

УДК 519.6

МЕТОДЫ УСКОРЕНИЯ РАСЧЕТА НАЧАЛЬНЫХ ДАННЫХ ПО КОМПЛЕКСУ ПРОГРАММ 3D-РНД ПРИ ЧИСЛЕННОМ МОДЕЛИРОВАНИИ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ В ПАРАЛЛЕЛЬНОМ РЕЖИМЕ

К. К. Олесницкая, А. Л. Потехин, Е. В. Потехина, В. И. Тарасов
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Описаны этапы и методы повышения эффективности расчета в параллельном режиме по комплексу программ 3D-РНД начальных данных задач математической физики в трехмерной постановке. В первом методе используется балансировка вычислительной нагрузки на основе оценки вычислительной сложности ячеек сетки, согласно которой осуществляется их распределение по процессам. Во втором методе выполняются деление всего объема ячеек на большое количество порций (блоков) и их динамическое распределение между процессами. Для сохранения рассчитанных данных в комплексе программ 3D-РНД используется библиотека ЕФР, которая обеспечивает эффективный доступ к файловым данным на многопроцессорных ЭВМ. Для методов описаны базовые алгоритмы расчета и сохранения начальных данных, проведен сравнительный анализ скоростных показателей расчета начальных данных ряда задач.

Ключевые слова: комплекс программ 3D-РНД, библиотека ЕФР, параллельный расчет, матричная декомпозиция, дисбаланс счетной нагрузки, балансировка счетной нагрузки, эффективность расчета.

Введение

Основными компонентами единой технологии подготовки начальных данных для численного моделирования задач математической физики в трехмерной постановке являются комплекс программ 3D-РНД [1] и библиотека ЕФР [2]. С использованием программы 3D-РНД строятся сетки и рассчитываются начальные данные для методик, численно моделирующих физические процессы в различных математических постановках — лагранжевой, лагранжево-эйлеровой и эйлеровой; в последних двух — с применением или без применения метода концентраций веществ [3]. Проблемно-ориентированная библиотека ЕФР обеспечивает эффективный доступ к файлам данных с учетом стратегии оптимального использования аппаратных и файловых ресурсов многопроцессорных ЭВМ.

По комплексу программ 3D-РНД сетки и начальные данные рассчитываются в два этапа.

На первом этапе прямым методом строится трехмерная матричная листовая лучевая сет-

ка [1]. При построении сетки учитываются требования методик к типу сетки и математической постановке задачи. В зависимости от постановки задачи при построении сетки границы раздела веществ могут учитываться (проходить по линиям сетки) или не учитываться.

На втором этапе, в случае, когда границы раздела веществ проходят по линиям сетки, в каждую ячейку сетки заносятся начальные данные одного вещества. Если границы раздела веществ проходят не по линиям сетки, в *смешанных* ячейках сетки рассчитываются концентрации веществ, с помощью которых выделяются границы раздела между веществами. В этом случае расчет концентраций веществ в ячейках сетки занимает основное время выполнения комплекса программ 3D-РНД.

Существуют два варианта комплекса программ 3D-РНД, обусловленные режимами выполнения: последовательный и параллельный.

Последовательный вариант функционирует под ОС Windows на ПЭВМ и имеет графиче-

ский интерфейс для интерактивного задания параметров расчета начальных данных. Используя этот вариант, можно проводить расчет начальных данных на сеточных моделях, содержащих 100—150 млн ячеек.

Параллельный вариант реализован с использованием MPI [4] для распределенной модели памяти и выполняется в пакетном режиме под ОС Unix на многопроцессорных ЭВМ. Задание параметров расчета осуществляется на ПЭВМ в последовательном варианте программы, а полученный набор файлов (файлы геометрии и файл проекта с заданными параметрами) переносится на многопроцессорную ЭВМ. Результатом расчета начальных данных в параллельном режиме является распределенный единый файловый разрез (ЕФР), содержащий информацию о сеточной модели, состоящей из нескольких миллиардов ячеек.

Необходимо отметить, что в целом задача расчета начальных данных и, в частности, построения сеток в параллельном режиме сама по себе является непростой. Однако данная работа посвящена не столько вопросам создания *параллельных* генераторов сеток, сколько вопросу оптимальной загрузки счетного поля многопроцессорной ЭВМ при расчете начальных данных с использованием структурированных сеток.

Два способа декомпозиции сеточных данных на матричные фрагменты

В параллельном варианте программы 3D-РНД при расчете сеток и начальных данных используется разбиение (декомпозиция) сеточных данных на матричные фрагменты. Декомпозиция сетки на фрагменты выполняется по матричным индексам в трех направлениях (i, j, k). Все значения индексов по каждому направлению разбиваются на интервалы (*интервалы разбиения*). Каждый интервал разбиения определяет *габарит* (размер) одного из счетных фрагментов в соответствующем направлении. Декомпозиция на фрагменты задается до начала расчета начальных данных сетки (в частности концентраций) двумя способами.

В первом способе для рационального разбиения задачи на фрагменты распределение индексов сетки по интервалам разбиения задается пользователем. Это приводит к дополнительным трудозатратам (зачастую при расчете ис-

пользуется несколько тысяч процессов), и, как правило, такая декомпозиция не всегда является эффективной с точки зрения времени расчета начальных данных.

Во втором способе для уменьшения времени расчета и снятия с пользователя нагрузки по заданию декомпозиции интервалы индексов сетки вычисляются автоматически исходя из общего количества процессов. В этом случае общее количество ячеек сеточной модели распределяется между фрагментами приблизительно равными порциями. Расчет сеток и начальных данных во фрагментах проводится каждым процессом независимо, т. е. обмен сеточными данными между процессами отсутствует.

Использование такой матричной декомпозиции на ряде задач выявило существенный недостаток. Фрагменты сетки, состоящие из одинакового количества ячеек, могут иметь разную сложность вычисления концентраций, что приводит к дисбалансу счетной нагрузки и большим временам расчета. В связи с этим стала актуальной задача реализации алгоритма балансировки вычислительной нагрузки по процессам.

Статическая балансировка вычислительной нагрузки на основе временных оценок обработки ячеек сетки

Анализ методов и алгоритмов, используемых для балансировки вычислительной нагрузки в параллельном режиме, показал, что все они так или иначе учитывают сложность модели, используя оценку времени обработки ее элементов (вершин, ребер графа сеточной модели), значения арифметической нагрузки для вычисления физических параметров в ячейке, весовые коэффициенты частиц и т. д. [5—12]. Фактически на основе ранее полученных данных вычисляются веса элементов модели и исходя из них с помощью алгоритмов балансировки реализуется попытка равномерно распределить вычислительную нагрузку между параллельными процессами.

В комплексе программ 3D-РНД выполнить балансировку вычислительной нагрузки на этапе задания матричной декомпозиции практически невозможно. Такая возможность появляется только после этапа построения сетки, когда можно провести оценку вычислительной сложности ячеек. Под вычислительной сложностью ячейки в данном случае понимается время расчета кон-

центраций находящихся в ней веществ. В основе алгоритма расчета концентраций лежит процедура нахождения пересечения лучей сетки с телами исходной трехмерной геометрической модели. Балансировка вычислительной нагрузки на основе вычислительной сложности ячеек проводится один раз, назовем ее *статической*.

Алгоритм расчета начальных данных с использованием статической балансировки состоит из следующих шагов:

1. Пользователем или автоматически выполняется начальная декомпозиция сетки на фрагменты. Каждый фрагмент рассчитывается своим процессом.
2. Каждый процесс предварительно выполняет построение сетки в своем фрагменте. Для оценки сложности ячеек фиксируется время выполнения процедуры нахождения пересечения лучей сетки с телами исходной трехмерной модели. Затем вычисляется время расчета концентраций в ячейках фрагмента сетки.
3. Полученные оценки собираются со всех процессов, и на их основе формируются сеточные индексы новых фрагментов, так чтобы суммарная сложность расчета ячеек в них была примерно одинаковой.
4. Далее для новой декомпозиции проводится полный расчет начальных данных задачи — повторное построение сетки, формирование распределения веществ, их концентраций.

Этапы формирования новых фрагментов схематично представлены на рис. 1 (см. также цветную вкладку).

На первом этапе формирования новых фрагментов сетки выбирается приоритетное индексное направление, т. е. направление с максимальным суммарным временем выполнения процедуры нахождения пересечения лучей сетки с телами исходной трехмерной модели. Индексы ячеек по этому направлению распределяются по интервалам разбиения в первую очередь. Суммарные оценки сложности на интервалах разбиения должны быть примерно одинаковыми: интервал увеличивается до тех пор, пока суммарная оценка сложности на нем не превзойдет среднюю суммарную оценку на выбранном направлении.

На втором этапе для вспомогательных фрагментов, полученных на первом этапе (на цветном рис. 1 слева они выделены синим, красным, зеленым и коричневым цветом), проводится пересчет суммарных оценок сложности и выбор следующего направления разбиения индексов. На каждом вспомогательном фрагменте направление разбиения может быть выбрано независимо от соседних вспомогательных фрагментов. Разбиение индексного направления внутри вспомогательного фрагмента осуществляется так же, как для приоритетного индексного направления (этап 1). Для полученных вспомогательных фрагментов (на цветном рис. 1 в центре они выделены оттенками синего, красного, зеленого и коричневого цвета) также пересчитываются суммарные оценки вычислительной сложности ячеек.

На третьем этапе распределения рассматриваются вспомогательные фрагменты, полученные на втором этапе. В них разбивается оставшееся индексное направление (у каждого фрагмента свое).

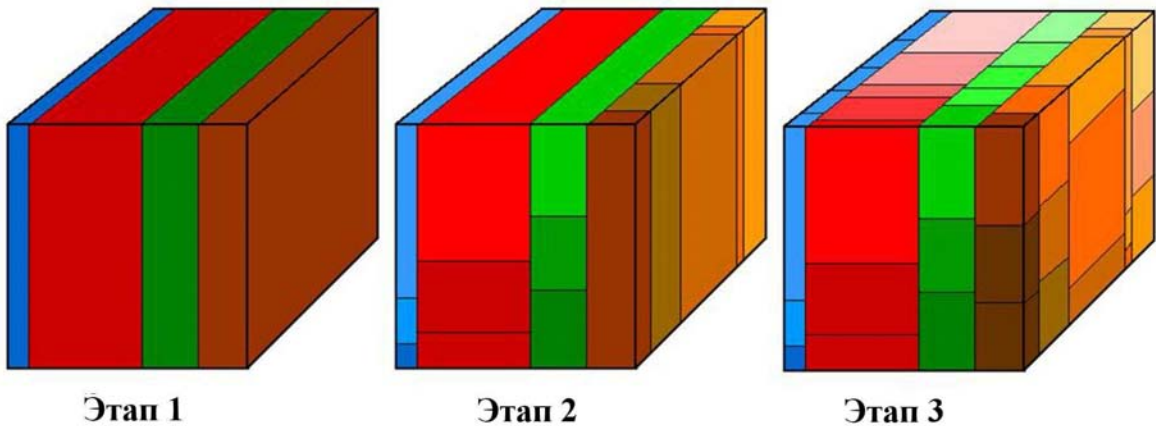


Рис. 1. Этапы формирования новых фрагментов

Алгоритм распределения ячеек сетки не меняет общего количества фрагментов. Разбиение на фрагменты по индексным интервалам теряет регулярную структуру, при этом фрагменты имеют примерно одинаковый вычислительный вес расчета концентраций.

Алгоритм статической балансировки дал на ряде задач ускорение расчета начальных данных в 1,5–2 раза, но на некоторых — замедление.

Замедление связано со следующими тремя причинами. Во-первых, на предварительном этапе расчета фрагмента сетки можно сделать только оценку вычислительной сложности столбца ячеек.* Это означает, что определить вычислительную сложность одной конкретной ячейки можно лишь приблизительно, поделив сложность столбца на число ячеек в нем. Во-вторых, на стадии счета концентраций для каждого столбца формируется около ста лучей, а не четыре, как при построении сетки. Если время обработки лучей внутри столбца будет существенно отличаться от времени обработки лучей на его ребрах, теоретическая и практическая оценка сложности также будут существенно отличаться. В-третьих, при смене декомпозиции меняются габариты фрагментов, а значит, потенциально меняется набор тел, обрабатываемых процессом. Это приводит к тому, что оценка сложности ячеек при исходной декомпозиции не совпадает с оценкой сложности ячеек при их новом распределении.

В силу указанных обстоятельств даже существенное изменение фрагментов по количеству ячеек может не дать ожидаемого результата по ускорению расчета начальных данных.

Невозможность качественного предсказания весовых оценок на предварительном этапе стала причиной отказа от использования классических весовых подходов балансировки и потребовала реализации принципиально нового алгоритма, который распределяет вычислительную нагрузку между процессами без вычисления весовых оценок.

Динамическая балансировка вычислительной нагрузки за счет деления сеточных фрагментов на блоки

Суть алгоритма заключается в разбиении фрагментов задачи, полученных в результате на-

чальной декомпозиции, на блоки, количество которых значительно больше количества используемых процессов, и динамическом распределении блоков между процессами по мере их готовности к обработке блока. Под блоком понимается часть листа регулярной сетки, ограниченная габаритами фрагмента задачи, которому она принадлежит. Таким образом, количество блоков в каждом фрагменте зависит от количества листов в нем, а общее число блоков составляет сумму листов по всем фрагментам. Рассматриваемый алгоритм назовем алгоритмом динамической балансировки.

На рис. 2 показана схема динамического распределения блоков по процессам.

Нулевой процесс выполняет функции управления: разбивает фрагменты на блоки, распределяет блоки по процессам, контролирует ход расчета и работу с файлами-фрагментами, собирает интегральную информацию и формирует головной файл. Остальные процессы являются *рабочими*, т. е. рассчитывают сетки, концентрации и дополнительные величины.

По мере готовности рабочие процессы посылают управляющему запрос на обработку очередного блока. Исходя из информации о состоянии блоков, управляющий процесс выбирает свободный блок и отправляет его номер процессу, который сделал запрос. Для этого на управляющем процессе создается и постоянно обновляется *карта фрагментов*, которая содержит информацию о состоянии блоков. Значения ее элементов (номер элемента — номер блока) интерпретируются следующим образом:

- 0 — блок не был обработан (является свободным);
- номер рабочего процесса со знаком «–» — блок обрабатывается указанным процессом;
- номер рабочего процесса со знаком «+» — блок уже обработан указанным процессом.

Эта информация позволяет быстро выбрать свободный блок или сделать вывод о том, что все блоки обработаны.

Поиск свободного блока сначала осуществляется среди необработанных блоков фрагмента, приписанного процессу, который сделал запрос. Если фрагмент обработан полностью, свободный блок выбирается из фрагмента, в котором содержится наибольшее число необработанных блоков. Блоки из фрагментов берутся по порядку их следования в карте фрагментов. При таком подходе рабочий процесс может рассчитать лю-

*Под столбцом понимается набор ячеек, заключенных между четырьмя соседними лучами сетки.

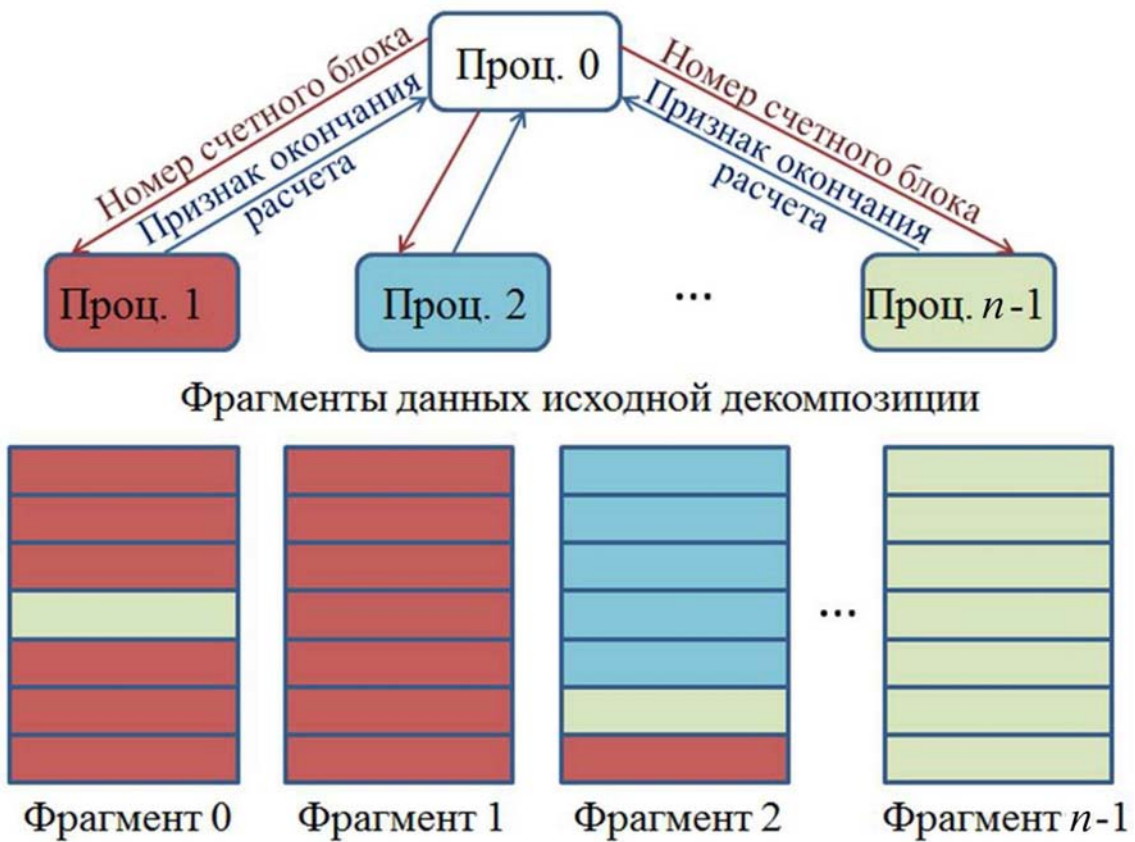


Рис. 2. Схема динамического распределения блоков по процессам

бое количество блоков любых фрагментов. Данная стратегия позволяет лучше балансировать счетную нагрузку, поскольку расчет "тяжелых" фрагментов будет проводиться большим числом процессов.

Алгоритм динамической балансировки позволил ускорить расчет начальных данных во всех задачах, причем в некоторых — в несколько раз.

Формирование файла ЕФР в параллельном режиме

Для сохранения данных, рассчитанных в комплексе программ 3D-РНД, используется библиотека ЕФР, которая предоставляет унифицированный формат представления расчетных данных в файле и функциональный интерфейс для эффективной работы с ними.

На рис. 3 показана схема расчета и сохранения начальных данных в параллельном варианте программы 3D-РНД. По окончании расчета каждый процесс записывает рассчитанный фрагмент задачи в свой файл-фрагмент независимо от других процессов. Управляющий про-

цесс дополнительно формирует головной файл, который содержит общую информацию по задаче, интегральную информацию о сеточных данных по фрагментам задачи и информацию о распределении данных по файлам-фрагментам.

Головной файл вместе с полученным набором файлов-фрагментов образуют единый для методик файловый разрез хранения сеточных данных (файл ЕФР). Работа с набором файлов осуществляется в параллельной файловой системе Lustre [13, 14].

Расчет начальных данных с использованием статической балансировки не потребовал изменения программ сохранения рассчитанных данных. Здесь отличие заключается лишь в том, что распределение данных по процессам и файлам-фрагментам соответствует не начальной матричной декомпозиции, а новой декомпозиции, нерегулярной по интервалам индексов сетки.

Расчет начальных данных с использованием динамической балансировки потребовал реализации нового подхода к формированию ЕФР (рис. 4).

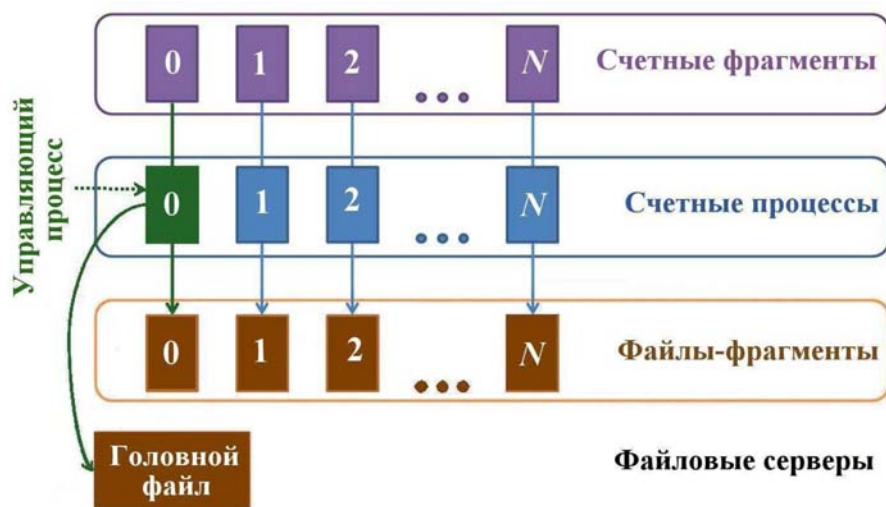


Рис. 3. Схема расчета и сохранения начальных данных в параллельном режиме

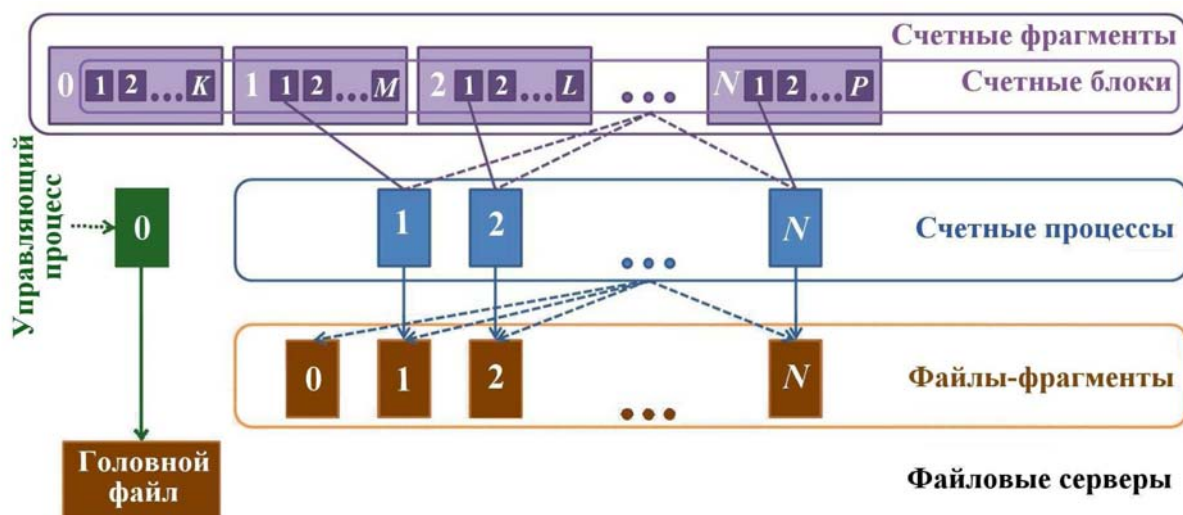


Рис. 4. Формирование ЕФР в параллельном режиме при использовании динамической балансировки

При использовании динамической балансировки запись начальных данных выполняется блоками в файл-фрагмент, ассоциированный с фрагментом начальной матричной декомпозиции. Когда все блоки записаны в файлы-фрагменты, управляющий процесс формирует головной файл ЕФР.

При записи возможна ситуация, когда несколько процессов одновременно записывают свой блок в один и тот же файл-фрагмент. При этом возникают две проблемы: необходимо обеспечить когерентность данных параллельной файловой системы и когерентность кэша библиотеки ЕФР. Кэш библиотеки ЕФР служит для

минимизации операций доступа к часто используемым данным файла и является локальным для каждого процесса. В кэше библиотеки ЕФР хранятся метаданные о структуре файла для быстрого позиционирования нужных данных.

Для решения указанных проблем был организован последовательный доступ рабочих процессов на запись в один файл-фрагмент и обеспечена синхронизация данных кэша библиотеки ЕФР на процессах. Координация выполнения записи рабочими процессами осуществляется управляющим процессом.

Запись блока в файл-фрагмент выполняется рабочим процессом по следующему алгоритму.

Сначала процесс проверяет необходимость создания файла-фрагмента. Если файла-фрагмента не существует, то он создается. Иначе файл-фрагмент открывается на редактирование. Если файл-фрагмент уже открыт процессом, то проверяется его предыдущий *владелец*. Если предыдущим владельцем был другой процесс, то выполняется синхронизация данных кэша библиотеки ЕФР путем двухточечных межпроцессорных обменов.

Затем процесс проверяет необходимость создания в файле фрагмента, соответствующего рассчитанному блоку. Если такого фрагмента не существует, то он создается.

Далее выполняется запись данных блока в файл-фрагмент. Если запись прошла успешно, то рабочий процесс отправляет управляющему процессу информацию о том, что блок записан и файл-фрагмент свободен.

Управляющий процесс, получив информацию об освобождении файла-фрагмента, проверяет очередь процессов на захват этого файла. Если очередь не пуста, управляющий процесс определяет для файла-фрагмента ряд характеристик и отправляет их процессу, первому в очереди на захват файла-фрагмента. Если очередь пуста, то файл-фрагмент помечается свободным на запись.

Алгоритм обеспечивает корректность данных в файлах-фрагментах, записанных несколькими процессами, и обеспечивает корректную работу с файлами в параллельной файловой системе.

Исследование времени и эффективности расчета концентраций в больших задачах

При оценке влияния различных типов балансировки на скорость расчета концентраций веществ как самого длительного этапа расчета начальных данных рассматривалось семь задач (таблица) с различным количеством веществ (тел) (см., например, рис. 5) и с разной степенью сеточной детализации геометрической модели. В задачах строилась равномерная прямоугольная сетка без учета границ раздела между веществами (рис. 6, см. также цветную вкладку); в этом случае для всех веществ рассчитываются концентрации. Расчет начальных данных проводился при разном количестве процессов. На рис. 7 представлено время расчета концентраций веществ для указанных задач (N — номер зада-

Задачи, рассчитанные в параллельном режиме

Номер задачи	Число тел	Число ячеек	Число процессов
1	42	$340 \times 1\,300 \times 680$	2 304
2	41	$1\,000 \times 1\,000 \times 1\,000$	4 096
3	355	$1\,000 \times 1\,000 \times 1\,000$	4 096
4	43	$1\,000 \times 1\,000 \times 1\,000$	240
5	40	$400 \times 200 \times 800$	64
6	1 495	$1\,840 \times 1\,040 \times 1\,040$	1 800
7	825	$2\,000 \times 2\,300 \times 2\,000$	3 000

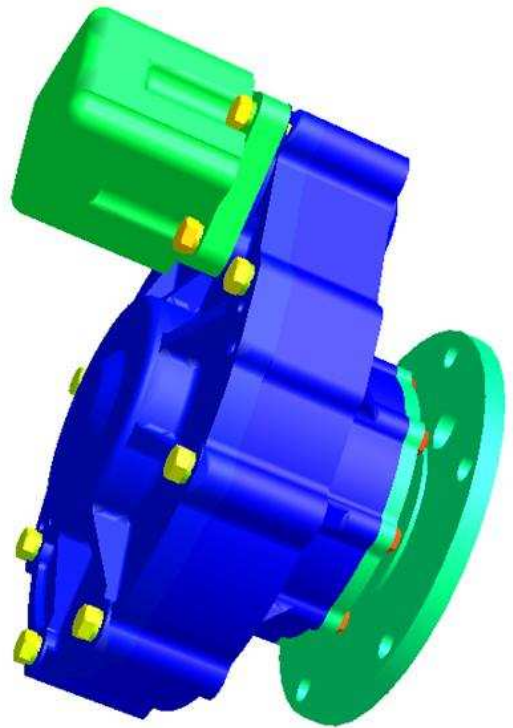


Рис. 5. Пример трехмерной геометрии, состоящей из 40 тел сложной формы, из задачи 5

чи) с использованием двух типов балансировки и без нее.

Как было сказано выше, использование статической балансировки может не дать ожидаемого результата по ускорению расчета начальных данных. Из рис. 8 видно, что в задачах 2, 3 и 4 применение статической балансировки привело к замедлению счета. Для задачи 7 расчет начальных данных с использованием статической балансировки отсутствует. Это связано с ограничением по памяти, поскольку при расчете на-

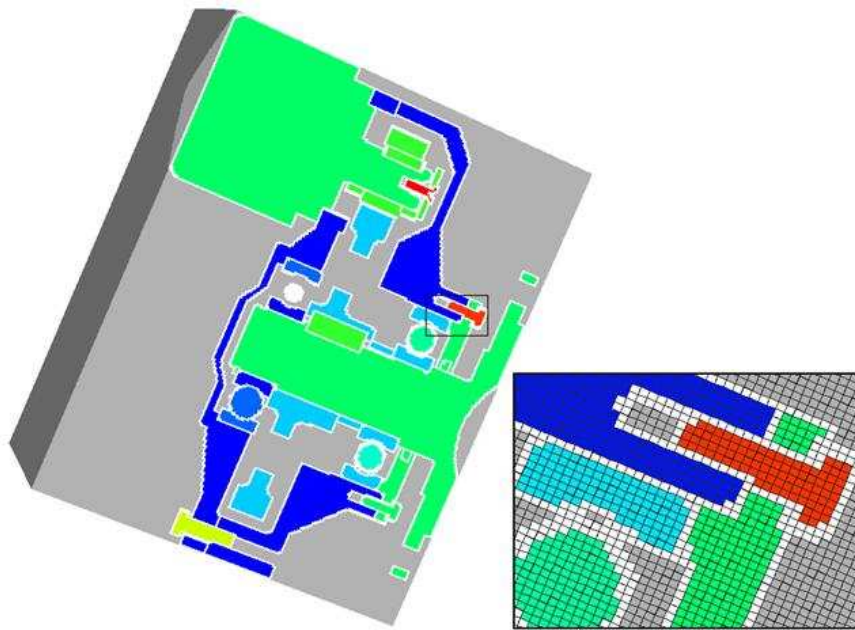


Рис. 6. Фрагмент структурированной прямоугольной сетки с расчетом распределения веществ по ячейкам

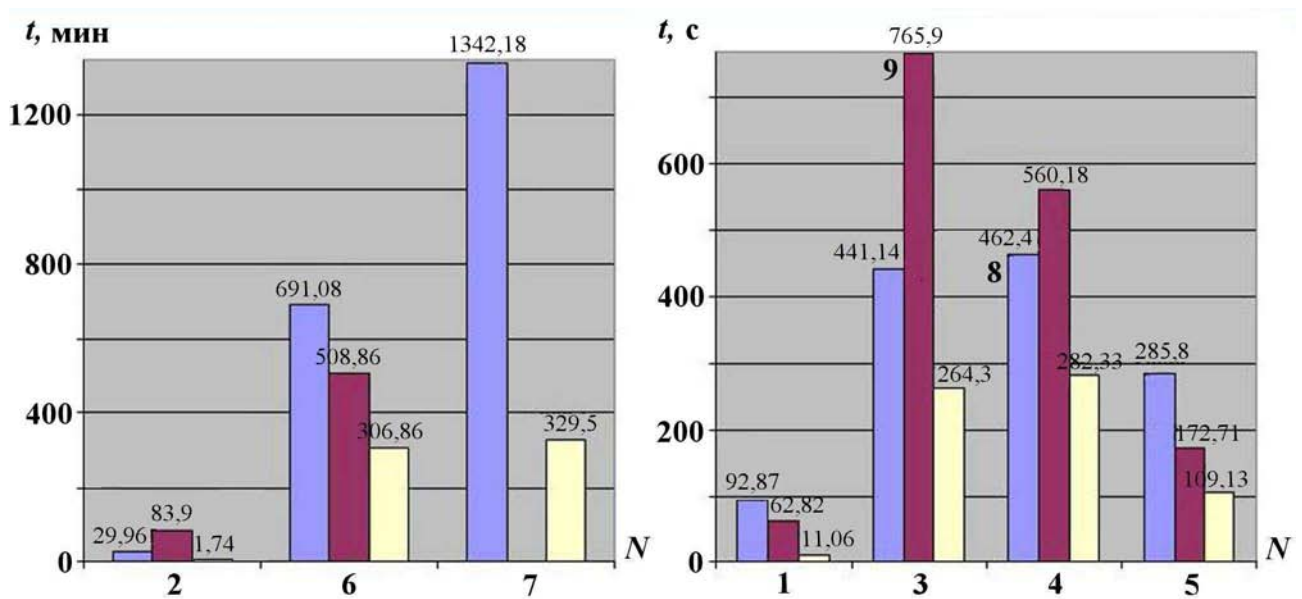


Рис. 7. Время расчета концентраций с использованием различных типов балансировки вычислительной нагрузки: слева — без балансировки; в центре — со статической балансировкой (для задачи 7 отсутствует); справа — с динамической балансировкой

чальных данных задачи 7 на нулевом процессе необходимо собрать временные оценки более чем для 9 млрд ячеек сетки.

При использовании динамической балансировки все расчеты были проведены в два и более раз быстрее, чем без нее.

На рис. 8 показана эффективность распараллеливания, рассчитанная по формуле

$$E = \frac{\sum_{i=1}^N t_i}{t_{\max} N} \cdot 100 \%,$$

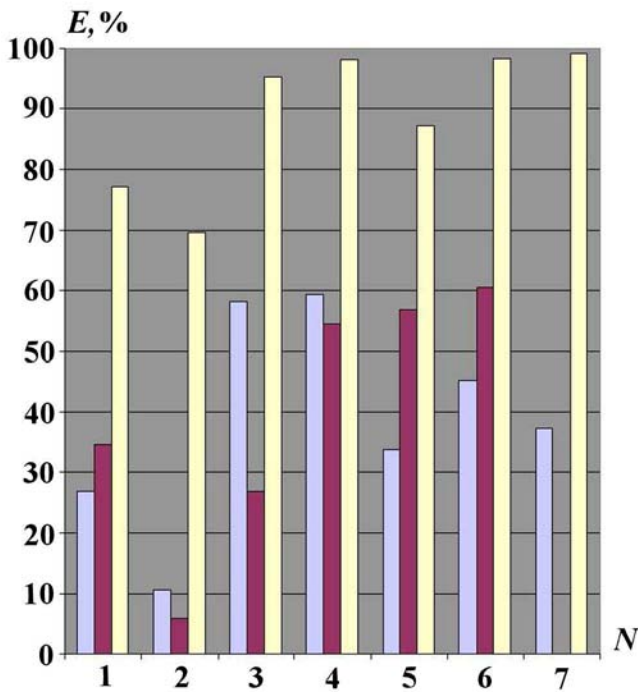


Рис. 8. Эффективность расчета концентраций с использованием различных типов балансировки вычислительной нагрузки: слева — без балансировки; в центре — со статической балансировкой (для задачи 7 отсутствует); справа — с динамической балансировкой

где $\sum_{i=1}^N t_i$ — суммарное время выполнения процессов по расчету концентраций; t_i — время выполнения i -го процесса; t_{\max} — максимальное время выполнения одного из процессов; N — число процессов.

В самом худшем случае эффективность расчета начальных данных с использованием динамической балансировки составила около 70 %, в то время как эффективность расчета без балансировки составила около 10 %.

Выводы

Разработаны и реализованы два подхода к балансировке счетной нагрузки во время подготовки начальных данных по комплексу программ 3D-РНД. Использование весовых оценок в подходе со статической балансировкой не дало ожидаемых результатов, в ряде случаев счет замедлился.

Расчеты больших трехмерных задач показали применимость реализованного в комплексе программ 3D-РНД алгоритма динамической ба-

лансировки загрузки процессов и его эффективность не менее 70 %. Динамическая балансировка дала ускорение для всех проведенных расчетов. Это относится и к задачам, в которых использование статической балансировки привело к замедлению счета.

На сегодняшний день работы по созданию алгоритмов балансировки в комплексе программ 3D-РНД завершены. Динамическая балансировка успешно применяется при расчете начальных данных в параллельном режиме для всех методик РФЯЦ-ВНИИЭФ.

Список литературы

1. Тарасов В. И., Ребров С. В., Волгин А. В. и др. Расчет начальных данных трехмерных задач по программе 3D-РНД // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2006. Вып. 3. С. 66—71.
2. Олесницкая К. К., Антипин И. А., Шубина М. А. Библиотека ЕФР для масштабируемого доступа к файловым данным на многопроцессорных ЭВМ // XI Забабахинские науч. чтения: сб. тезисов. Снежинск, 2012. С. 335—336.
3. Бахрах С. М., Глаголева Ю. П., Самигулин М. С. и др. Расчет газодинамических течений на основе метода концентраций // Докл. АН СССР. 1981. Т. 257, № 3. С. 566—569.
4. MPI: A Message-Passing Interface Standard Version 2.2. <http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf>.
5. Воронинов А. А. Декомпозиция данных для распараллеливания методики ТИМ-2D и критерии оценки ее качества // Вестник ЮУрГУ. Сер. Математическое моделирование и программирование. 2009. № 37(170). Вып. 4. С. 40—50.
6. Веселов Р. А. Алгоритмы балансировки комплекса р-МД с учетом вычислительной загрузки процессоров // Тр. XIII межд. семинара "Супервычисления и математическое моделирование". Саров: РФЯЦ-ВНИИЭФ, 2012. С. 140—146.
7. Беляев С. П., Дегтяренко Л. И., Турутина И. Ю. Проблемы и возможности разработки параллельных программ с динами-

- ческой балансировкой вычислительной нагрузки для решения задач механики сплошной среды // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 1996. Вып. 4. С. 43–44.
8. *Беляев С. П., Турутина И. Ю.* Алгоритмы автоматического перераспределения вычислительной нагрузки // Там же. С. 45–48.
 9. *Бутнев О. И., Пронин В. А.* Алгоритмы декомпозиции нерегулярной сетки с учетом вычислительной нагрузки // Там же. 2004. Вып. 4. С. 81–87.
 10. *Юрич М. Ю.* Анализ систем и методов балансировки нагрузки вычислительных систем // Науковий вісник Чернівецького університету. Сер. Комп'ютерні системи та компоненти. 2010. Т. 1. Вып. 1. С. 14–19.
 11. *Волков К. Н.* Балансировка нагрузки процессоров при решении краевых задач механики жидкости и газа сеточными методами // Вычислительные методы и программирование. 2012. Т. 13. С. 107–129.
 12. *Бершадский А. М., Курилов Л. С., Финогеев А. Г.* Исследование стратегий балансировки нагрузки в системах распределенной обработки данных // Известия вузов. Поволжский регион. Технические науки. 2009. Вып. 4. С. 38–48.
 13. Lustre. <http://www.whamcloud.com>, <http://www.lustre.org>.
 14. Keeping the Lustre File System Open. <http://www.opensfs.org>.

Статья поступила в редакцию 23.07.14.

ACCELERATION METHODS FOR INPUT DATA CALCULATION UNDER THE 3D-RND PROGRAM COMPLEX AT THE PARALLEL NUMERICAL SIMULATION OF MATHEMATICAL PHYSICS PROBLEMS / K. K. Olesnitskaya, A. L. Potekhin, E. V. Potekhina, V. I. Tarasov (FSUE "RFNC-VNIIEF", Sarov, Nizhny Novgorod Region).

The paper describes the phases and methods of the parallel calculation efficiency improvement under the 3D-RND complex for the input data calculation of the mathematical physics problems in the 3D setting. In the first method we use the computing load balancing basing on the estimation of the grid cell complexity, according to which they are distributed over the processors. In the second method the whole scope of the cells is divided into big amount of portions (blocks) and their dynamic distribution between the processors. To preserve the computed data in the 3D-RND program complex the EFR library is used, which provides the effective access to the file data on the multi-processor computers. We describe, for the methods, the basic algorithms of calculation and input data preservation, the comparative analysis of the input data calculation rate parameters for a number of problems.

Keywords: 3D-RND program complex, EFR library, parallel computations, matrix decomposition, computation load, computation load balancing, computation efficiency.
