

# Моделирование нейтронно-физических процессов активной зоны реактора АЭС в реальном времени с применением распределенных вычислений

О.О. Казьмин<sup>1</sup>, И.А. Капацкая<sup>3</sup>, С.А. Карпов<sup>2</sup>

НИИ Механики МГУ<sup>1</sup>, ОАО «ВНИИАЭС»<sup>2</sup>, ОАО «ДжЭТ»<sup>3</sup>

В данной статье представлены результаты работ, направленных на адаптацию существующих алгоритмов моделирования нейтронно-физических процессов в активной зоне реактора АЭС для работы на суперкомпьютерных платформах в рамках создания ПТК «Виртуальная АЭС с ВВЭР». За основу модификации была взята модель активной зоны реактора ВВЭР-1000 полномасштабного тренажера Нововоронежской АЭС 2. Проанализирована производительность данной модели с целью выявления наиболее затратных ее частей. Исследованы отдельные компоненты модели и связи между ними для определения возможности распараллеливания. Разработана и реализована распределенная версия алгоритмов модели. Реализованные методы были апробированы, исследована их эффективность и предложены направления дальнейшей их оптимизации.

## 1. Введение

Задача обеспечения безопасности существующих и проектируемых АЭС становится еще более актуальной в связи с рядом чрезвычайных ситуаций, произошедших на атомных электростанциях в последние десятилетия. Принимая во внимание цену каждой ошибки при эксплуатации АЭС и отсутствие реальной альтернативы атомной энергетике, следует самое серьезное внимание уделить моделированию различных внештатных ситуаций и их вероятных последствий, в первую очередь - чрезвычайного характера.

В настоящее время во Всероссийском научно-исследовательском институте атомных электростанций (далее ВНИИАЭС) создается суперкомпьютерная полномасштабная модель атомной электростанции, именуемая "Виртуальной АЭС", с помощью которой можно серьезно продвинуться в решении таких задач, как

- 1) расчет штатных режимов эксплуатации, а также режимов с нарушением нормальных условий эксплуатации существующих и проектируемых АЭС;
- 2) качественная подготовка оперативного персонала АЭС с помощью симуляторов управления АЭС в режиме реального времени;
- 3) разработка оптимальных сценариев минимизации последствий внештатных ситуаций на существующих и проектируемых АЭС;
- 4) упреждающее моделирование в процессе принятия решений по выходу из опасных внештатных ситуаций.

Полномасштабная модель создается на основе прошедших апробацию и хорошо зарекомендовавших себя алгоритмов и их программных реализаций на ЭВМ традиционной последовательной архитектуры. Среди них немаловажную роль играет нейтронно-физический код ТРЕК, предназначенный для моделирования нестационарных процессов в активной зоне реактора - изменения пространственного распределения плотности потока нейтронов, энерговыделения, а также остаточного энерговыделения в активной зоне.

Повышение точности расчетов при условии моделирования в режиме реального времени, необходимость многовариантных расчетов а также упреждающего моделирования требуют высокопроизводительной реализации расчетных кодов. Для этого в настоящее время ведется их адаптация к современным суперкомпьютерным вычислительным установкам, особенностью которых является массовый параллелизм и распределенная по узлам оперативная память. Компактные варианты подобных систем могут в перспективе позволить оснастить высокоточными системами моделирования все заинтересованные организации, в первую очередь ситуационные и учебные центры.

Однако современным суперкомпьютерам присущи некоторые ограничения, которые делают распараллеливание данного программного комплекса достаточно сложной задачей. В настоящей статье рассматриваются следующие вопросы, связанные с переносом указанного кода на суперЭВМ:

- 1) производится анализ компонентов системы, занимающих наибольшее количество времени, для определения возможности их распараллеливания;
- 2) предлагается схема распараллеливания;
- 3) проводится анализ эффективности и выявляются механизмы, ограничивающие ее производительность;
- 4) предлагаются перспективные методы оптимизации расчетов с учетом целевой платформы, механизмов распараллеливания и особенностей модели.

## 2. Общее описание модели

Существуют различные методы моделирования нейтронно-физических процессов в активной зоне атомного реактора, программные реализации которых могут существенно отличаться друг от друга. При адаптации программной модели к суперкомпьютерным вычислительным установкам следует учитывать особенности как алгоритма, так и его программной реализации. По этой причине прежде, чем переходить к описанию схемы распараллеливания, необходимо представить использующиеся алгоритмы и особенности их программной реализации. В данной главе описывается методика нейтронно-физического расчета и особенности структуры ее реализации, которые оказывают существенное влияние на адаптацию расчетов к суперкомпьютерным установкам.

### 2.1. Методика нейтронно-физического расчета

В нейтронно-физическом расчете используется 3-х мерное, 2-х групповое диффузионное приближение с разделением переменных с 6 группами запаздывающих нейтронов. Амплитуда изменения мощности определяется в точечном приближении.

При расчете остаточного энерговыделения учитываются 11 групп осколков деления, распределенных по объему активной зоны (АЗ). Из продуктов деления рассматриваются только ксенон, самарий, йод и прометий, т.к. существенное влияние на временное поведение реактора оказывают только эти элементы, остальные продукты деления учтены в сечениях, рассчитанных для определенного момента времени. Зависимость средней времени жизни нейтрона от плотности воды и выгорания незначительна, поэтому при расчетах используется усредненное по реактору время жизни нейтрона.

Программа обеспечивает расчет объемного распределения энерговыделения в стационарных и нестационарных режимах работы реактора до 10 раз в секунду, в режиме реального времени.

Выгорание топлива не моделируется. Поведение реактора в различные моменты кампании моделируется через использование сечений, зависящих от выгорания топлива.

Исходная система уравнений, отвечающая диффузионной двухгрупповой модели, выглядит следующим образом:

$$\left\{ \begin{array}{l} \frac{1}{v^{(1)}} \frac{\partial \varphi^{(1)}(\vec{r}, t)}{\partial t} = \nabla D^{(1)} \nabla \varphi^{(1)}(\vec{r}, t) + (1 - \beta) \left[ v_f^{(1)} \Sigma_f^{(1)}(\vec{r}) \varphi^{(1)}(\vec{r}, t) + v_f^{(2)} \Sigma_f^{(2)}(\vec{r}) \varphi^{(2)}(\vec{r}, t) \right] - \\ \quad - \Sigma_a^{(1)}(\vec{r}) \varphi^{(1)}(\vec{r}, t) - \Sigma_a^{(1 \rightarrow 2)}(\vec{r}) \varphi^{(1)}(\vec{r}, t) + \sum_{m=1}^6 \lambda_m C_m(\vec{r}, t) + S^{(1)}(\vec{r}, t) \\ \frac{1}{v^{(2)}} \frac{\partial \varphi^{(2)}(\vec{r}, t)}{\partial t} = \nabla D^{(2)} \nabla \varphi^{(2)}(\vec{r}, t) - \Sigma_a^{(2)}(\vec{r}) \varphi^{(2)}(\vec{r}, t) + \Sigma_a^{(1 \rightarrow 2)}(\vec{r}) \varphi^{(1)}(\vec{r}, t) + S^{(2)}(\vec{r}, t) \\ \frac{\partial C_m(\vec{r}, t)}{\partial t} = \beta_m \left[ v_f^{(1)} \Sigma_f^{(1)}(\vec{r}) \varphi^{(1)}(\vec{r}, t) + v_f^{(2)} \Sigma_f^{(2)}(\vec{r}) \varphi^{(2)}(\vec{r}, t) \right] - \lambda_m C_m(\vec{r}, t), \quad m = 1..6 \end{array} \right. \quad (2.1.1)$$

где  $\varphi^{(g)}(\vec{r}, t)$  - функция потока нейтронов,  $g$  - номер энергетической группы

$D^{(g)}$  – коэффициент диффузии

$\beta$  - доля запаздывающих нейтронов

$\Sigma_{f,a,d}^{(g)}(\vec{r})$  – макросечение деления, поглощения, увода нейтронов

$\lambda_m, C_m$  - постоянная распада и концентрация ядер эмиттеров запаздывающих нейтронов

$S^{(g)}(\vec{r}, t)$  – источники нейтронов

Функция потока нейтронов представляется как  $\varphi^{(g)}(\vec{r}, t) = T(t)\Phi^{(g)}(\vec{r}, t)$ , где  $T(t)$  - амплитудная функция,  $\Phi^{(g)}(\vec{r}, t)$  – функция пространственного распределения, предполагается слабое ее изменение по отношению к  $T(t)$ . Для нахождения пространственной функции распределения потока нейтронов используется итерационный метод.

## 2.2. Структура программной модели

Нейтронно-физический расчетный код ТРЕК реализован в рамках базовой среды моделирования S3, которая допускает использование MPI-процессами всех вычислительных ядер суперкомпьютера. Запуск расчетного кода осуществляется в S3 в виде запуска с определенной частотой контрольных модулей программы, в которых уже вызываются расчетные модули. Все модули системы реализованы на языке Фортран. Описание всех модулей нейтронно-физического кода представлено в таблице 1.

Таблица 1. Структура нейтронно-физического кода ТРЕК.

Частота вызова	Контрольный модуль	Подпрограммы	Описание
10	cr4dy1	scrd01	Интерфейс с моделями других технологических систем АЭС
		scrd02 yscrstk yscsrcor	Нейтронно-физические константы для ТВС <sup>1</sup> Полиномы для ТВС Полиномы для СУЗ <sup>2</sup>
		scrd04	Реактивность. Амплитудная функция
		scrd05	Остаточное тепловыделение
10	cr4dy2	scrd03	Пространственная функция
10	cr1dy1	scrd06	Матрица для пространственной функции, коэффициентов-утечек
1	cr1dy2	scrd07	Отравление

Расчетные модули комплекса представляют собой относительно самостоятельные расчетные единицы. Как следствие, в большинстве своем они связаны друг с другом довольно небольшим объемом данных. По указанной причине было принято решение проводить модификацию кода программного комплекса на уровне модулей.

## 2.3. Результаты профилировки расчетного кода ТРЕК

В целях выявления наиболее затратных компонентов программного комплекса была проведена профилировка работы модулей. Анализ производился при моделировании трех минут работы реактора. Результаты представлены в таблице 2.

<sup>1</sup> Тепловыделяющая сборка

<sup>2</sup> Система управления и защиты

**Таблица 2.** Результаты профилировки расчетного кода ТРЕК.

Контрольный модуль	Подпрограммы	Затраченное время	
		Абсолютное, сек	Относительное, %
cr4dy1	scrd01	менее 0.01	менее 0.01%
	scrd02	0.35	0.47%
	yscrstk	20.51	27.47%
	yscrsor	23.21	31.10%
	scrd04	0.05	0.07%
	scrd05	0.13	0.17%
cr4dy2	scrd03	25.78	34.54%
cr1dy1	scrd06	4.24	5.68%
cr1dy2	scrd07	0.37	0.50%

Нетрудно отметить, что три модуля занимают в общей сложности более 90% времени всей системы. Причем, модули расчета полиномов для ТВС и СУЗ (yscrstk и yscrsor) являются подпрограммами модуля расчета нейтронно-физических констант (scrd02). На основе этих данных было принято решение ограничиться модификацией только указанных компонентов системы.

В модуле scrd06 происходит расчет перетечек нейтронов между точками и поправок к этим перетечкам. Задача распараллеливания немного усложняется, поскольку перетечки между нодами зависят как от состояния самой ноды, так и от состояния соседних нод. Модуль scrd06 занимает порядка 5% времени расчета всей системы, однако он предоставляет большинство входных данных для модуля scrd03. Разработка параллельной версии модуля расчета коэффициентов перетечесcrd06 позволяет существенно снизить количество пересылок, что повышает производительность всей системы.

### 3. Распараллеливание расчета нейтронно-физических констант

Расчет нейтронно-физических констант основывается на сеточном методе вычислений. Расчетная сетка является цилиндром высотой в 15 точек и с 163 точками в основании. Основание поделено на 7 областей, а именно - центральную в виде круга и 6 секторов. По высоте расчетная сетка поделена на 5 групп по 3 слоя в каждой. Схематичное изображение расчетной сетки представлено на рисунке 1, где цифрами обозначено количество точек расчетной сетки в области. На рисунке 2 представлена расчетная сетка в горизонтальной проекции. В каждой точке указанной сетки выполняется расчет коэффициентов для органов регулирования систем управления защиты и для тепловыделяющей сборки, а затем пересчет по ним констант в каждой точке. Особенностью данных вычислений в плане возможности распараллеливания является то, что для каждой точки значения полиномов зависят только от теплогидравлических параметров в самой этой точке и не зависят от параметров других точек. Из этого следует, что расчет в каждой области расчетной сетки может осуществляться независимо друг от друга, а следовательно, может быть распараллелен без дополнительных обменов данными, кроме рассылки начальных параметров расчета.

Были разработаны две схемы, использующие различные методы распараллеливания - статический и квазистатический для произвольного количества процессов. Данный подход обусловлен различной спецификой этих методов. При статическом распараллеливании снижаются затраты на накладные расходы за счет того, что область расчетов для каждого процесса определена на этапе реализации алгоритма. Однако при использовании большого количества процессов КПД распараллеливания будет падать, как следствие, придется выбирать оптимум с точки зрения затраченные ресурсы / производительность. Квазистатическое распараллеливание, в отличие от статического, позволяет подобрать описанное оптимальное количество процессов. Далее представлены схемы указанных методов.

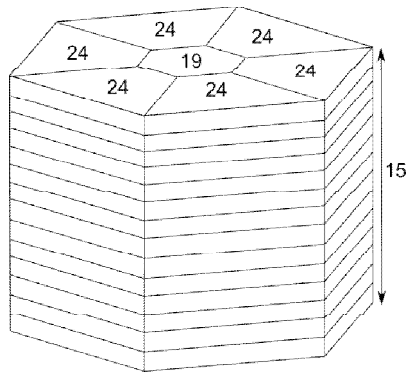


Рис. 1. Разбиение А3 реактора.

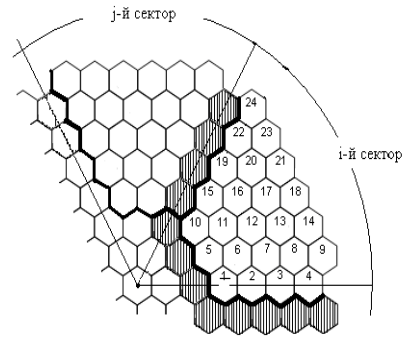


Рис. 2. Разбиение А3 реактора в горизонтальной проекции.

Для использования произвольного количества процессов вводится одномерная параметризация областей расчетной сетки. Каждая область нумеруется числом от 1 до 105, что сводит распределение вычислительной работы между процессорными элементами к разбиению множества 105 целых чисел на определенное количество подмножеств. В настоящее время процесс с рангом  $n$  вычисляет полиномы в областях с номерами от  $(n-1) \cdot 3 + 1$  до  $n \cdot 3$ , которые вычисляются по следующим формулам:

$$x_{i,j} = \dots$$

$$y_{i,j} = \dots$$

3

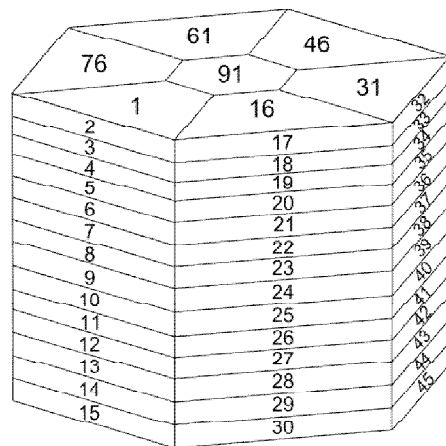


Рис. 3. Одномерная параметризация расчетной области.

Указанный метод гарантирует максимально равномерное распределение вычислительной нагрузки между произвольным количеством узлов, однако уступает по накладным расходам статическому распараллеливанию на несколько арифметических операций и одну передачу данных (параметры отображения трехмерной параметризации в одномерную). Следует также отметить тот факт, что подобный подход не зависит от способа нумерации расчетных областей. Следовательно, возможна оптимизация алгоритма с учетом архитектуры целевой вычислительной установки. В настоящее время используется одномерная параметризация, представленная на рисунке 3.

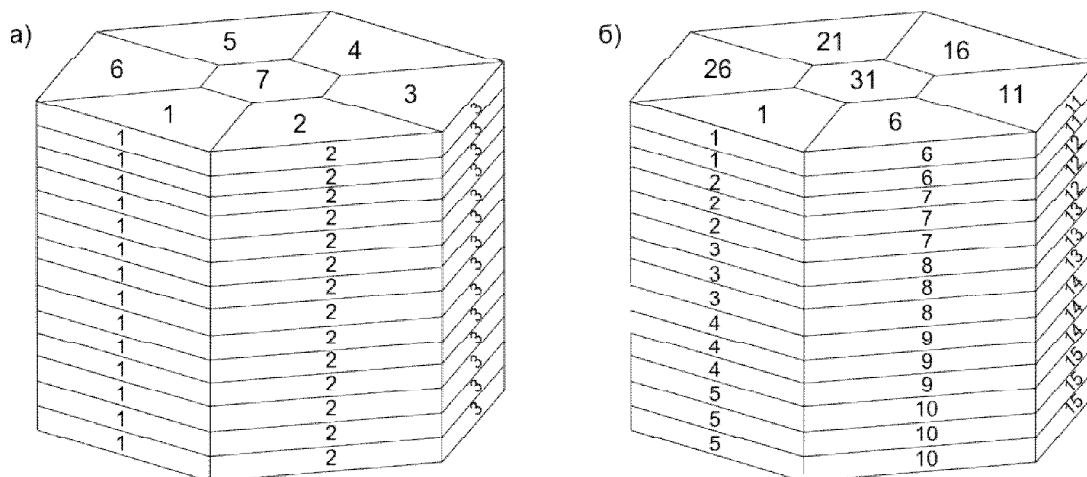


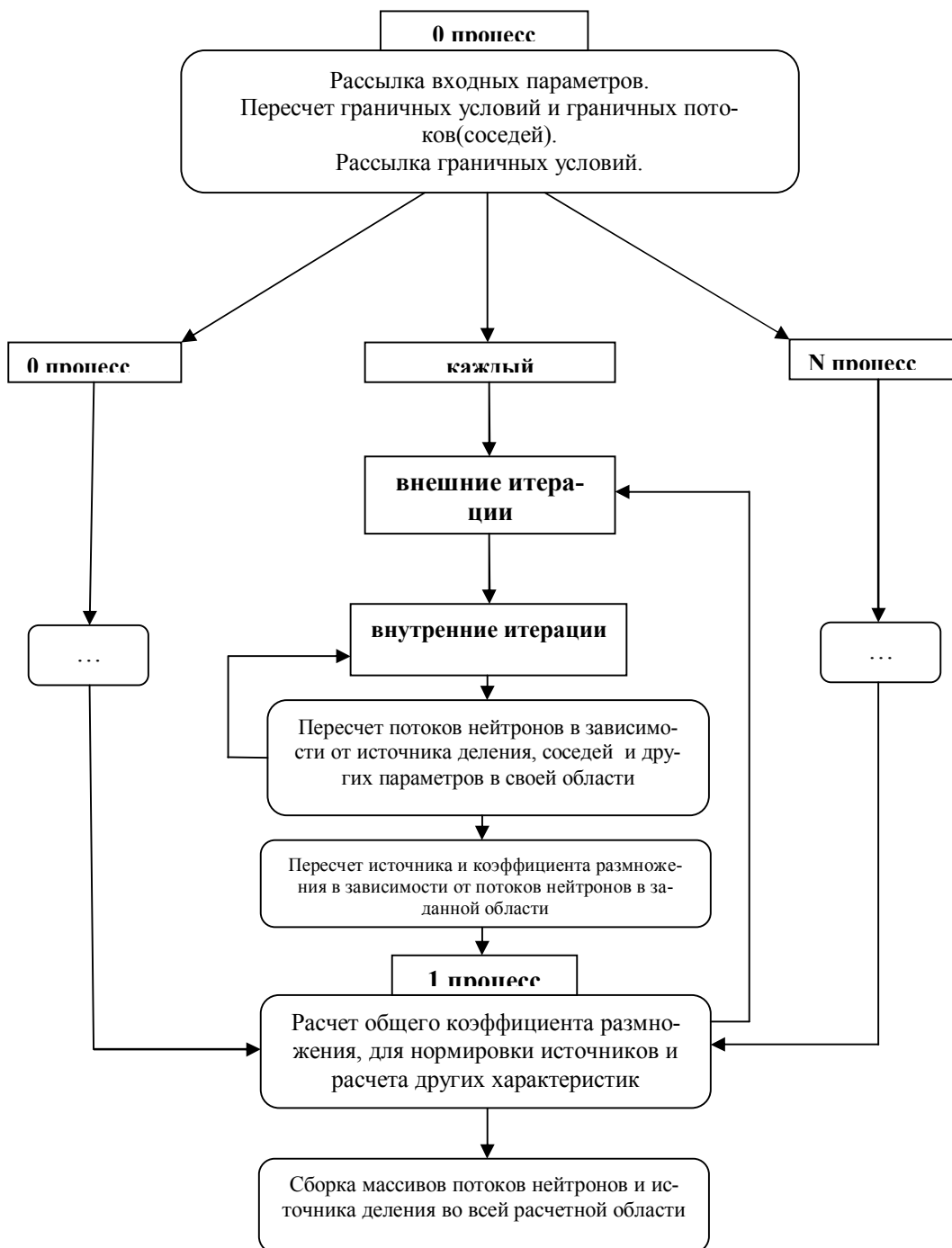
Рис. 4. Распределение расчетных областей по процессам.

Схема со статическим распараллеливанием является схемой расчета с минимальным количеством накладных расходов, которые обусловлены разнесением задачи на несколько процессоров. Данная схема была реализована для 7 и 35 процессоров. При запуске на 7 процессорах каждый процесс считает те области расчетной сетки, которым соответствуют одинаковые горизонтальные проекции. При запуске на 35 процессорах указанные области дополнительно делятся на 5 областей по вертикали. Распределение расчетных областей по процессам схематично изображено на рисунке 4 (часть а - для 7 процессоров, часть б - для 35), где цифрами указаны номера процессов, рассчитывающих константы в точках данной области. В целом необходимо отметить, что структура алгоритма и реализующего его программного модуля расчета нейтронно-физических констант имеет высокий внутренний параллелизм. Как следствие этот модуль распараллеливается с высоким КПД. Анализ эффективности реализации представленного метода распараллеливания можно найти в главе 6.

#### 4. Распараллеливание расчета пространственной функции

Расчет пространственной функции производится методом итерации источников [2]. Используется та же расчетная сетка, что и в расчете нейтронно-физических констант (см. главу 3). В указанном методе поток нейтронов в точке определяется значениями потока в этой и 8 соседних точках (для гексагональной геометрии), а также источником нейтронов в этой точке. Источник нейтронов в точке определяется суммой потоков нейтронов по группам в этой точке. Потоки и источники нейтронов определяются внутри итерационного цикла. Итерации прекращаются, когда эффективный коэффициент размножения нейтронов перестает меняться с заданной точностью. На каждой итерации производится нормировка источника нейтронов на суммарное энерговыделение по зоне. Таким образом, чтобы приступить к следующей итерации, вначале нужно узнать локальные потоки и источники нейтронов по всей активной зоне.

Расчет пространственной функции на текущий момент распараллелен статически для 7 и 35 процессоров, используя разбиение расчетной области, описанное в разделе 3. В отличие от модуля расчета нейтронно-физических констант, в модуле вычисления пространственной функции присутствуют зависимости по данным между гранулами параллелизма. По этой причине количество пересылок увеличивается, и, соответственно, эффективность распараллеливания снижается. На рисунке 5 схематично приводится алгоритм параллельного вычисления пространственной функции.



**Рис. 5.** Алгоритм вычисления пространственной функции.

На схеме алгоритма можно заметить, что во время выполнения внутренних итераций не используются данные, получаемые другими процессами во время выполнения итераций внутренних. Следовательно, весь цикл внутренних итераций может выполняться на удаленном вычислительном узле без дополнительных обменов данными. Однако при пересчете потоков нейтронов требуются данные о всей расчетной области, полученные на прошлой внешней итерации, например общий коэффициент размножения и поток нейтронов в соседних ячейках. По этой причине при инициализации новой внешней итерации реализован механизм коллективного обмена данными и синхронизация процессов.

Описанный выше алгоритм содержит коллективные обмены данными о потоках нейтронов в соседних ячейках. Они были реализованы по схеме "один-всем" и "все-одному". Указанный подход позволяет без существенных издержек поддерживать механизмы сохранения состояния

расчета, которые накладывают условие "все данные при выходе из модуля должны содержаться в памяти нулевого процесса".

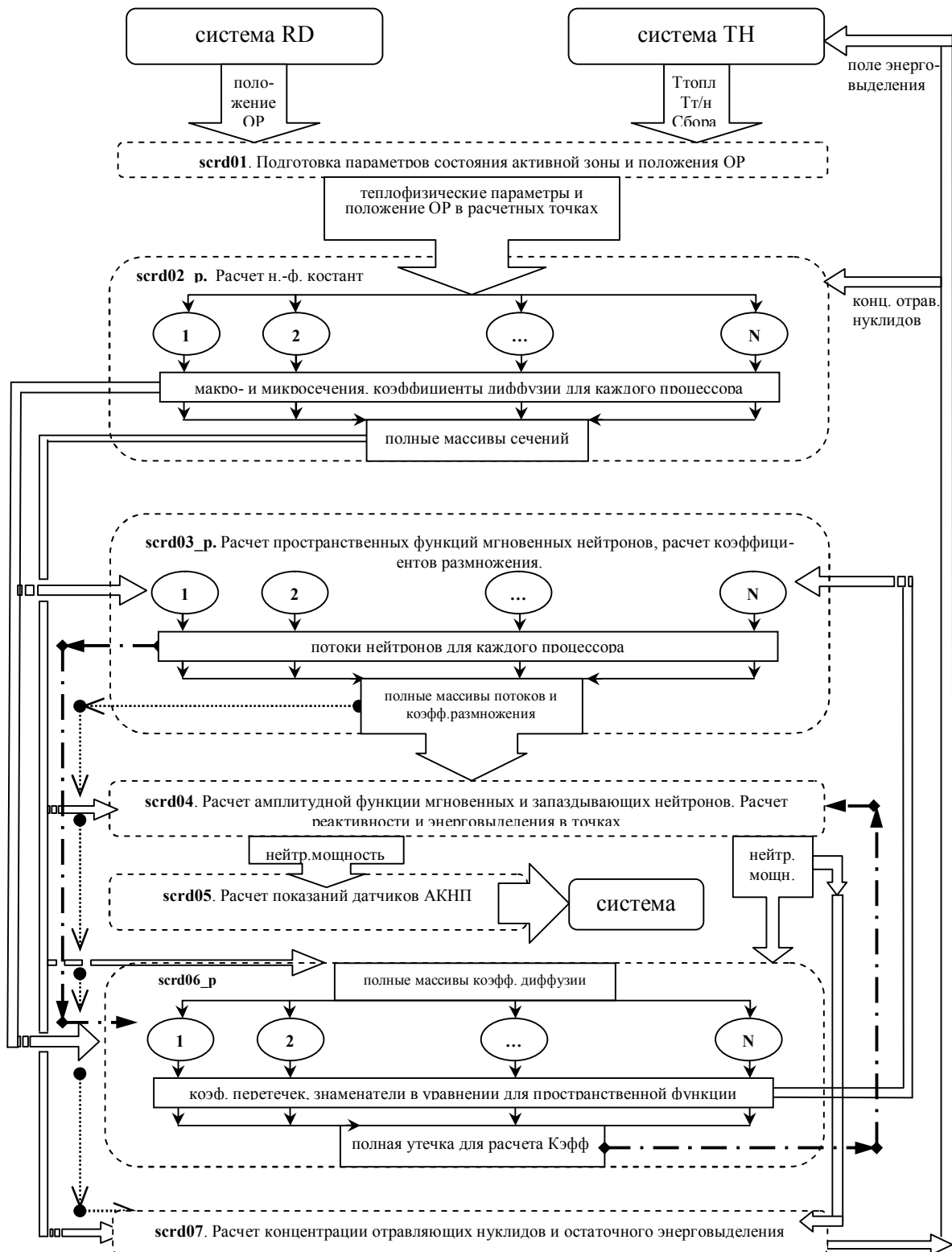


Рис. 6. Схема работы нейтронно-физического кода.



## 5. Интеграция параллельных модулей в расчетный код

В предыдущих разделах описана схема распараллеливания компонентов системы, которые занимают в сумме более 90% времени работы расчетного кода. Однако в нем также присутствуют модули, работающие последовательно на главном процессоре, и потому возникает проблема "стыковки" параллельных и последовательных частей расчета.

На схеме, изображенной на рисунке 6, можно видеть, что модуль расчета нейтронно-физических констант предоставляет данные для модулей, которые не подвергались модификации. В частности в `scrd02` рассчитываются полные массивы сечений, которые используются при вычислении амплитудной функции нейтронов, реактивности и энерговыделения (`scrd04`, `scrd07`). Аналогичные зависимости можно отметить и у расчета пространственной функции. В силу этих обстоятельств при завершении параллельного расчета необходим сбор указанных данных в памяти главного процесса, что несколько снижает эффективность распараллеливания.

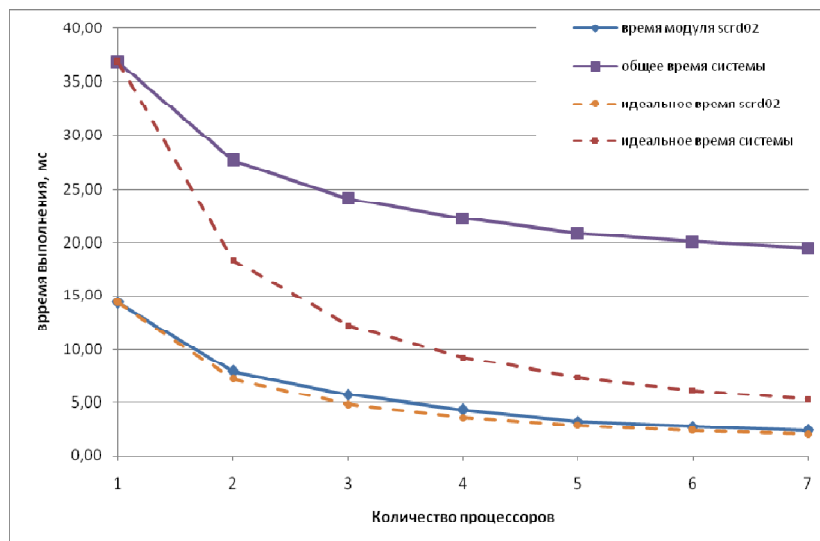
Подобных издержек можно избежать, разработав параллельные версии для всех модулей системы. В таком случае большая часть данных будет распределена между процессами и необходимость частых пересылок исчезнет. Однако этому препятствует несколько факторов. Во-первых, алгоритмы не всех компонентов системы имеют внутренний параллелизм и могут быть распараллелены. Во-вторых, нейтронно-физический расчетный код ТРЕК реализован в рамках среды моделирования S3, которая накладывает некоторые ограничения на запускаемые расчеты. В данной системе присутствует механизм сохранения и восстановления текущего состояния расчетов, что обуславливает необходимость в определенные моменты собирать все параметры расчета на главном процессоре. И в-третьих, данная задача достаточно трудоемка, что в совокупности с неизвестным по большей части результатом делает ее решение нецелесообразным в рамках данного исследования.

## 6. Оценка производительности расчета нейтронно-физических констант

Как было указано в главе 3, расчет нейтронно-физических констант имеет высокий внутренний параллелизм и распараллеливается с высокой эффективностью. Под эффективностью (или КПД) распараллеливания здесь и далее понимается отношение минимально возможного времени вычисления на заданном количестве процессоров к полученному времени расчета. Результаты временных раскладок это подтверждают. Была проведена промежуточная оценка времени выполнения модели с распараллеленным модулем `scrd02`. В качестве объекта тестирования был принят процесс моделирования минуты работы реактора. Тестирование проводилось на сервере ВНИИАЭС. Особенностью данной машины является то, что она использует общую память. Ее наличие сильно ускоряет пересылки данных. Результаты тестирования и их сравнение с идеальным распараллеливанием представлены на рисунке 7. Под идеальным распараллеливанием понимается распараллеливание без каких-либо накладных расходов, то есть ускоряющее расчет в  $N$  раз, если  $N$  - количество процессоров. Для оценки времени использовался профилировщик `gprof`.

Представленные на рисунке 7 данные указывают на эффективность распараллеливания при использовании небольшого количества процессоров. При работе на 7 и меньше процессорах КПД распараллеливания колеблется между 80% и 90%. Однако дальнейшие измерения на целевой универсальной компактной Супер-ЭВМ производства ФГУП «РФЯЦ-ВНИИЭФ» [3] выявили сильное падение КПД при увеличении количества процессоров.

Измерения производились при помощи встроенного механизма S3 для получения временной статистики. Он показывает время работы только контрольных модулей. Однако модуль `st4du1`, в котором вызывается расчет нейтронно-физических констант, содержит модули, которые работают последовательно. В силу этого обстоятельства замеры производились не только на главном процессоре, но и на одном второстепенном. Результаты времени счета второстепенного процесса могут быть несколько занижены в силу того, что во всех распараллеленных модулях присутствуют последовательные участки кода.



**Рис. 7.** Промежуточное тестирование параллельной версии расчета нейтронно-физических констант

Результаты, представленные в таблице 3, отражают среднее время выполнения модуля `cr4dy1` при моделировании 3 минут работы реактора. В таблице представлены результаты работы контрольного модуля `cr4dy1` при моделировании на 1, 7 и 35 процессорах. Результаты счета при параллельном моделировании представлены как на главном процессоре, так и на второстепенном.

**Таблица 3.** Время счета контрольного модуля `cr4dy1`.

время выполнения последовательного расчета, мс	время выполнения на 7 процессорах, мс		время выполнения на 35 процессорах, мс	
	главный процесс	второстепенный процесс	главный процесс	второстепенный процесс
17.92	3.43	3.27	3.45	1.39

Время работы распараллеленной модели существенно сокращается по сравнению с временем последовательного расчета, однако полученное на 7 процессорах ускорение сохраняется и на 35 процессорах. Для более подробного анализа производительности было принято решение отделить сам параллельный расчет и пересылки данных. Вследствие этого контрольный модуль `cr4dy1` был разделен на 4 контрольных модуля, описание которых представлены в таблице 4.

**Таблица 4.** Модификация контрольного модуля `cr4dy1`.

Контрольный модуль	Вызываемые расчетные модули	Описание
<code>cr4dy1_p1</code>		рассылка начальных входных теплофизических параметров для модуля <code>scrd02p</code>
<code>cr4dy1_p2</code>	<code>scrd01</code>	последовательный модуль интерфейса с системами (расчет идет только на главном процессоре)
	<code>scrd02p (scrd02p_1)</code>	расчет нейтронно-физических констант для ТВС (без обмена данными между процессами)
<code>cr4dy1_p3</code>		сбор посчитанных данных на главном процессе
<code>cr4dy1_p4</code>	<code>scrd04</code>	последовательный модуль расчета реактивности и амплитудной функции потоков нейтронов
	<code>scrd05</code>	Последовательный модуль расчета остаточного тепловыделения

Было измерено время выполнения всех указанных частей по тому же принципу (3 минуты работы реактора). Результаты приведены в таблице 5.

**Таблица 5.** Детализованное время исполнения модуля cr4dy1.

название модуля	время выполнения последовательного расчета, мс	время выполнения на 7 процессорах, мс		время выполнения на 35 процессорах, мс	
		главный процесс	второстепенный процесс	главный процесс	второстепенный процесс
cr4dy1_p1	17.92	0.12	0.31	0.24	0.74
cr4dy1_p2		2.69	2.62	0.62	0.53
cr4dy1_p3		0.51	0.34	2.48	0.12
cr4dy1_p4		0.11	0.00	0.11	0.00

Исходя из данных, представленных в таблице 5, можно утверждать, что резкое снижение КПД распараллеливания обусловлено все большей долей времени, затраченной на пересылки данных. Если при расчете на 7 процессорах она составляла 20% времени расчета, то при 35 уже 82%. В свою очередь скорость параллельных вычислений без передачи данных увеличивается более чем в 4,3 раза, что довольно близко к идеальному значению.

Таким образом, для увеличения эффективности вычислений следует снизить долю времени, затраченного на пересылки данных. Решение данной задачи является наиболее перспективным направлением дальнейшего исследования.

## 7. Производительность расчета пространственной функции

В таблице 6 приведены результаты распараллеливания модуля расчета пространственной функции (scrd03), произведенные по той же методике, как и в предыдущем разделе.

**Таблица 6.** Время расчета пространственной функции.

время выполнения последовательного расчета, мс	время выполнения на 7 процессорах, мс		время выполнения на 35 процессорах, мс	
	главный процесс	второстепенный процесс	главный процесс	второстепенный процесс
14.01	5.62	5.27	5.99	5.28

На 7 процессорах достигнуто ускорение всего в 2,5 раза, что составляет КПД 35%. На 35 процессорах ускорение немного ниже (2,33), что составляет КПД всего в 6%. Данный факт обуславливается тем, что сам алгоритм расчета пространственной функции имеет довольно тесные связи между гранулами параллелизма. Как следствие количество пересылок увеличивается, доля времени, затраченного на пересылки, растет, из-за чего падает производительность расчета.

К сожалению, из-за структуры алгоритма невозможно проанализировать долю времени, затраченного на пересылки. Однако по времени последовательного расчета и количеству пересылок в коде можно приблизительно ее оценить. Время последовательного исполнения модулей примерно одинаковое, однако при расчете нейтронно-физических констант функции коллективного обмена вызываются 5 раз (4 BCAST при рассылке входных параметров и 1 REDUCE при сборе посчитанных данных), а при расчете пространственной функции они вызываются более 40 раз, что больше на порядок. Исходя из этого можно предположить, что пересылки данных занимают почти все время расчета.

Таким образом, сокращение затрат на обмен данными становится основным способом оптимизации вычислений. Альтернативой обычному коллективному обмену данными является алгоритм передачи данных, учитывающий особенности разбиения расчетной сетки и алгоритма расчета. Основой такого подхода является тот факт, что для расчета каждому процессу нужны

данные не во всей расчетной сетке, а в ее части. Расчетная область каждого процесса граничит максимум с 8 областями других процессов (для разбиения на 35 процессов). В силу этого существует возможность применения децентрализованного обмена данными на этапе внешних итераций, который осуществляется только между процессами с соседними областями расчета.

Данный подход более затратен, как с точки зрения количества пересылок, так и с точки зрения вычисления номеров процессов с граничащими областями. Однако он позволяет проводить расчет асинхронно в большей части времени исполнения программы, что может дать существенный выигрыш по времени, который покроет излишние расходы на пересылки и вычисление соседей.

К тому же описанный выше подход может быть существенно оптимизирован под целевую аппаратную платформу. Это обуславливается тем, что пропускная способность канала между процессорами разная, следовательно, минимизировав количество общих границ расчетных областей "далеких" процессоров минимизируется среднее время одной пересылки. Указанный подход является довольно трудоемкой задачей, потому он не был реализован в рамках данного исследования. Однако его реализация и апробация является перспективным направлением для дальнейших работ по настоящей теме.

## 8. Заключение

Проведенные на данном этапе работы является лишь первым шагом к эффективной адаптации нейтронно-физического расчетного кода ТРЕК для использования на суперкомпьютерных платформах. В первую очередь необходима дальнейшая оптимизация механизмов обмена данными в системе.

Снижение затрат времени на передачи данных позволит использовать массовый параллелизм, присущий современным и архитектурным суперЭВМ, которые зачастую используют не только традиционные CPU, но и специализированные аппаратные ускорители.

Поддержка массового параллелизма предоставляет возможность для дальнейшего безболезненного увеличения точности расчетов за счет использования более точной расчетной сетки. Как правило, чем объемнее задача, тем лучше она решается распределенными вычислительными методами.

В случае невозможности повысить эффективность параллельной версии нейтронно-физического расчетного кода будет иметь смысл рассмотреть возможность перехода от итерационного метода поиска решения к вычислению точного решения. Данный метод не рассматривался в рамках данной работы, однако с учетом возможностей современных высокопроизводительных вычислительных установок он имеет право на жизнь.

## Литература

1. MPI-2: Extensionsto the Message-PassingInterface // Message PassingInterface Forum, <http://www.mpi-forum.org/docs/mpi-20.ps>, 1997г.
2. Вычислительные методы в физике реакторов // Сб. статей под ред. Х. Гринспена, К. Келбера, Д. Окрента, Атомиздат, М., 1972г.
3. Компактные Супер-ЭВМ производства ФГУП «РФЯЦ-ВНИИЭФ» // Эл. ресурс: <http://www.vniief.ru/directions/grazrab/catalog/infprod/razrabotki/super-evm.html>
4. Бартеньев О.В. Современный Фортран // Диалог-МИФИ, М., 2005г.
5. Глестон С, Эдлунд М., Основы теории ядерных реакторов // Иностранлит., М., 1954г.
6. Демидович Б.П., Марон И.А. Основы вычислительной математики. // Наука, М., 1966г.
7. Марчук Г.И., Численные методы расчета ядерных реакторов. // Атомиздат, М., 1958г.