10. Menikoff R. Constitutive model for polymethyl methacrylate at high pressure // J. of App. Phys. 2004. Vol. 96, N 12. P. 7696–7704.

11. Мержиевский Л. А., Реснянский А. Д. Численное моделирование ударно-волновых процессов в металлах // ФГВ. 1984. Т. 20, № 5. С. 114–122.

12. Мержиевский Л.А. Моделирование динамического сжатия поликристаллического Al_2O_3 // ФГВ. 1998. Т. 34, № 6. С. 85–94.

13. Годунов С. К. Элементы механики сплошной среды. М.: Наука, 1978.

14. Мержиевский Л. А., Шамонин С. А. Построение зависимости времени релаксации касательных напряжений от параметров состояния среды // ПМТФ. 1980. № 5. С. 170–179.

ИСПОЛЬЗОВАНИЕ СТАНДАРТОВ OpenMP И MPI ПРИ РАСПАРАЛЛЕЛИВАНИИ РЕШЕНИЯ УРАВНЕНИЯ ПЕРЕНОСА МЕТОДОМ МОНТЕ-КАРЛО

А. В. Миронов

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, г. Саров

Введение

В 2007 году в математическом отделении ИТМФ была создана методика С-007 [1], предназначенная для решения методом Монте-Карло системы спектральных нестационарных линейных уравнений переноса нейтронов, гамма-квантов, электронов и позитронов в трехмерной неподвижной геометрии. Параллельный вариант программ методики был ориентирован на многопроцессорные вычислительные комплексы с распределенной памятью. Поэтому использовалась библиотека межпроцессорных обменов MPI [2].

Существуют задачи переноса излучения, геометрические данные которых занимают слишком большой объем памяти. При использовании стандарта MPI для каждого ядра процессора создается своя копия рабочих данных, что увеличивает риск переполнения оперативной памяти узла вычислительной системы. При этом большой объем данных, включающий в себя информацию о геометрии задачи и массивы констант, описывающих свойства изотопов, является одинаковым для каждого ядра. С точки зрения экономии оперативной памяти такое избыточное копирование нецелесообразно.

Узлы современной вычислительной системы, имеющие многоядерные процессоры и большой объем оперативной памяти, а также применение стандарта OpenMP [3] для создания параллельных программ на многопроцессорных системах с общей памятью позволяют решить вышеизложенную проблему нехватки ресурсов. Объявление не изменяющихся в процессе счета и одинаковых для каждого ядра данных «общими» (public) приводит к созданию единственной копии этих данных для каждого узла (или для каждого многоядерного процессора) многопроцессорной вычислительной системы. Применение стандартов MPI для обмена сообщениями между узлами (или между многоядерными процессорами) вычислительной системы и OpenMP на узле (процессоре) дает возможность более эффективно использовать оперативную память и решать ресурсоемкие задачи.

Вышеописанное решение было применено к процедурам методики С-007 путем изменения их кода с использованием программного интерфейса OpenMP.

В работе представлена общая схема изменений, внесенных в код процедур методики С-007, необходимых для реализации комбинированного подхода к распараллеливанию решения систем уравнений переноса. Исследуется влияние некоторых конструкций кода на скорость работы программ методики. Особое внимание уделено вопросам использования модульных переменных, опи-

санных с помощью OpenMP-директивы threadprivate, и их влияния на быстродействие процедур методики. Также в работе приводятся результаты сравнительных расчетов некоторых задач с помощью новой и старой методик.

Основные изменения программы С-007. Общая схема процедур методики С-007

Для решения методом Монте-Карло системы уравнений переноса требуется провести моделирование процесса прохождения ансамбля частиц различного типа (нейтроны, гамма-кванты, электроны, позитроны) через среду, определяемую условиями задачи. В процедурах методики С-007 моделирование производится пачками из TRJ_IN_PAC траекторий частиц, являющихся основными счетными единицами, и по результатам моделирования которых, производится оценка рассчитываемых функционалов.

Процедура, которая это выполняет, называется SIMPAC. Она осуществляет в цикле моделирование траекторий частиц пачки. Количество моделируемых пачек траекторий равно NUM_PAC.

Каждый MPI процесс считает свои пачки траекторий. После моделирования каждой пачки производится проверка условий остановки работы и сохранение промежуточных результатов. При завершении работы из промежуточных результатов, полученных с каждого MPI процесса, формируются окончательные результаты расчета.

Изменения процедуры SIMPAC

Была применена условная компиляция, которая позволила отделить части кода, необходимые для случая применения стандарта OpenMP. Это дало возможность получать из одного и того же кода процедуры для различных методик.

Каждый MPI процесс выполняет процедуру SIMPAC. Программный интерфейс OpenMP был применен к распараллеливанию внутреннего цикла этой процедуры, последовательно моделирующего независимые траектории частиц. Весь цикл заключен в блоке !$omp parallel. Для параллельного выполнения этого блока создаются OMP_NUM_THREADS (переменная окружения) нитей OpenMP. После описания того, какие переменные должны быть «общими», а какие «частными», следует инициализация переменных, выделение памяти и перенаправление указателей для всех нитей. Затем следует сам цикл, заключенный в блоке !$omp do. Распределение итераций между нитями в этом блоке было установлено по типу runtime, позволяющему менять распределение в зависимости от предпочтения пользователя через переменную окружения OMP_SCHEDULE.

Для того чтобы некоторые операторы выполнялись в один и тот же момент времени лишь одной нитью, использовались критические блоки. Эти блоки были применены к операторам инкрементации общих для всех нитей счетчиков и к операторам суммирования массивов промежуточных результатов.

Кроме директив, также применялись библиотечные функции OpenMP. Они использовались для установки количества нитей, создаваемых в параллельном блоке, для получения идентификатора нити и для временных засечек.

Каждый MPI процесс имеет свой идентификатор $NPROC_{MPI}$, который имеет остаток от деления на количество OpenMP нитей в одном процессе равный единице. Каждая нить получает свой уникальный номер $NPROC = NPROC_{MPI} + OMP\_TID$, где $NPROC_{MPI}$ – это идентификатор процесса, а OMP_TID – локальный номер нити. Таким образом, у каждого ядра вычислительной системы имеется свой идентификатор. По нему определяется стартовая четверка чисел для генератора псевдослучайных чисел, что обуславливает наличие неповторяющейся последовательности случайных чисел и независимость траекторий, моделируемых каждым ядром. Этим ограничивается применение программного интерфейса OpenMP.

При внесении изменений требовалось уделить особое внимание работе с объектами, имеющими в своем описании спецификатор pointer или allocatable. Такие объекты, если они были созда-

ны вне процедуры SIMPAC, было необходимо объявлять и инициализировать вручную для каждой нити. Специально для этого были созданы отдельные подпрограммы. Также были добавлены процедуры для создания копий массивов хранения промежуточных результатов .для каждой нити и сбора этих результатов в одном массиве по окончании счета пачки частиц. Каждая нить копит результаты счета в своем массиве, а по окончании счета пакета эти результаты суммируются, что приводит к отсутствию очередей на запись в одну область памяти.

Модульные переменные и их влияние на быстродействие

В программах методики С-007 передача параметров подпрограммам обычно осуществляется через модульные переменные. В параллельном блоке OpenMP для каждой нити эти переменные по умолчанию становятся общими, т. е. существует только одна копия переменной, которую могут менять все нити. Большинство таких переменных должно быть личными для каждой нити, так как в них хранятся параметры, уникальные для каждой траектории частицы.

В стандарте OpenMP для таких ситуаций предусмотрена директива !$omp threadprivate. Она определяет переменные, находящиеся в common блоке или в модуле, локальными для каждой нити. Также существует директивная клаузула (оператор) OpenMP copyin, позволяющая инициализировать переменные служебных нитей, описанные как threadptivate значениями переменных управляющей нити, имеющих одинаковые имена. Описанные инструменты были применены для создания новых процедур. К сожалению, быстродействие полученных программ оказалось неудовлетворительным. Даже последовательный вариант новой методики (1 MPI-процесс и 1 OpenMP-нить) проигрывал в скорости старой версии на одном ядре более чем в 3 раза! Анализ ситуации показал, что столь низкое быстродействие обуславливается использованием директивы threadprivate.

После этого была начата работа по переводу локальных для нитей модульных переменных в векторный вид. Предполагалось написать своеобразную интерпретацию механизма получения модульных переменных, являющихся локальными для каждой нити. Это было сделано путем векторизации этих переменных. Каждый объект заменялся массивом таких объектов с количеством элементов, равным количеству создаваемых нитей в параллельном блоке. Каждая нить обращалась к своему элементу массива. Были добавлены подпрограммы выделения памяти и инициализации этих массивов. В итоге быстродействие программ новой методики по сравнению с быстродействием программ методики С-007 было ниже на одном ядре в 1,25 раза. Кроме того, при увеличении количества используемых ядер новая методика проигрывала все больше. Так на 16-ти ядрах замедление составляло 16,5 раз! Причина такого замедления работы заключалась в том, что, введя векторные переменные, мы получили ситуацию, называемую в англоязычной литературе «false sharing» [4].

В итоге было принято решение отказаться от модульных переменных при передаче значений параметров в подпрограммы. Данные были сгруппированы в объекты новых пользовательских типов, которые объявляются в начале программы и процедуры SIMPAC, а затем передаются как явные параметры в процедуры и функции. В итоге программы новой методики работают в среднем на 10 % медленнее программ методики С-007.

Сравнительные расчеты

В этом разделе приведены результаты сравнительных расчетов некоторых типовых задач по программам методик С-007 и С007-RAZ (новая методика).

Тесты проводились на 8, 16, 32, 64 и 128 ядрах многопроцессорной вычислительной системы.

На одном узле вычислительной системы для программ методики С007-RAZ работали 2 MPI процесса с 8 нитями OpenMP на каждом. Переменная окружения OMP_SCHEDULE имела значение DYNAMIC.

Расчеты проводились в двух основных режимах: расчет эффективного коэффициента размножения нейтронов и перенос нейтронов и гамма-квантов.

Расчеты переноса нейтронов и гамма-квантов

Эти задачи с точки зрения моделирования траекторий отличаются от задач на расчет эффективного коэффициента размножения нейтронов намного более длинными и разветвленными траекториями частиц в пачке. Кроме того, длина траекторий может сильно изменяться от частицы к частице в пачке. Поэтому значение переменной окружения OMP_SCHEDULE уместно установить равным DYNAMIC. В этом случае каждая нить будет получать очередную порцию частиц, как только она освободится. Таким образом, возможна динамическая балансировка нагрузки на ядра.

Задача № 1. Вложенные друг в друга сферы.

Геометрия задачи представляет собой систему из вложенных друг в друга десяти сфер и отображена на рис. 1.

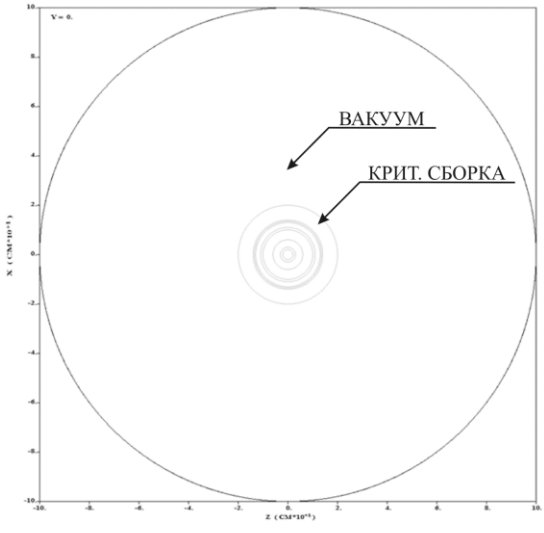

Рис. 1. Геометрия задачи № 1

Слои с первого по девятый заполнены различным веществом, как активным, так и не активным. В десятом, внешнем, слое находится вакуум. Источник нейтронов с энергией 14,1 Мэв находится в девятом слое.

На рис. 2 отображены результаты сравнительных расчетов этой задачи. В среднем программы методики C007-RAZ считают эту задачу медленнее программ методики C-007 на 9,5 %.

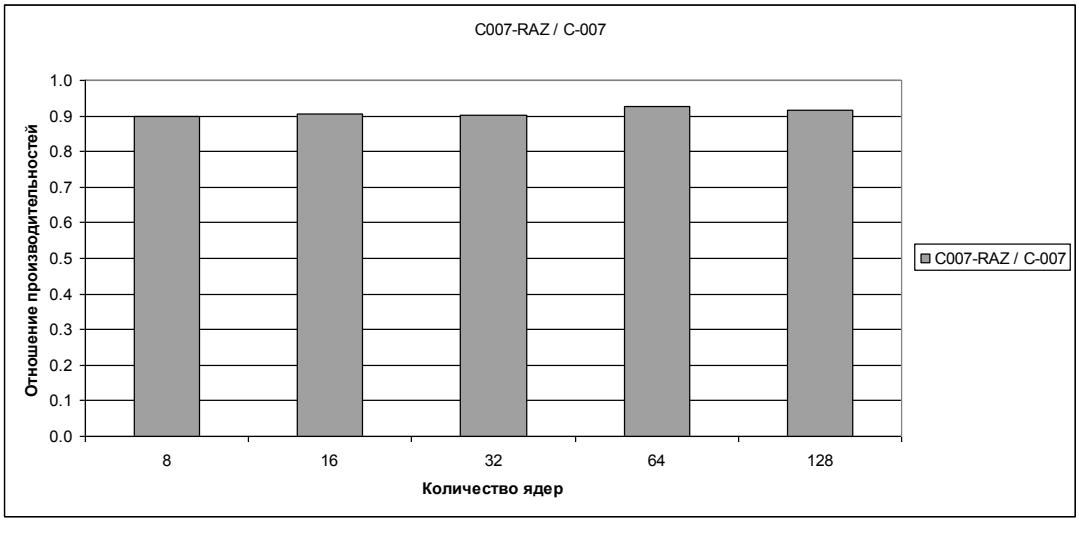

Рис. 2. Отношение производительностей методик C007-RAZ и C-007 для задачи № 1

Задача № 2. Бокс с активными материалами и детекторами.

Геометрия задачи представляет собой подвал с двумя комнатами, соединенными отверстием в стене, перекрытым заслонкой. В одной комнате находится активное вещество, в другой – 4 детектора, прикрепленных к стене. Некоторые сечения геометрии задачи представлены на рис. 3 и 4.

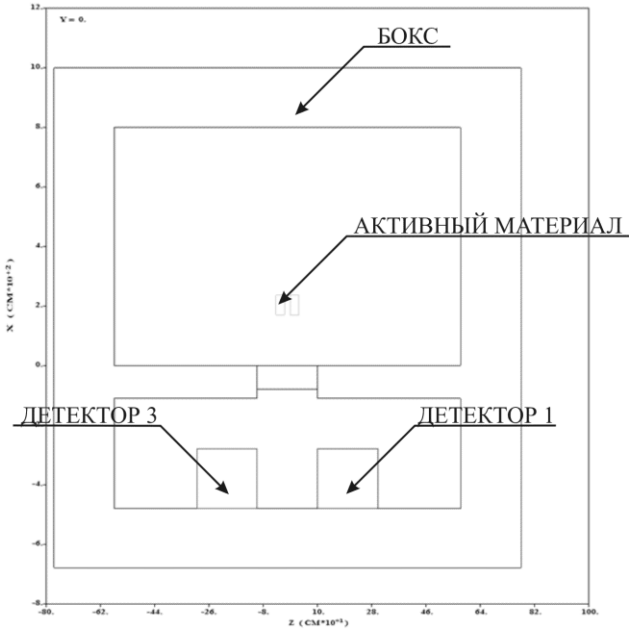

Рис. 3. Горизонтальное сечение геометрии задачи № 2 плоскостью Y = 0

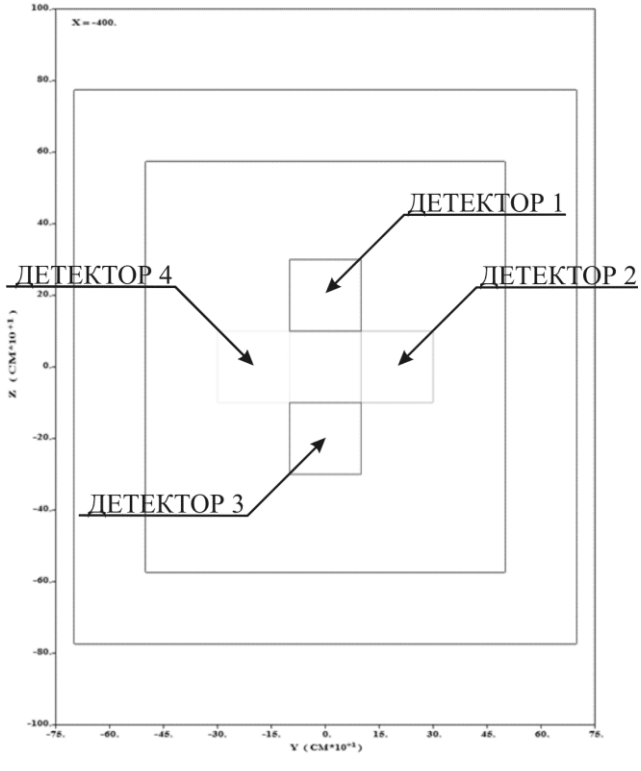

Рис. 4. Вертикальное сечение геометрии задачи № 2 плоскостью X = –400, схема расположения детекторов

В этой задаче моделировались нейтроны и гамма-кванты. Источник задавался в активном веществе. Энергетический спектр испускаемых им нейтронов задавался равным спектру деления U^{235}.

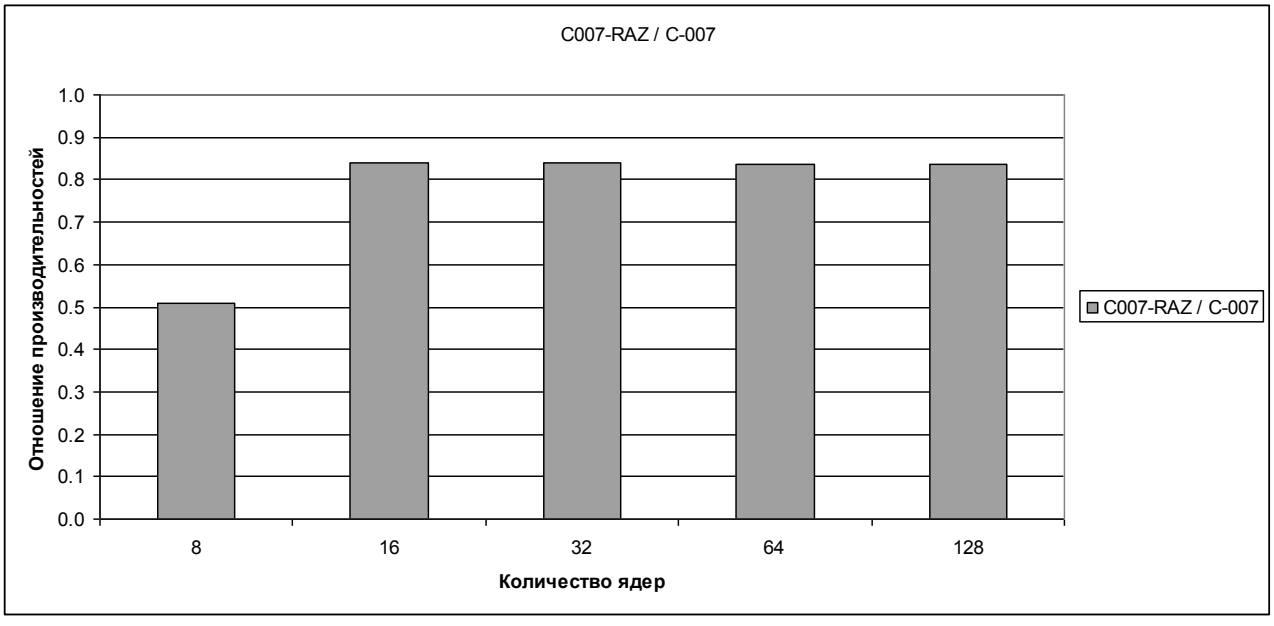

Рис. 5. Отношение производительностей методик C007-RAZ и C-007 для задачи № 2

Как видно из рис. 5 OpenMP + MPI проигрывает 20 % MPI. Профилирование работы программ методики C007-RAZ показало, что выполнение поиска расстояния до границы геометрического блока происходит медленнее, чем в базовой методике, что может быть вызвано недостаточно эффективной обработкой рекурсивных функций в стандарте OpenMP.

Расчеты эффективного коэффициента размножения нейтронов

Представленные в этом подразделе задачи являются реакторными. Геометрия таких задач отличается большими размерами и подробностью своего описания, что сказывается на трудоемкости подготовки и проведении расчетов. Большую часть времени при моделировании траекторий частиц здесь занимает расчет расстояния до геометрических границ.

Задача № 3. Бесконечная система ТВЭЛ.

Геометрия этой задачи довольно проста, ее поперечный срез представлен на рис. 6.

Бесконечность ТВЭЛа была реализована при помощи тактики отражения частиц на торцах цилиндрической геометрии. Источник нейтронов испускал высокоэнергетические частицы.

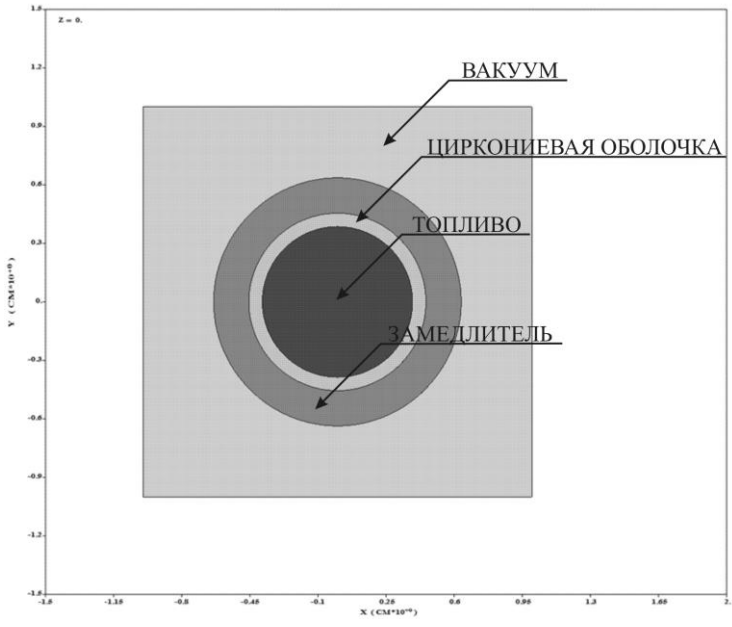

Рис. 6. Поперечный разрез ТВЭЛ

На рис. 7 представлены результаты сравнений программ методик на задаче ТВЭЛ. За некоторым исключением проигрыш в производительности не превышает 5 %.

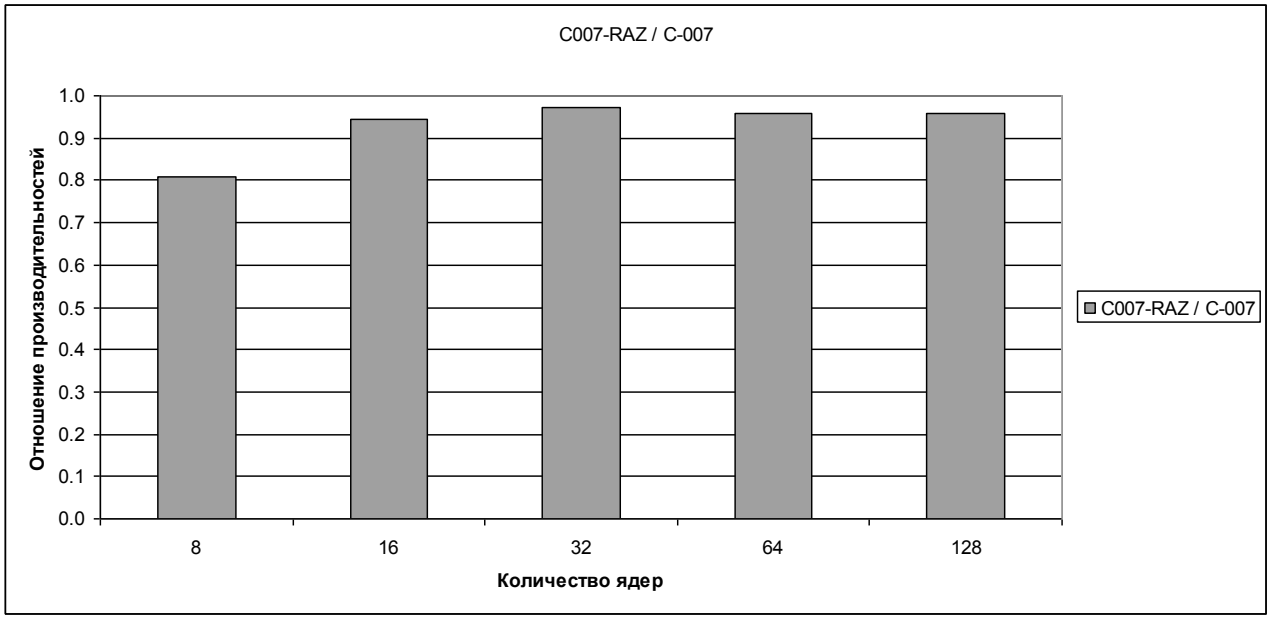

Рис. 7. Отношение производительностей методик C007-RAZ и C-007 для задачи № 3

Задача № 4. Бесконечная система ТВС.

Геометрия этой задачи сложнее предыдущей, поперечный срез представлен на рис. 8.

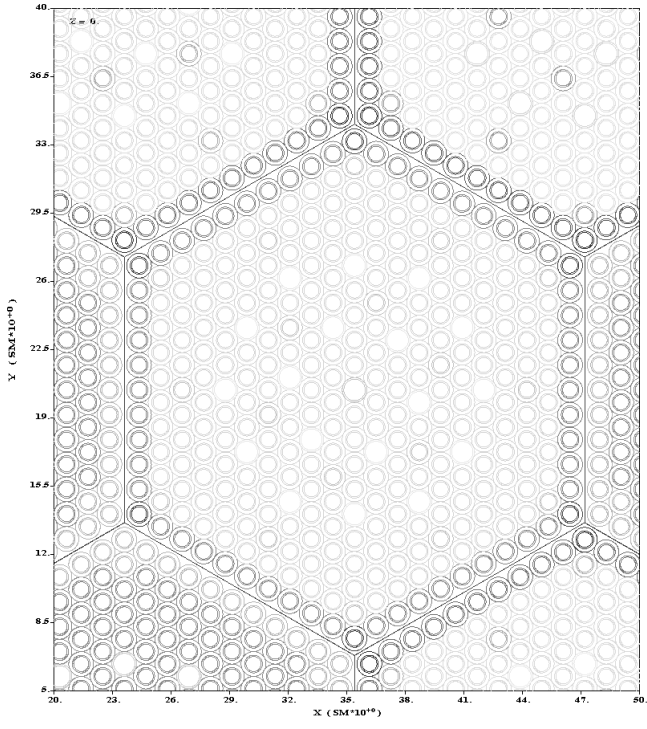

Рис. 8. Поперечный разрез бесконечной системы ТВС

Аналогично предыдущей задаче источник испускал высокоэнергетические 14,1 Мэв нейтроны. На границе внешнего геометрического блока задавалась тактика отражения

На рис. 9 представлены результаты расчетов. Отставание программ методики C007-RAZ не превышает 6 %.

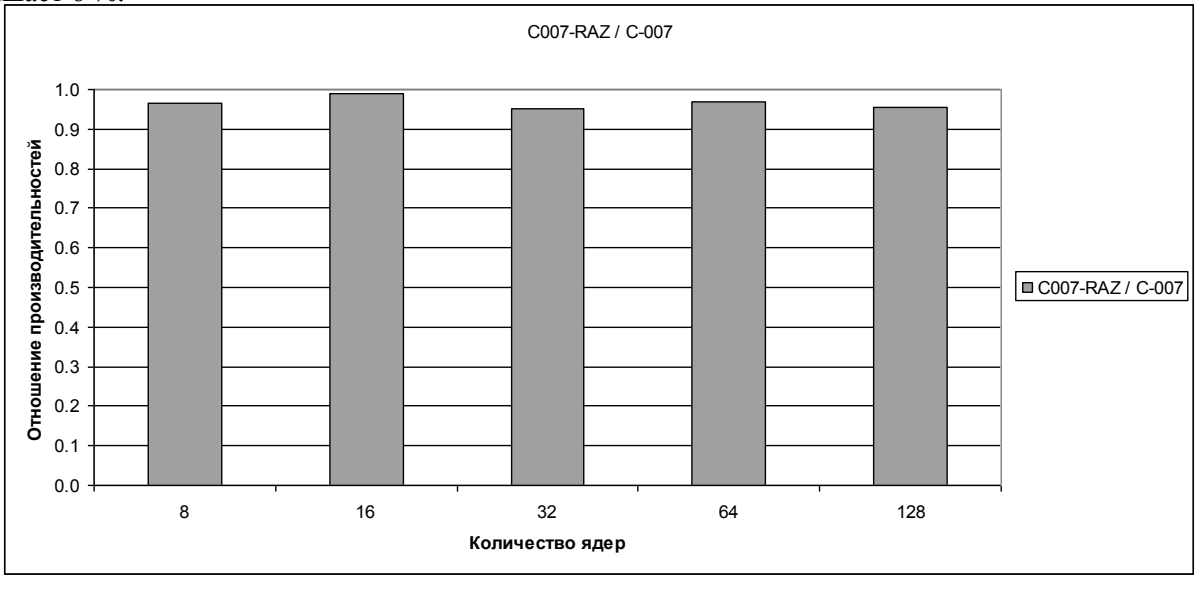

Рис. 9. Отношение производительностей методик C007-RAZ и C-007 для задачи № 4

Задача № 5. Активная зона реактора ВВЭР-1000.

Геометрия задачи представляет собой шестую часть активной зоны ВВЭР-1000, окруженную поверхностью, на которой задана тактика отражения. Поперечное сечение представлено на рис. 10. Частицы в источнике рождаются с энергией 14,1 Мэв.

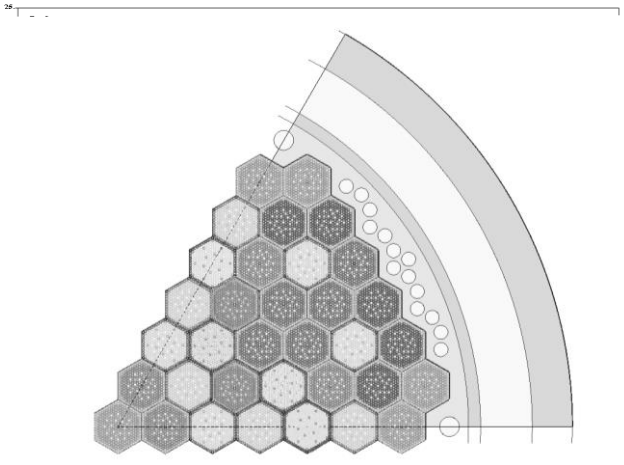

Рис. 10. Поперечный разрез шестой части ВВЭР-1000

На рис. 11 представлены результаты сравнительных расчетов. За некоторым исключением новые программы отстают не более чем на 6 %.

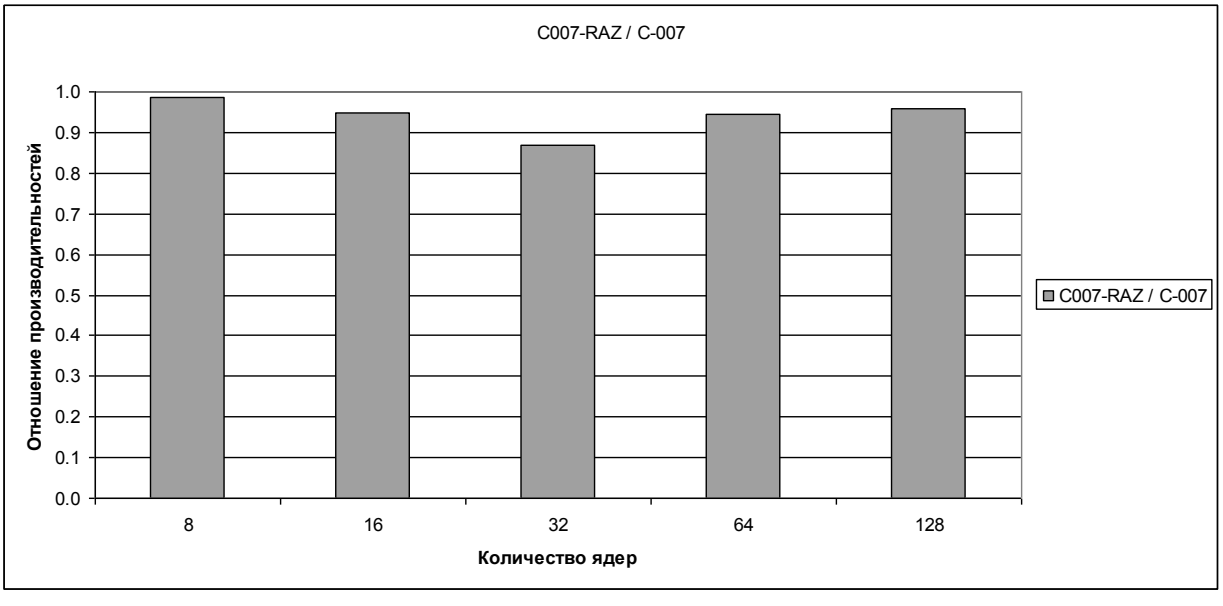

Рис. 11. Отношение производительностей методик C007-RAZ и C-007 для задачи № 11

Выводы из сравнительных расчетов

На основании представленных данных можно сделать вывод, что полученные программы уступают своим предшественницам в среднем не более 10 %. Некоторые задачи, при решении которых необходимо часто вычислять расстояния до границ трехмерных блоков, являются сравнительно тяжелыми для OpenMP, что дает проигрыш в производительности до 25 %.

Полученные данные об использовании ресурсов оперативной памяти, представленные в таблице, подтверждают более рациональное использование памяти. Выигрыш в использовании ресурсов памяти может достигать вплоть до 16 раз.

Использование оперативной памяти задачами

№ задачи	16OpenMPx1MPI, КБ	16MPI, КБ	Отношение
3	38119	594240	15,58
4	42717	671232	15,71
5	74200	1158400	15,61

Заключение

На базе программ методики С-007 созданы их модификации для методики С007-RAZ, в которой использовались одновременно два стандарта создания параллельных программ – MPI и OpenMP. В ходе проделанной работы был проведен анализ использования различных конструкций OpenMP для распараллеливания программы.

Полученные программы имеют быстродействие, сравнимое с быстродействием базовых программ (среднее отставание методики С007-RAZ от С-007 составляет 10 %). Основным преимуществом новых программ является более рациональное (до 16 раз) использование ресурсов памяти узла многопроцессорной вычислительной системы.

Литература

1. Житник А. К., Донской Е. Н. и др. Методика С-007 решения методом Монте-Карло связанных линейных уравнений переноса нейтронов, гамма-квантов, электронов и позитронов // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2011. Вып. 1. С. 17–24.

2. Snir M., Otto S., Huss-Lederman S. et al. MPI: The Complete Reference. Cambridge, Massachusetts, London, England: The MIT Press, 1996.

3. OpenMP website. [Electronic resource]. Mode of access: http://www.openmp.org.

4. Chapman B., Jost G., van der Pas R. Using OpenMP, portable shared memory parallel programming. Cambridge, Massachusetts, London, England: The MIT Press, 2008.

ПРОГРАММА МОДЕЛИРОВНИЯ ПРОТОЧНЫХ ИЗОБРАЖЕНИЙ ПИ: КРАТКОЕ ОПИСАНИЕ И ПРИМЕРЫ ПРИМЕНЕНИЯ

К. Л. Михайлюков, И. В. Храмов, А. В. Скобеев, С. В. Потапов, Н. В. Фролова, М. Д. Романова

Российский федеральный ядерный центр –
Всероссийский НИИ экспериментальной физики, г. Саров

Введение

Метод протонной радиографии с использованием магнитных линз, впервые появившийся в Лос-Аламосе в середине 90-х годов прошлого века и применяемый ВНИИЭФ с 2005 года на ускорителе У-70 в Протвино, зарекомендовал себя качественным инструментом по исследованию газо-