

УСКОРЕНИЕ РАСЧЁТНОГО КОДА ODETTA ДЛЯ РЕШЕНИЯ ЗАДАЧ ТРАНСПОРТА ЧАСТИЦ

А.В. Шошина*, В.И. Белоусов**

* *Национальный исследовательский ядерный университет «МИФИ»
115409, Москва, Каширское шоссе, д. 31*

** *Национальный исследовательский центр «Курчатовский институт»
123182, Москва, пл. Курчатова, 1*



При математическом моделировании ядерных реакторов на быстрых нейтронах возникает необходимость проводить высокоточные расчёты задач защиты на неструктурированных сетках. В работе рассматривается и анализируется параллельная версия расчётного кода ПС ODETTA [1] с применением технологии библиотеки MPI (Message Passing Interface) [2]. Код предназначен для численного моделирования нейтронно-физических процессов в защитных композициях реакторных установок на быстрых нейтронах (в том числе и с жидкометаллическим теплоносителем) в режимах нормальной эксплуатации, а также может применяться при расчётах радиационных условий использования элементов конструкции и оборудования объектов атомной энергетики, являющихся источниками и (или) находящимися под воздействием ионизирующих излучений при обосновании безопасности. Проводится сравнение работы полученной программы с предыдущей версией. Описана разработка алгоритмической части программы ODETTA с применением MPI. Представлены особенности и специфика распараллеливания программы, приведена модификация расчётного кода, рассмотрены соответствующие алгоритмы. Дано краткое описание структуры модулей программы ODETTA с использованием MPI. Представлены результаты работы последовательной и параллельной версий программы ODETTA в ОС Linux [3] на HPC-кластере НИЯУ МИФИ [4]. Приведён сравнительный анализ двух программных реализаций с точки зрения скорости и точности результатов при использовании двух различных кластеров и разного количества узлов на них. Отмечены особенности выполнения вычислений на кластере.

Ключевые слова: параллельное программирование, MPI, программа ODETTA, метод конечных элементов, радиационная безопасность, HPC-кластер.

ВВЕДЕНИЕ

Цель работы – применить параллельные вычисления для решения задач переноса нейтронов и гамма-квантов в многогрупповом SnPm-приближении методом конечных элементов на неструктурированных тетраэдральных сетках, включая работу с сеточными данными. Работа проводится в рамках исследования ИБРАЭ РАН в вычислительном центре НИЯУ МИФИ. Крайне сложные задачи возникают при математическом моделировании ядерных реакторов на быстрых нейтронах. Именно при

© *А.В. Шошина, В.И. Белоусов, 2020*

решении таких проблем необходимо проводить массовые высокоточные расчёты задач защиты на неструктурированных сетках. Аналогом является программа ATTILA [5], использующая линейный метод конечных элементов DFEM для решения уравнения переноса на пространственных сетках, состоящих из четырёхгранных и шестигранных конечных элементов с неплоскими гранями.

ОСОБЕННОСТИ РАСЧЁТНОГО КОДА ODETTA

При математическом моделировании ядерных реакторов на быстрых нейтронах возникает необходимость проводить высокоточные расчёты задач защиты на неструктурированных сетках. Программа ODETTA в модуле реализации нейтронно-физического расчёта моделирует решение уравнения переноса методом конечных элементов (МКЭ) на неструктурированных тетраэдральных сетках в приближении метода дискретных ординат [6].

Уравнение переноса с использованием дискретных ординат записывается в виде

$$\mu_m \frac{\partial \Psi_m}{\partial x} + \eta_m \frac{\partial \Psi_m}{\partial y} + \xi_m \frac{\partial \Psi_m}{\partial z} + \Sigma_T(x, y, z) \cdot \Psi_m(x, y, z) = Q_m(x, y, z),$$

где индекс номера группы g опущен, а индексу m ($m = 1, 2, \dots, M$) соответствует дискретное направление $\Omega_m = (\mu_m, \eta_m, \xi_m)$ из квадратурного набора (с равными весами ESn или Чебышева-Лежандра CLn), причём величина поверхности единичной сферы измеряется в единицах 4π , т.е. $\Sigma\omega_m = 1$. Формулы МКЭ выводятся аппроксимацией уравнения переноса по Галёркину методом взвешенных невязок. В уравнение входит полное макроскопическое сечение взаимодействия Σ_T и функция Q_m – правая часть уравнения переноса, зависящая от решения Ψ_m . В правую часть входят источник межгрупповых и внутригрупповых переходов, источник деления и заданный внутренний источник. На границе Γ рассматриваемой трёхмерной области задаются нулевые значения углового потока для направлений внутри области либо условия отражения. В результате получается краевая задача для решения уравнения переноса частиц в выпуклой трёхмерной области.

Анизотропное рассеяние представляется разложением в ряд по присоединённым функциям Лежандра до пятого порядка. Для ускорения сходимости внутренних итераций используется метод пространственного ребаланса. Расчётный код разработан на языке Fortran (стандарт Fortran 90 и выше) с применением распараллеливания по OpenMP. Этапы работы с расчётным кодом ODETTA таковы:

- подготовка сеточных данных;
- подготовка групповых макросечений;
- формирование источника излучения;
- нейтронно-физический расчёт;
- расчёт функционалов.

Для работы с CAD-моделями и неструктурированными сетками используется программа SALOME [7]. Для подготовки групповых констант используется программа CONSYST с библиотекой БНАБ-РФ [8]. Для обеспечения наиболее комфортной работы пользователя с программой ODETTA используется препроцессор SIEGFRIED, разработанный в ИБРАЭ РАН специально для расчётного кода ODETTA, включающего в себя модули Readconst (интерфейс между форматом ANISN хранения многогрупповых констант и внутренним форматом кода ODETTA), Source (формирование источника излучения в исходной геометрии и интерполяция в узлах неструктурированной сетки), MKE3D (модуль нейтронно-физического расчёта) и RECONST (пост-процессинг, расчёт функционалов).

Визуализация решения, полученного после моделирования в ODETTA, может осуществляться графическим 3D-построителем, например, программой VisIt [9].

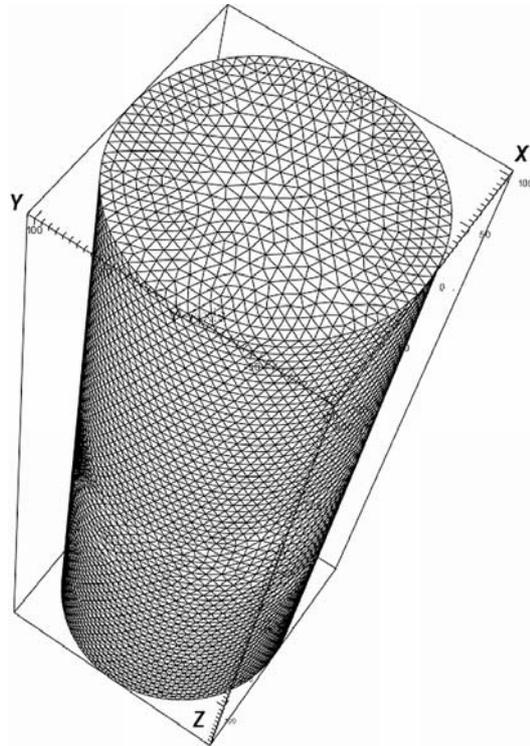


Рис. 1. Пример разбиения области на конечное число тетраэдров

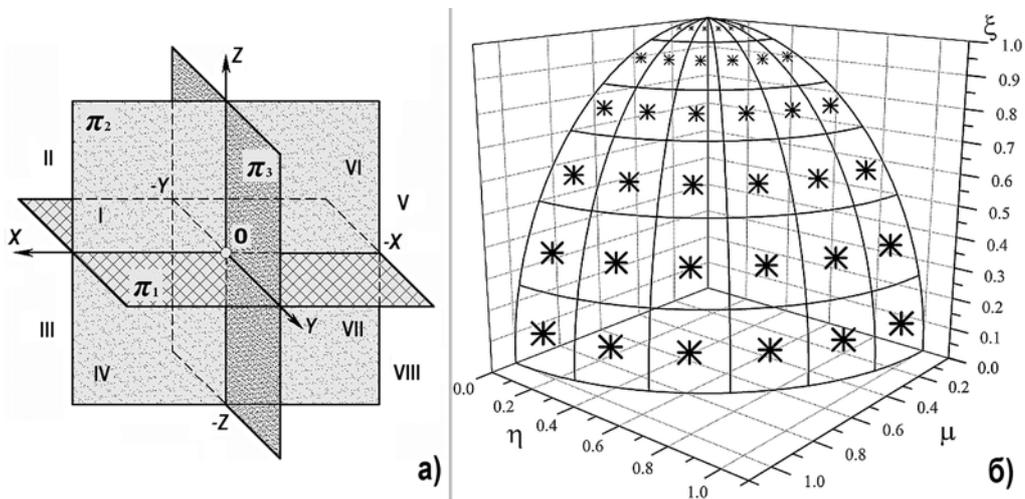


Рис. 2. Разбиения единичной сферы направлений на октанты (а); единичной сферы по направлениям в первом октанте (б)

Рассматриваемая область пространства разбивается на конечное число элементов при фиксированном числе конечных точек, называемых узлами. От качества разбиения во много зависит точность получаемых результатов. В качестве элементов в расчёте используются тетраэдры. Они имеют общие узловые точки и в совокупности аппроксимируют форму области. Обычно разбиение начинают с границ области с целью наиболее точной аппроксимации формы границы (рис. 1).

Рассматриваемую область и тетраэдры можно разделить по направлениям. Единичная сфера направлений (сетка по угловой переменной) разбита на восемь ок-

тантов (рис. 2а). Так как расчёт в каждом из них можно проводить независимо, то использование параллельных вычислений заметно ускорит время работы программы. Это является основой использования MPI в нейтронно-физическом модуле расчёта программы ODETTA.

Для проведения расчёта в каждом октанте используют разбиение по направлениям (рис. 2б) – оно дано на примере первого октанта. Такое дробление пространства необходимо для ускорения нейтронно-физического расчёта программы ODETTA с помощью OpenMP [10]. Разбиение единичной сферы направлений на октанты, а также независимость вычислений в каждом из них позволяют проводить расчёты параллельно и независимо по каждому направлению заданного пространства, используя MPI. Таким образом между октантами использовалось распараллеливание через MPI, а внутри октантов – через OpenMP.

РАЗРАБОТКА АЛГОРИТМИЧЕСКОЙ ЧАСТИ ПРИМЕНЕНИЯ MPI ДЛЯ РАСЧЁТНОГО КОДА ODETTA

Рост интереса к параллельному программированию в наши дни связан с переходом к массовому производству многоядерных архитектур [11]. Технология MPI представляет собой набор утилит и библиотечных функций (для языков C/C++ [12], Fortran [13 – 17]), позволяющих создавать и запускать приложения, работающие на параллельных вычислительных установках различной природы.

Исходный код программы ODETTA создан на языке Fortran, как и большинство современных программ, используемых в атомной отрасли. Многие из них не имеют возможности использовать более одного процесса, поэтому для ускорения их работы необходимо использовать преобразование к параллельным вычислениям. Это можно сделать двумя способами.

Первый способ – использовать компилятор, при этом разработчику необходимо указывать область кода, где именно следует применять распараллеливание внутри текста программы. Такой подход используется в системах OpenMP, но возможен лишь в вычислительных системах с общей памятью. Этот вариант распараллеливания уже был внедрён ранее в расчётный код ODETTA.

Второй подход заключается в том, что разработчик сам прописывает в коде программы распределение процессов и коммуникацию между ними. В новой версии кода программы ODETTA использовался второй подход. В этом разделе описываются изменения, внесённые в исходный код программы ODETTA для её новой реализации с технологией MPI.

Разбиение пространства на восемь октантов, расчёт в которых можно выполнять независимо, позволяет использовать несколько вычислительных узлов для оптимизации времени работы расчётного кода. Алгоритм расчёта кода ODETTA выполняется параллельно, и теоретически чем больше используется узлов на кластере, тем меньше время моделирования задач. В исходном коде программы использовался один вычислительный узел, а расчёт по октантам осуществлялся последовательно.

При тестировании полученной программы использовались два и четыре узла кластера НИЯУ МИФИ, хотя самым оптимальным вариантом предполагался кластер, состоящий из восьми узлов. Возможность использования MPI существенно сокращает время расчётов, так как все вычисления производятся параллельно. Разные количества узлов и соответствующие им распределения процессов показаны на рис. 3.

Предполагались вычисления на восьми октантах, поэтому при реализации программного кода максимально использовались восемь процессов, распределённых по узлам, запрашиваемым на кластере. Возможность параллельных вычислений на процессах может дать ощутимое преимущество в скорости выполнения расчётов.

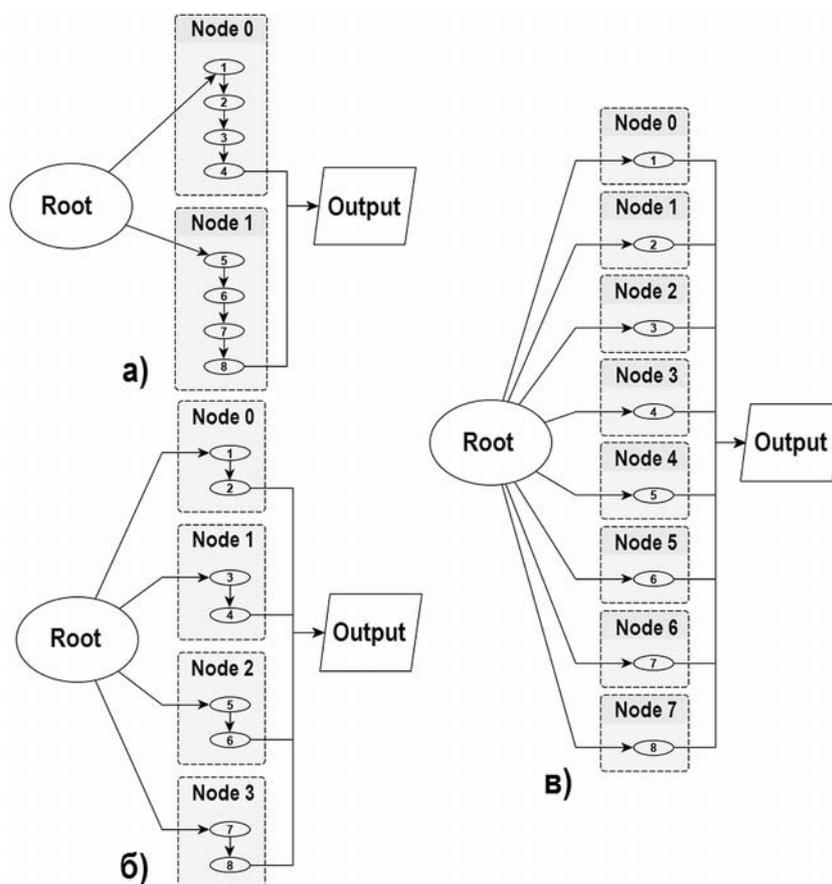


Рис. 3. Распределение октантов по вычислительным узлам: а) – два узла; б) – четыре узла; в) – восемь узлов

Для реализации расчётного кода программы ODETTA с использованием MPI подключались директивы, управляющие работой компилятора, что позволило расширить возможности расчётного кода.

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

Итоговый алгоритм программы ODETTA с использованием MPI представлен блок-схемой с общепринятыми в структурном программировании стандартными обозначения: овал – начало или конец рассматриваемой программной единицы; прямоугольник – блок операций; шестиугольник – цикл алгоритма, содержащий «тело» повторяющихся операций, один вход и один выход (рис. 4).

Блоки использования MPI в реализованном алгоритме выделены серым фоном и текстом, набранным курсивом. Изменения, в основном, касаются использования данных при взаимодействии между параллельными процессами.

РЕЗУЛЬТАТЫ

В качестве тестовой задачи для расчётного кода ODETTA была взята задача радиационной безопасности (РБ), в которой моделировалась топливная сборка бенчмарк-модели реактора MOX-1000 на быстрых нейтронах со смешанным оксидным уран-плутониевым топливом и натриевым теплоносителем [18].

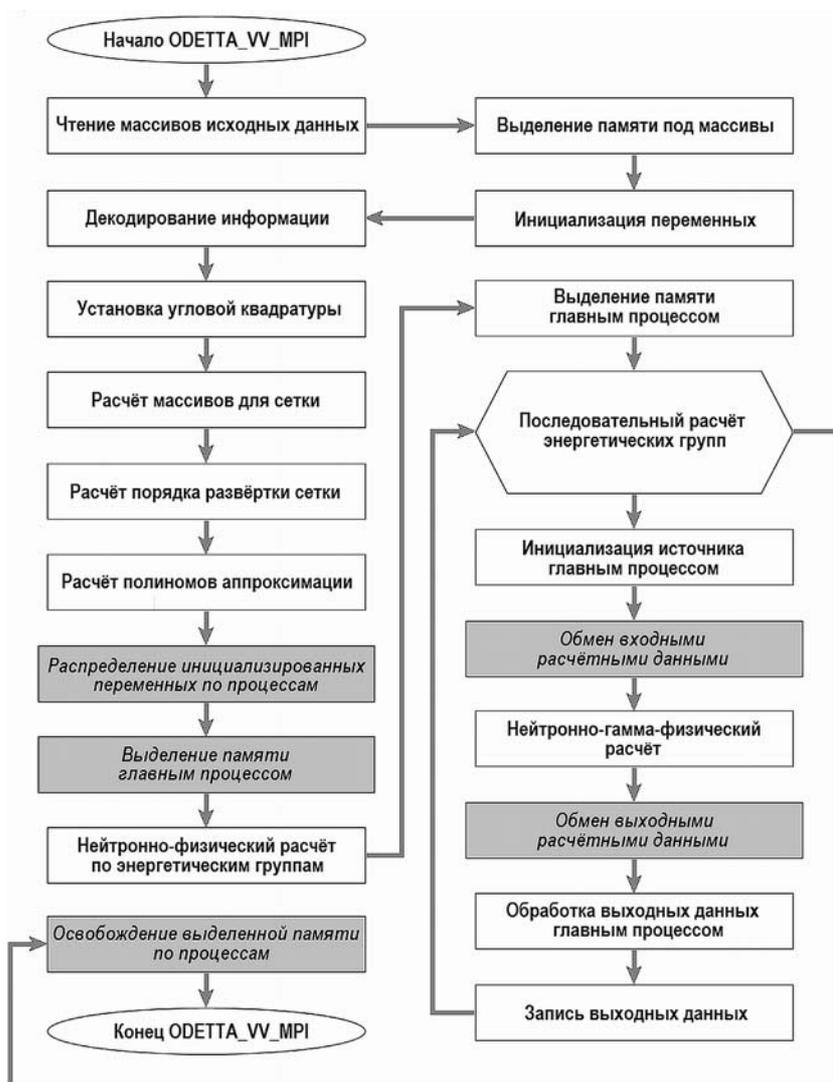


Рис. 4. Схема программы ODETTA с использованием MPI

Сборка после нахождения в работающем реакторе извлекалась и погружалась в стальной цилиндрический контейнер (обечайку), который затем заполнялся свинцом. Общие элементы моделирования расчёта приведены в табл. 1.

Таблица 1

Общая схема расчёта с кодом ODETTA

Этап	Объект моделирования	Решаемая задача	Входные данные	Выходные данные
Расчёт защиты	Топливная сборка в стальном контейнере (бочке), заполненном свинцом	Неоднородная задача с заданным источником	299-групповой источник нейтронов, 127-групповой фотонный источник, геометрические и физические параметры	Полный поток нейтронов, полный поток гамма-квантов, время расчёта
Сравнение результатов	ODETTA с MPI и без MPI			

Для упрощения расчёта модели полагалось, что оболочка цилиндрического кон-

тейнера (обечайки) изготовлена из стали НТ-9. При моделировании задачи использовалось SnPrm-приближение с параметрами $n = 12$ и $m = 3$, что означает 288 направлений всего для единичной сферы или 36 направлений на один октант (квадратура Чебышева-Лежандра CL12). Количество энергетических групп для нейтронов – 299, а для гамма-квантов – 127. Количество тетраэдров в сетке разбиения – 132883.

Программная реализация расчётного кода ODETTA выполнялась в высокопроизводительном вычислительном центре НИЯУ МИФИ, кластеры которого функционируют на базе ОС Linux. Управление удалённым компьютером через командную строку по протоколу SSH осуществлялось посредством программы Putty [19]. При расчёте тестовой задачи РБ с помощью программы ODETTA использовались кластеры «Basov» и «Cherenkov»[4] с различными числами узлов. Результаты приведены в табл. 2, 3.

Таблица 2

Реализация программы ODETTA на кластере «Basov»

Количество процессов	1	2	3	4
Время, ч	10.036	8.922	10.824	9.372
Ускорение	–	1.125	0.927	1.071
Эффективность	–	0.562	0.309	0.268

Таблица 3

Реализация программы ODETTA на кластере «Cherenkov»

Количество процессов	1	2	3	4
Время, ч	13.435	8.809	12.103	9.579
Ускорение	–	1.525	1.110	1.403
Эффективность	–	0.763	0.370	0.351

Ускорение $S_p = t_1/t_p$ и эффективность $E_p = S_p/p$ параллельных алгоритмов (t_1 – время выполнения алгоритма на одном процессе, t_p – время выполнения алгоритма для системы из p процессов) определяются в зависимости от числа процессов.

Расчёт тестовой задачи радиационной безопасности проводился на вычислительных узлах в количестве от одного до четырёх, причём использование одного узла было эквивалентно моделированию задачи без использования MPI, количество процессов OpenMP для каждого процесса было равно 32, т.е. максимальным для одного узла.

Вычисление на восьми узлах, при котором ожидалась максимальная эффективность решения задачи по времени, не состоялось из-за чрезмерной загруженности кластера. Поскольку на кластерах используются разные системы процессоров, то можно отметить, что время выполнения задач отличается в пользу кластера «Basov», а время выполнения программы на четырёх узлах больше, чем на двух из-за особенностей передачи данных между узлами кластера. Алгоритм программы ODETTA с MPI требует обмена больших объёмов данных между узлами кластера, поэтому передача данных вносит дополнительную задержку при выполнении расчёта. Однако ожидалось получить дополнительное ускорение на восьми узлах кластера за счёт процессоров, которые своим быстродействием минимизировали бы издержки обмена данными между узлами.

Сравнительный анализ по точности вычисления полных потоков частиц представлен в табл. 4, 5, где за эталонный результат был принят расчёт на одном узле, т.е. без применения MPI. Там же приведены оценки максимальной относительной погрешности δ_{\max} и среднеквадратичного отклонения σ . Для вычисления погрешностей использовались соотношения

$$\delta_{\max} = \max(\delta_1, \delta_2, \dots, \delta_n), \quad \delta_i = 1 - |x_i/x_i^*| = |\Delta x_i/x_i^*|,$$

где n – количество точек решения; x_i – расчётное решение; x_i^* – эталонное решение; $\Delta x_i = x_i - x_i^*$. Среднеквадратичное отклонение от эталонного решения

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n \Delta x_i^2}.$$

Таблица 4

Сравнение по результатам расчётов полных потоков частиц на кластере «Basov»

		Количество процессов	2	3	4
Нейтроны	$\delta_{\max}, \%$		$1.26 \cdot 10^{-3}$	$9.31 \cdot 10^{-4}$	$1.55 \cdot 10^{-3}$
	σ		$4.52 \cdot 10^5$	$1.48 \cdot 10^5$	$8.21 \cdot 10^5$
Гамма-кванты	$\delta_{\max}, \%$		$1.49 \cdot 10^{-3}$	$1.07 \cdot 10^{-3}$	$1.45 \cdot 10^{-3}$
	σ		$2.39 \cdot 10^4$	$7.57 \cdot 10^3$	$3.92 \cdot 10^4$

Таблица 5

Сравнение по результатам расчётов полных потоков частиц на кластере «Cherenkov»

		Количество процессов	2	3	4
Нейтроны	$\delta_{\max}, \%$		$2.03 \cdot 10^{-3}$	$1.48 \cdot 10^{-3}$	$1.49 \cdot 10^{-3}$
	σ		$1.82 \cdot 10^6$	$1.56 \cdot 10^6$	$1.99 \cdot 10^6$
Гамма-кванты	$\delta_{\max}, \%$		$2.04 \cdot 10^{-3}$	$1.47 \cdot 10^{-3}$	$1.50 \cdot 10^{-3}$
	σ		$1.10 \cdot 10^5$	$7.84 \cdot 10^4$	$1.09 \cdot 10^5$

Относительная погрешность при расчётах на двух и четырёх узлах не превышает 0.00155% при использовании кластера «Basov» и 0.00204% на кластере «Cherenkov», что говорит о незначительном отклонении от эталонного результата.

Таблица 6

Реализация программы ODETTA на одном узле при двух процессах MPI

Кластер	«Basov»	«Cherenkov»
Время, ч	6.951	7.029
Ускорение	1.444	1.911
Эффективность	0.722	0.956

Таблица 7

Характеристики вычислений программы ODETTA на одном узле при двух процессах MPI

		Кластер	«Basov»	«Cherenkov»
Нейтроны	$\delta_{\max}, \%$		$1.34 \cdot 10^{-3}$	$1.80 \cdot 10^{-3}$
	σ		$3.16 \cdot 10^5$	$1.75 \cdot 10^6$
Гамма-кванты	$\delta_{\max}, \%$		$1.25 \cdot 10^{-3}$	$1.96 \cdot 10^{-3}$
	σ		$1.69 \cdot 10^4$	$9.65 \cdot 10^4$

Однако оптимальным вариантом получения результатов вычислений программы ODETTA на кластерах является использование одного узла при установке двух процессов MPI и 16-ти процессов OpenMP для каждого процесса. Возможно, что это связано с особенностями устройства узлов кластера, которые позволяют алго-

ритмам распараллеливания программы получать максимальное ускорение по скорости выполнения расчёта. Преимущества этого варианта вычислений показаны в табл. 6, 7.

Относительная погрешность при расчётах на одном узле с использованием двух процессов MPI не превышает 0.00134% при использовании кластера «Basov» и 0.00196% на кластере «Cherenkov», что говорит о высокой достоверности полученных результатов.

ЗАКЛЮЧЕНИЕ

С помощью технологии MPI достигнуто ускорение расчётного кода ODETTA для решения задачи переноса нейтронов и гамма-квантов в многогрупповом SnPm-приближении методом конечных элементов на неструктурированных тетраэдральных сетках. Результаты модификации исходного кода были протестированы на примере задачи РБ.

Модификации программы тестировались с помощью задачи РБ [20]. Для проверки точности полученных результатов было проведено сравнение с результатами последовательной версии программы ODETTA, которые были приняты за эталон. Отклонение от эталона составило примерно 0.002%, что говорит о корректной реализации алгоритма MPI-распараллеливания.

Анализируя проведённые запуски вычислений с использованием разного числа узлов MPI и 32-х максимально возможных процессов OpenMP, можно сделать вывод, что самым эффективным оказалось распределение по двум узлам кластера «Cherenkov», где ускорение составило примерно 1.52. При использовании четырёх вычислительных узлов расчёт выполнялся медленнее, чем на двух, что связано с особенностями алгоритма и архитектуры кластера, а именно, с применением на «Basov» и «Cherenkov» технологии Hyper-threading [4], позволяющей использовать два вычислительных потока на одном физическом ядре процессора. Увеличение производительности изменяется от приложения к приложению. Однако скорость выполнения некоторых программ может уменьшаться, как и произошло в нашем случае. Одна из возможных причин связана с «системой повторения» процессоров Xeon, занимающей необходимые вычислительные ресурсы, что приводит к простоям других потоков.

В результате самым оптимальным вариантом работы на кластере программы ODETTA стало использование одного узла с двумя процессами MPI и 16-ти OpenMP. Максимальное ускорение, полученное на кластере «Cherenkov», составило 1.911 (почти двукратное). Таким образом, максимальный выигрыш по времени составил около 48 %. В свою очередь минимальный выигрыш по времени составил примерно 30% по сравнению даже с лучшим однопроцессорным расчётом.

В дополнение следует отметить, что алгоритм программы ODETTA с MPI требует обмена значительных объёмов данных между узлами кластера, поэтому передача данных внесла сильную задержку при выполнении вычислений. Использование нескольких процессоров отрицательно влияет на скорость из-за дополнительного межпроцессорного взаимодействия, однако время расчёта по сравнению с последовательной версией остаётся существенно меньшим.

Литература

1. Белоусов В.И., Березнев В.П., Селезнёв Е.Ф. Расчётный код ODETTA для решения задач переноса нейтронов и гамма-квантов в многогрупповом SnPm-приближении методом конечных элементов на неструктурированных тетраэдральных сетках, включая работу с сеточными данными. Версия 2.1 (ODETTA). / Аттестационный паспорт программы для ЭВМ, рег. № 497 от 19 декабря 2019 г. – М.: ФБУ «НТЦ ЯРБ», 2019. – 6 с.
2. Князева М.А., Молчанова Л.А., Тарасов Г.В. Параллельное программирование. – Владивосток: Дальневосточный университет, 2006. – 61 с.
3. Костромин В.А. Самоучитель Linux для пользователя. – СПб: БХВ-Петербург,

2012. – 672 с.

4. *Савченко А.В., Аникеев А.А., Окунев Д.Ю.* Высокопроизводительный вычислительный центр НИЯУ МИФИ. Руководство пользователя. – М.: НИЯУ МИФИ, 2020. – 24 с. Электронный ресурс: <https://ut.mephi.ru/pdf/projects/hpc/userguide.pdf> (дата доступа 10.02.2020).

5. *McGhee J.M., Wareing T.A., Barnett D.A.* Attila User's Manual. – Transpire Inc., 15 January, 2007. – 1077 p.

6. *Сычугова Е.П., Селезнёв Е.Ф.* Метод конечных элементов для решения уравнения переноса на неструктурированных тетраэдральных сетках. / Препринт № IBRAE-2014-03, 2014. – М.: ИБРАЭ РАН, 2014. – 21 с.

7. SALOME – The Open Source Integration Platform for Numerical Simulation. Электронный ресурс: <https://www.salome-platform.org> (дата доступа 10.02.2020).

8. *Мантуров Г.Н.* Методическое константное и программное обеспечение нейтронно-физических расчётов быстрых реакторов и оценки погрешностей расчётных предсказаний. / Дисс. на соиск. уч. степ. д-ра техн. наук. – Обнинск: ГНЦ РФ-ФЭИ, 2017. – 202 с.

9. About VisIt. Электронный ресурс: <https://wci.llnl.gov/simulation/computer-codes/visit> (дата доступа 10.02.2020).

10. OpenMP Application Program Interface. Электронный ресурс: <http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf> (дата доступа 10.02.2020).

11. *Немнюгин С.А.* Средства программирования для многопроцессорных вычислительных систем. Intel Multicore Curriculum Initiative. – СПб.: Санкт-Петербургский государственный университет, 2007. – 88 с.

12. *Павловская Т.А.* C/C++. Программирование на языке высокого уровня. – СПб.: Питер, 2003. – 461 с.

13. *Shterenlikht Anton.* Parallel Programming with Fortran 2008 and 2018 Coarrays. – Mech Eng Dept, The University of Bristol, Bristol BS8 1TR, UK, 2018. – 27 p.

14. *Рыжиков Ю.И.* Программирование на Фортране Powerstation для инженеров. Практическое руководство. – СПб: Корона-Принт, 2000. 161 с.

15. *Бартедьев О.В.* Современный Фортран. 3-е изд., доп. и перераб. – М.: Диалог МИФИ, 2000. – 449 с.

16. *Chapman Stephen J.* Fortran for Scientists and Engineers. Fourth Edition. – BAE Systems Australia, 2018. – 1049 p.

17. Сборка программ с помощью GNU make. Электронный ресурс: <http://coderway.ru/cpp/make> (дата доступа 10.02.2020).

18. *Nuclear Science Committee.* Benchmark for Neutronic Analysis of Sodium-cooled Fast Reactor Cores with Various Fuel Types and Core Sizes. NEA/NSC/R(2015)9, 25-Feb-2016. Электронный ресурс: <https://www.oecd-nea.org/science/docs/2015/nsc-r2015-9.pdf> (дата доступа 10.02.2020).

19. Документация Putty. Электронный ресурс: <https://putty.org.ru/docs.html> (дата доступа 10.02.2020).

20. *Bereznev V.P., Belousov V.I., Grushin N.A. et al.* New neutronic calculation codes based on discrete ordinates method using methods of finite differences and finite elements / Proc. of the Int. Conf. on Fast Reactors and Related Fuel Cycles (FR17). Paper CN-195. – Ekaterinburg: ROSATOM, 2017. – 10 p. Электронный ресурс: https://www.researchgate.net/publication/339912318_IAEA-CN245-195_New_neutronic_calculation_codes_based_on_discrete_ordinates_method_using_methods_of_finite_differences_and_finite_elements (дата доступа 10.02.2020).

Поступила в редакцию 17.02.2020 г.

Авторы

Шошина Анастасия Васильевна, студентка

E-mail: iamnastish@gmail.com

Белюсов Виктор Иванович, начальник лаборатории, к.ф.-м.н.

E-mail: mirror08@yandex.ru

SPEEDING-UP THE ODETTA CALCULATION CODE FOR SOLVING PARTICLE TRANSPORT PROBLEMSShoshina A.V.*, Belousov V.I.*** National Research Nuclear University MEPHI
31 Kashirskoe Hwy, 115409 Moscow, Russia** NRC «Kurchatov Institute»
1 Academician Kurchatov Sq., 123182 Moscow, Russia

ABSTRACT

In the mathematical modeling of fast breeding reactors, it becomes necessary to carry out high-precision calculations of protection problems on unstructured grids. In this paper, we consider and analyze a parallel version of the ODETTA calculation code using the MPI (Message Passing Interface) library technology. This code is intended for numerical modeling of neutronic processes in shielding compositions of fast breeding reactors with a liquid metal coolant in normal operation modes, and can also be useful in calculating the radiation conditions for the structural elements and equipment of nuclear energy facilities that are sources and/or exposed to ionizing radiation, to justify their safety. In addition, a comparison is made between the developed program and the previous version in terms of their efficiencies. The development of the ODETTA software algorithmic part using MPI is described. The features and specifics of this program parallelizing are presented. The modification of the calculation code is given. The corresponding software algorithms are considered. The structure of ODETTA software modules using MPI is briefly described. The results of operation of ODETTA serial and parallel versions in OS Linux using MPI on the HPC cluster of National Research Nuclear University MEPHI are presented. Finally, a comparative analysis of two software implementations is made in terms of speed and accuracy of results when two different clusters with different numbers of nodes on them are used. The features of performing calculations on a cluster are noted.

Key words: parallel programming, MPI, ODETTA software, finite element method, radiation safety, HPC cluster.

REFERENCES

1. Belousov V.I., Bereznev V.P., Seleznev E.F. ODETTA computational code for solving problems of neutron and gamma-ray transfer in the multi-group SnPm approximation by the finite element method on unstructured tetrahedral grids, including work with grid data. Ver. 2.1. (ODETTA). *Attestation Certificate of the Program for Electronic Computing Machines, Reg. No. 497, Dec. 19, 2019*. Moscow. FBU «NTC YaRB» Publ., 2019, 6 p. (in Russian).
2. Knyazeva M.A., Molchanova L.A., Tarasov G.V. *Parallel Programming*. Vladivostok. Dal'nevostochny Universitet Publ., 2006, 61 p. (in Russian).
3. Kostromin V.A. *Linux Tutorial for User*. Saint Petersburg. BHV-Petersburg Publ., 2012, 672 p. (in Russian).
4. Savchenko A.V., Anikeev A.A., Okunev D.Yu. *High-performance computing center NRNU MEPHI. User's manual*. Moscow. NRNU MEPHI Publ., 2020, 24 p. Available at: <https://ut.mephi.ru/pdf/projects/hpc/userguide.pdf> (accessed Feb 10, 2020) (in Russian).
5. McGhee J.M., Wareing T.A., Barnett D.A. *Attila User's Manual*. Transpire Inc., Jan 15, 2007, 1077 p.
6. Sychugova E.P., Seleznev E.F. *The Finite Element Method for Solving the Transport Equation on Unstructured Tetrahedral Meshes*. Preprint No. IBRAE-2014-03, 2014.

- Moscow. IBRAERAN Publ., 2014, 21 p. (in Russian).
7. SALOME – The Open Source Integration Platform for Numerical Simulation. Available at: <https://www.salome-platform.org> (accessed Feb 10, 2020).
 8. Manturov G.N. *Methodical Constant and Software for Neutron-Physical Calculations of Fast Reactors and Estimation of Errors of Computational Predictions*. Dr. tech. sci. diss. Obninsk. SSC RF-IPPE Publ., 2017, 202 p. (in Russian).
 9. About VisIt. Available at: <https://wci.llnl.gov/simulation/computer-codes/visit> (accessed Feb 10, 2020).
 10. OpenMP Application Program Interface. Available at: <http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf> (accessed Feb 10, 2020).
 11. Nemnyugin S.A. *Programming Tools for Multiprocessor Computing Systems*. Intel Multicore Curriculum Initiative. Saint Petersburg. Saint Petersburg State University Pubp., 2007, 88 p. (in Russian).
 12. Pavlovskaya T.A. *C/C++. High-Level Language Programming*. Saint Petersburg. Peter Publ., 2003, 461 p. (in Russian).
 13. Shterenlikht Anton. *Parallel Programming with Fortran 2008 and 2018 Coarrays*. Mech Eng Dept, The University of Bristol, Bristol BS8 1TR, UK, 2018, 27 p.
 14. Ryzhikov Yu.I. *Fortran Powerstation Programming for Engineers. A Practical Guide*. Saint Petersburg. Korona-Print Publ., 2000, 161 p. (in Russian).
 15. Barteniev O.V. *Modern Fortran*. 3-rd ed. Moscow. Dialog MIFI Publ., 2000, 449 p. (in Russian).
 16. Chapman Stephen J. *Fortran for Scientists and Engineers*. 4-th ed. BAE Systems Australia, 2018. 1049 p.
 17. *Building Programs with GNU Make*. Available at: <http://coderway.ru/cpp/make> (accessed Feb 10, 2020) (in Russian).
 18. Nuclear Science Committee. *Benchmark for Neutronic Analysis of Sodium-cooled Fast Reactor Cores with Various Fuel Types and Core Sizes*. NEA/NSC/R(2015)9, 25-Feb-2016. Available at: <https://www.oecd-neo.org/science/docs/2015/nsc-r2015-9.pdf> (accessed Feb 10, 2020).
 19. Putty Documentation. Available at: <https://putty.org.ru/docs.html> (accessed Feb 10, 2020) (in Russian).
 20. Bereznev V.P., Belousov V.I., Grushin N.A. et al. New neutronic calculation codes based on discrete ordinates method using methods of finite differences and finite elements. *Proc. of the Int. Conf. on Fast Reactors and Related Fuel Cycles (FR17)*. Paper CN-195. Ekaterinburg. ROSATOM Publ., 2017, 10 p. Available at: https://www.researchgate.net/publication/339912318_IAEA-CN245-195_New_neutronic_calculation_codes_based_on_discrete_ordinates_method_using_methods_of_finite_differences_and_finite_elements (accessed Feb 10, 2020).

Authors

Shoshina Anastasiya Vasilievna, Student

E-mail: iamnastish@gmail.com

Belousov Victor Ivanovich, Head of Laboratory, Cand. Sci. (Phys.-Math.)

E-mail: mirror08@yandex.ru