algorithms,” in *V. Voevodin, S. Sobolev (eds) Supercomputing, RuSCDays 2018, Moscow, Russia, September 24–25, 2018.* (CCIS, Springer, Cham, 2019). Vol. 965, 29–39. doi [10.1007/978-3-030-05807-4_3](https://doi.org/10.1007/978-3-030-05807-4_3).
10. F. Cesarini, S. Thompson, *Erlang Programming* (O’Reilly, 2009).
11. Index - Erlang/OTP. <http://www.erlang.org/>. Cited December 18, 2025.
12. D. D. Ryutov, “Open-ended traps,” *Soviet Physics Uspekhi.* **31** (4), 300 (1988). doi [10.1070/PU1988v03n04ABEH005747](https://doi.org/10.1070/PU1988v03n04ABEH005747).
13. I. Chernykh, I. Kulikov, V. Vshivkov, E. Genrikh, D. Weins, G. Dudnikova, I. Chernoshtanov, M. Boronina, “Energy Efficiency of a New Parallel PIC Code for Numerical Simulation of Plasma Dynamics in Open Trap,” *Mathematics.* **10** (19), 3684 (2022). doi [10.3390/math10193684](https://doi.org/10.3390/math10193684).
14. B. Glinsky, A. Rodionov, M. Marchenko, D. Podkorytov, D. Weins, “Scaling the Distributed Stochastic Simulation to Exaflop Supercomputers,” *Proceedings of the 14th International Conference on High Performance Computing and Communication & IEEE 9th International Conference on Embedded Software and Systems (HPCC-2012), Liverpool, UK, June 25–27, 2012.* (Liverpool, UK, 2012). pp. 1131–1136. doi [10.1109/HPCC.2012.166](https://doi.org/10.1109/HPCC.2012.166).
15. D. A. Migov, D. V. Weins, “Parallel implementation and simulation of network reliability calculation by Monte Carlo method,” *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naja tekhnika i informatika [Tomsk State University Journal of Control and Computer Science].* **47**, 66–74 (2019). [in Russian]. doi [10.17223/19988605/47/8](https://doi.org/10.17223/19988605/47/8).
16. I. S. Chernoshtanov, I. G. Chernykh, G. I. Dudnikova, M. A. Boronina, T. V. Liseykina, and V. A. Vshivkov, “Effects Observed in Numerical Simulation of High-Beta Plasma with Hot Ions in an Axisymmetric Mirror Machine,” *Journal of Plasma Physics.* **90** (2), Article No. 905900211 (2024). doi [10.1017/S0022377824000333](https://doi.org/10.1017/S0022377824000333).
17. B. Glinsky, A. Sapetina, V. Martynov, D. Weins, I. Chernykh, “The Hybrid-Cluster Multilevel Approach to Solving the Elastic Wave Propagation Problem,” in *L. Sokolinsky, M. Zymbler (eds) Parallel Computational Technologies. PCT 2017.* (CCIS, Springer, Cham, 2017). Vol. 753, 261–274. doi [10.1007/978-3-319-67035-5_19](https://doi.org/10.1007/978-3-319-67035-5_19).

18. I. Kulikov, I. Chernykh, B. Glinskiy, D. Weins, and A. Shmelev, “Astrophysics simulation on RSC massively parallel architecture,” Proc. in *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2015, Shenzhen, China, May 4–7, 2015.* pp. 1131–1134. doi [10.1109/CCGrid.2015.102](https://doi.org/10.1109/CCGrid.2015.102).
19. M. A. Boronina, I. M. Kulikov, I. G. Chernikh, D. V. Vins, “Using a Combination of Roe and Rusanov Schemes for the Numerical Solution of the Equations of Magnetohydrodynamics in the Problems of Cosmic Plasma,” *J. Appl. Ind. Math.* **16** (4), 596–605 (2022). doi [10.1134/S1990478922040020](https://doi.org/10.1134/S1990478922040020).
20. D. Weins, V. Vorobyev, I. Chernykh, I. Logashenko, “Development of simulation model of HPC system for Super Charm-Tau factory,” *Journal of Physics: Conference Series*, **1336** (1), Article No. 012025 (2019). doi [10.1088/1742-6596/1336/1/012025](https://doi.org/10.1088/1742-6596/1336/1/012025).
21. Particle-in-Cell Method — WarpX 22.11 documentation. https://warpx.readthedocs.io/en/22.11/theory/pic_sar_theory.html. Cited December 18, 2025.
22. M. A. Boronina, I. S. Chernoshtanov, I. G. Chernykh, et al., “Three-Dimensional Model for Numerical Simulation of Beam-Plasma Dynamics in Open Magnetic Trap,” *Lobachevskii J. Math.* **45**, 1–11 (2024). doi [10.1134/S1995080224010074](https://doi.org/10.1134/S1995080224010074).
23. R. W. Hockney, J. W. Eastwood, *Computer Simulation Using Particles* (Taylor & Francis Group, NY, USA, 1988). doi [10.1201/9780367806934](https://doi.org/10.1201/9780367806934).
24. S. E. Kireev, “Parallel Realization of the Particle-in-Cell Method for Simulating Gravitational Cosmodynamics Problems,” *Avtometriya*, No. 3. 3–35 (2006). <https://sibran.ru/upload/iblock/d40/d4055c89a69fe12026b16920879c8ba6.pdf>. [in Russian]. Cited December 18, 2025.
25. Fix Performance Bottlenecks with Intel® VTune™ Profiler. <https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html>. Cited December 18, 2025.

Received
September 25, 2025

Accepted
December 1, 2025

Published
December 22, 2025

Information about the authors

Igor G. Chernykh — Ph.D., Head of the Laboratory; Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Ac. Lavrentiev prospekt, 6, 630090, Novosibirsk, Russia.

Dmitry V. Vins — Ph.D., Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Ac. Lavrentiev prospekt, 6, 630090, Novosibirsk, Russia.

Vitaly A. Vshivkov — Dr. Sci., Chief Researcher; The Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Ac. Lavrentiev prospekt, 6, 630090, Novosibirsk, Russia.

Marina A. Boronina — Ph.D., The Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Ac. Lavrentiev prospekt, 6, 630090, Novosibirsk, Russia.