

Метод трехуровневого распараллеливания методики ТИМ-2D

А.А. Воропинов

ФГУП Российский Федеральный Ядерный Центр –
Всероссийский Научно-Исследовательский Институт Экспериментальной Физики

Методика ТИМ-2D предназначена для решения задач механики сплошной среды на неструктурированных многоугольных лагранжевых сетках произвольной структуры. Для методики используется метод трехуровневого распараллеливания. На первом уровне осуществляется распараллеливание счета по математическим областям. На втором уровне производится распараллеливание счета внутри математической области по параобластям. На первых двух уровнях используется модель распределенной памяти и интерфейс MPI. На третьем уровне осуществляется распараллеливание итераций счетных циклов в модели общей памяти с использованием интерфейса OpenMP. Уровни могут использоваться как по отдельности, так и в различных сочетаниях при решении одной задачи.

1. Введение

Методика ТИМ-2D [1] предназначена для решения двумерных нестационарных задач механики сплошной среды на неструктурированных многоугольных лагранжевых сетках произвольного вида. Ячейками сетки могут быть не самопересекающиеся многоугольники, в узлах может сходиться произвольное количество ребер. Для расчета задач газодинамики и упругопластичности используются явные конечно-разностные схемы. Для моделирования диффузионных приближений используются неявные конечно-разностные схемы. Кинематические величины отнесены к узлам разностной сетки, термодинамические величины к ячейкам. Для решения уравнений механики сплошных сред используется цилиндрическая или плоская декартова система координат.

Для получения результатов в сжатые календарные сроки с необходимой точностью используются различные методы распараллеливания, позволяющие рассчитывать задачу одновременно на большом количестве процессорных ядер. В настоящее время наиболее распространенными прикладными стандартами для распараллеливания являются: интерфейс передачи сообщений MPI, для модели распределенной памяти, и интерфейс OpenMP, предназначенный для модели общей памяти.

При решении сложных двумерных задач начальную геометрию системы часто приходится разбивать на матобласти. Такое разбиение иногда бывает необходимо для более точного описания взаимодействия тел на выделенной линии скольжения. Методика ТИМ-2D позволяет проводить расчеты в многообластной постановке. При этом в каждой матобласти используется независимая расчетная сетка. В начале каждого счетного шага между матобластями решается задача контактного взаимодействия, в дальнейшем положение и скорости границ выступают в качестве навязанных граничных условий при расчете матобластей. Сами матобласти рассчитываются независимо друг от друга. Это первый ресурс распараллеливания методики ТИМ-2D – по матобластям, который достаточно легко использовать. При этом более удобно использовать модель распределенной памяти, т.к. все взаимодействие между матобластями решается в одном блоке – при расчете контактного взаимодействия и при этом используется небольшой объем информации.

С другой стороны, из-за неструктурированности сетки в методике ТИМ-2D все счетные алгоритмы построены поточно, без опоры на упорядоченность структуры сетки. Благодаря этому каждый элемент сетки можно обчислить независимо от соседей, используя только информацию с предыдущего шага. Это второй ресурс распараллеливания – по элементам сетки внутри матобласти. В зависимости от модели памяти это приводит к двум различным подходам. В модели распределенной памяти это мелкозернистое распараллеливание с простран-

венным разделением области на фрагменты-параобласти (декомпозиция по пространству). При этом недостающая для расчета информация может быть восполнена двумя различными способами – путем наложения между параобластями в один слой ячеек и обменом информацией в слое наложения, либо путем расчета контактного взаимодействия между параобластями по алгоритмам, аналогичным распараллеливанию по матобластям. Если использовать независимость расчета элементов сетки в модели общей памяти, то естественным будет использование распараллеливания итераций счетных циклов (которые, как правило, и построены по элементам сетки). В результате для такого распараллеливания не требуется выполнения декомпозиции по пространству, что позволяет использовать его в сочетании с другими подходами. Таким образом, в методике ТИМ-2D имеется три основных ресурса распараллеливания. Первые два основаны на пространственном разбиении задачи на фрагменты (по матобластям, и внутри матобластей по параобластям) и использовании модели распределенной памяти. И третий – распараллеливание итераций счетных циклов в модели общей памяти. Эти ресурсы и были использованы для разработки распараллеливания методики ТИМ-2D.

На первом этапе было реализовано распараллеливание итераций счетных циклов в модели общей памяти с использованием интерфейса OpenMP [2]. На втором этапе было реализовано распараллеливание по областям в модели распределенной памяти с использованием стандарта MPI, а также режим счета в смешанной модели памяти с использованием двух уровней распараллеливания [3, 4]. Последними были реализованы алгоритмы мелкозернистого распараллеливания [5, 6]. Данный доклад посвящен вопросам одновременного использования уровней распараллеливания при расчете одной задачи.

2. Декомпозиция данных

Эффективное использование счетных программ методик на параллельных машинах требует выполнения декомпозиции таким образом, чтобы ядра вычислительной системы были загружены равномерно, а взаимодействия между ними были минимальны. В модели распределенной памяти задача декомпозиции состоит в распределении данных между процессами (декомпозиция данных) таким образом, чтобы количество обменов и объем передаваемых данных между ними были минимальны. В методике ТИМ-2D для распараллеливания в модели распределенной памяти используется декомпозиция по пространству.

При распараллеливании по областям максимальное количество узлов, которые можно задействовать, равно количеству областей. Разбиение задачи на матобласти выполняется на этапе подготовки расчета, исходя из геометрии задачи и начальных данных. В результирующей сетке может существенно различаться объем вычислений, приходящихся на каждую область, из-за различного количества точек, веществ, используемых приближений и счетных алгоритмов. Это может приводить к снижению эффективности распараллеливания при расчете на максимальном количестве узлов. Для того чтобы обойти эту проблему, в рамках методики ТИМ-2D реализована возможность отнесения нескольких матобластей к одному MPI процессу. То есть одним процессом может рассчитываться не одна матобласть, а несколько. Благодаря этому, как правило, удастся сбалансировать вычислительную нагрузку, при этом уменьшается количество задействованных процессов. В таком виде задача формирования декомпозиции для распараллеливания по областям сводится к задаче близкой к классической задаче «о ранце» теории графов: с каждой областью связывается некоторое число (вес). Этот вес отражает время на расчет области, либо количество счетных точек, если время на расчет области недоступно. Таким образом, получается множество из K чисел, где K – количество матобластей в задаче. Затем решается задача разбиения данного множества на N групп, где N – количество доступных процессов ($N \leq K$). При выполнении разбиений минимизируется максимальная сумма чисел в группах (минимизируется вес самого тяжелого «ранца»). Распределение чисел по группам и определяет распределение областей по процессам, т.е. декомпозицию.

Принципы декомпозиции для мелкозернистого распараллеливания следующие:

- Декомпозиция осуществляется по ячейкам (ячейка – основной счетный элемент сетки в методике ТИМ-2D).
- Все ячейки области распределяются по компактам, при этом каждая ячейка принадлежит одному и только одному компактному. Компакт – набор ячеек одной области.

- Каждая математическая область разбивается на компакты независимо от других.
- На основе каждого компакта формируется параобласть – полноценная счетная область пригодная для проведения расчета (при этом в зависимости от способа мелкозернистого распараллеливания между параобластями может формироваться слой наложения, или параллельная контактная граница).

Задача декомпозиции для мелкозернистого распараллеливания сводится к решению задачи о разрезании графа на подграфы. При этом используется следующий алгоритм:

- На основе неструктурированной сетки строится граф, отображающий ее структуру. При этом ячейкам сетки соответствуют вершины графа, а соседству между ячейками – ребра графа.
- Вершинам графа присваивается вес, отражающий вычислительную нагрузку, связанную с соответствующей ячейкой. Веса ребер графа используются для придания декомпозиции дополнительных свойств – для вытягивания компактов вдоль границ для уменьшения количества обменов при расчете контактного взаимодействия.
- Непосредственно задача о разрезании графа на подграфы решается при помощи алгоритмов библиотек ParMeTiS или SCOTCH [7].
- Для задач, состоящих из нескольких математических областей, используется алгоритм двухпроходной декомпозиции. На первом проходе каждая область разбивается независимо от других на большое количество микрокомпактов. На втором проходе строится макрограф, вершинами которого являются компакты первого прохода и границы между областями. Ребра макрографа отражают соседство между микрокомпактами первого прохода и соседство с границами области.

Подробное описание алгоритмов декомпозиции для методики ТИМ-2D приведено в работе [8].

3. Уровень распараллеливания по областям

На верхнем уровне распараллеливание осуществляется по математическим областям в модели распределенной памяти. Поскольку расчет математических областей полностью не зависит, то взаимодействие между процессами осуществляется только на этапе расчета контактного взаимодействия между ними. При этом сама контактная граница между двумя областями рассчитывается на одном из двух процессов, к которым отнесены области, формирующие эту границу.

Расчет контактного взаимодействия в параллельном режиме производится по следующему алгоритму (курсивом выделены блоки, относящиеся к распараллеливанию):

1. Для каждого контура формируются «свои» стороны (в последовательном режиме все стороны – «свои»).
2. *Производится обмен информацией о сторонах контуров («свои» отправляются, «чужие» принимаются).*
Все последующие операции выполняются только для тех контуров, которые рассчитываются текущим процессом.
3. Совмещение сторон контура.
4. Формирование основного и вспомогательного слоя узлов на контуре.
5. Расчет расстояний между соседними точками на контуре.
6. Посадка узлов вспомогательного слоя на основной слой.
7. *Обмен информацией о сформированных контурах между процессами.*
8. Расчет ускорений и скоростей для граничных узлов «своих» областей.
9. *Получение процессом, рассчитывающим контур вещественной информации о «чужой» стороне контура. (Соответственно эта сторона подготавливается и отправляется процессом, рассчитывающим соседнюю область).*
10. Расчет взаимодействия сторон контура. Вычисление новых координат и скоростей точек контура.
11. *Пересылка рассчитанного контура.*

Заметим, что обмены информацией производятся только для тех контуров, которые разделяют области, рассчитываемые на разных процессах. Если контур разделяет области, рассчиты-

ваемые одним и тем же процессом, то и контур рассчитывается этим же процессом. Аналогично внешний контур задачи (типа «свободная граница» или «жесткая стенка») рассчитываются процессом, рассчитывающим соответствующую область.

4. Уровень мелкозернистого распараллеливания

В методике ТИМ-2D деление на математические области производится исходя из начальных данных, таких как параметры геометрии и распределение веществ. При этом не преследуется цель достижения высокой эффективности распараллеливания. В разных областях может быть задано различное количество точек, используются разные вещества и приближения. Это приводит к тому, что, использовать такое распараллеливание удастся для небольшого количества процессоров. Дополнительное использование распараллеливания в модели общей памяти позволяет увеличить количество задействованных ядер почти на порядок. Тем не менее, в реальных расчетах удастся использовать не более нескольких десятков процессорных ядер. Все это показывает, что такого распараллеливания оказывается недостаточно и необходимо использование мелкозернистого распараллеливания.

Основной вопрос, который возникает при мелкозернистом распараллеливании, это способ расчета узлов сетки, которые окружают ячейки, отнесенные к разным компактам (параграничные узлы). Рассмотрим фрагмент сетки, представленный на рисунке 1. На рисунке ячейки с белой заливкой отнесены к компакт 1, с желтой заливкой к компакт 2.

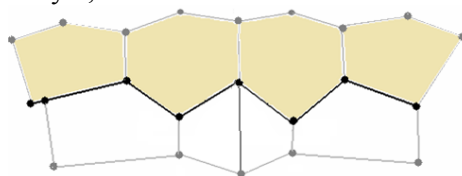


Рис. 1. Фрагмент сетки с разделением на компакты

В соответствии с разностной схемой методики [1] для расчета узла строится замкнутый контур интегрирования из центров ячеек и «центров» ребер («центр» ребра в цилиндрическом случае – точка, которая делит поверхность, образуемую вращением ребра вокруг оси симметрии, на две равные части, в плоском случае – это середина ребра). Пример контура интегрирования для узла V на рисунке 2 выделен зеленой линией. В определении контура интегрирования участвуют центры окружающих ячеек C_1, C_2, C_3, C_4 и «центры» ребер VV_1, VV_2, VV_3, VV_4 . В соответствии с контуром интегрирования при расчете узла V используются величины ячеек C_1, C_2, C_3, C_4 и узлов V, V_1, V_2, V_3, V_4 .

Если сохранять контур интегрирования узла неизменным, то получается первый тип мелкозернистого распараллеливания – с наложением в один слой ячеек.

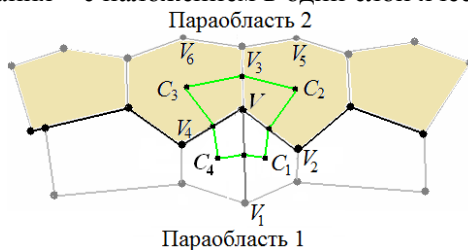


Рис. 2. Контур интегрирования узла

С другой стороны контур интегрирования можно представить как совокупность замкнутых контуров со стороны каждой из ячеек (такой способ используется и в разностной схеме для определения массы узла). Аналогично можно представить контур интегрирования как совокупность замкнутых контуров со стороны каждой параобласти. Для рассматриваемого примера такое разделение представлено на рисунке 3. Линия раздела контура интегрирования на рисунке выделена красным. При разделении контура интегрирования узлы вдоль линии между параобластями разделяются на пары (или больше по количеству параобластей сходящихся в узле),

например, узел V на V' и V'' . Для каждого из узлов используется свой контур интегрирования для определения массы и ускорений, которые затем сочетаются для расчета единой скорости. Такое разделение контура интегрирования при расчете позволяет не использовать наложение между параобластями. Подобная схема используется при расчете контактной границы «без скольжения», поэтому для мелкозернистого распараллеливания был использован блок расчета контактных границ [9] с небольшими доработками.

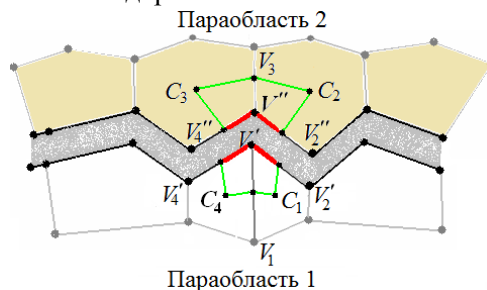


Рис. 3. Разрыв контура интегрирования для узла при мелкозернистом распараллеливании без наложения

В методике ТИМ-2D расчет газодинамических величин разделен на два основных этапа:

1. Расчет узловых величин – скорости, координаты (расчёт уравнения движения). При расчете узла используется информация из соседних узлов и окружающих ячеек с предыдущего момента времени.
2. Расчет ячейечных величин – плотность, давление, энергия (расчёт уравнения энергии). При этом используется информация о новом положении узлов рассматриваемых ячеек.

Первый метод мелкозернистого распараллеливания основан на сохранении контура интегрирования для параграничных узлов. В результате возникает необходимость наложения между параобластями в один слой ячеек. При этом параграничные узлы могут рассчитываться любым из процессов, рассчитывающих окружающие параобласти. Расчет узлов является относительно дешевой операцией в методике ТИМ-2D, поэтому для дополнительного контроля расчет параграничных узлов дублируется. Вторым возникающий вопрос – это обновление информации в присоединенных элементах сетки (ячейки и узлы). Это обновление производится путем асинхронных обменов информацией в параграничных, приграничных и присоединенных элементах. Вызовы процедур выполнения обменов расположены так, чтобы необходимая информация обновлялась до ее использования.

Второй метод мелкозернистого распараллеливания основан на разбиении контура интегрирования параграничных узлов. В результате между параобластями не требуется наложение по ячейкам. При разрыве контура интегрирования в каждой параобласти появляется свой параграничный узел со своим контуром интегрирования. Для восстановления исходного контура интегрирования необходимо их объединение. В этом режиме мелкозернистого распараллеливания для каждого из пары узлов рассчитывается ускорение, производится обмен рассчитанными ускорениями и массами и затем расчет единой скорости. Получение единой скорости является относительно дешевой операцией, поэтому для уменьшения количества обменов получение единой скорости дублируется процессами.

5. Согласование уровней распараллеливания

Отдельной проблемой оказалось согласование алгоритмов трех уровней распараллеливания между собой. Наиболее удобно рассматривать решение этой проблемы с точки зрения мелкозернистого распараллеливания, т.к. с одной стороны это средний уровень, с другой стороны, оно было реализовано последним, и проблема согласования стояла для него наиболее остро.

5.1 Использование смешанной модели памяти

Первый вопрос заключается в использовании смешанной модели памяти и одновременном применении алгоритмов распараллеливания использующих MPI и OpenMP. При распараллели-

вании в модели смешанной памяти в рамках каждого вычислительного узла выполняется один процесс. Процесс выполняет цикл по счетным областям задачи, но области, отнесенные к другим узлам, пропускает. Областью может быть как матобласть, так и параобласть, в зависимости от режима проведения расчета. Нумеруются эти процессы так, как это принято в стандарте MPI (от 0 до $N - 1$, где N – общее количество используемых вычислительных узлов для расчета задачи). Эти процессы и выступают в качестве MPI процессов, они управляют расчетом задачи в рамках вычислительного узла, производят MPI обмены. В рамках каждого вычислительного узла MPI процесс производит независимое от других узлов порождение параллельных нитей OpenMP для проведения расчетов областей, отнесенных к данному вычислительному узлу (и только для этих областей). Таким образом, OpenMP распараллеливание не зависит от распараллеливания в модели распределенной памяти.

5.2 Согласование мелкозернистого и пообластного распараллеливания

Алгоритмы пообластного и мелкозернистого распараллеливания используют модель распределенной памяти и декомпозицию по пространству. В связи с этим возникли вопросы организации описания параобластей в структурах данных и эффект «фрагментации» границ между матобластями по нескольким параобластям.

5.2.1 Организация параобластей в структуре данных

Одним из первых вопросов при разработке алгоритмов мелкозернистого распараллеливания встал вопрос об организации параобластей в структурах данных. Основным требованием при этом являлось то, что эта организация должна быть прозрачной для счетных алгоритмов. Т.е. счетный алгоритм должен иметь возможность одинаково работать как с матобластями, так и параобластями. Для этого параобласти структурно оформляются также как матобласти. Кроме того, в структуре «области» заведен ряд дополнительных величин, описывающих ее как «параобласть» и делающих привязку к «родительской» матобласти. Помимо этого организуется дополнительная структура, содержащая информацию о глобальной нумерации элементов сетки, информацию для обменов между параобластями (параграницах) и другие данные.

Поскольку параобласть в структуре данных описывается также как матобласть, то встает вопрос о согласовании их нумераций. Параобласти в счетных структурах нумеруются в списке матобластей. При этом параобласти получают номера после матобластей. Матобласть, для которой были порождены параобласти, из счета исключается. Параобласть так же, как и матобласть, является глобальным объектом уровня задачи. Благодаря использованию такого подхода возможно использование единых алгоритмов распараллеливания по матобластям для распределения параобластей по процессам и дальнейшего управления. Рассмотрим нумерацию параобластей на примере: в задаче 3 матобласти, из первой области формируется 2 параобласти, из второй области параобласти не формируются совсем, а из третьей формируется 3 параобласти. В результате параобласти, полученные из первой матобласти, получают номера 4, 5, а параобласти из 3 матобласти номера 6, 7, 8. Общее количество областей в структуре данных устанавливается равным 8. Отметим, что исходные матобласти 1 и 3 сохраняют за собой номера, но из счета исключаются. Матобласть 2 рассчитывается под своим номером – параобласть для нее не формируется.

5.2.2 Контактные границы

При мелкозернистом распараллеливании многообластных задач может возникать эффект «фрагментации» границ, когда точки стороны границы распределены по нескольким параобластям. Одним из первых шагов в алгоритме расчета контура – является сборка сторон. В случае «фрагментации» границ этот блок требует дополнительного обмена – пересылки фрагментов границы на управляющий процесс матобласти.

Кроме того, после расчета результирующий контур рассылается всем процессам, имеющим фрагменты сторон данного контура. В остальном алгоритмы расчета фрагментированных границ остаются такими же как и при распараллеливании по областям.

5.3 Взаимодействие мелкозернистого распараллеливания с распараллеливанием итераций счетных циклов

Взаимодействие между мелкозернистым распараллеливанием и распараллеливанием итераций счетных циклов логически организуется достаточно просто, так как в этих режимах используются разные парадигмы. На общей памяти распараллеливаются итерации счетных циклов, такие как циклы по ячейкам или по узлам. В параобласти вводится локальная нумерация элементов сетки, поэтому для распараллеливания в общей памяти параобласть идентична ма-тобласти и модификаций не требуется. Тем не менее, два обстоятельства потребовали специальной доработки алгоритмов.

5.3.1 Разделение счетных циклов

Для мелкозернистого распараллеливания с наложением необходимо выполнение обменов приграничными элементами сетки между соседними параобластями. При этом используются асинхронные обмены для того, чтобы наложить вычисления и обмены. Для того чтобы как можно раньше начать отправки данных о приграничных элементах сетки при формировании параобластей, им присваиваются номера меньше внутренних. В результате, в цикле по ячейкам или узлам они обрабатываются раньше, и можно начать их отправки соседней параобласти. Тут и возникает конфликт. Дело в том, что при распараллеливании счетных циклов нарушается последовательность обработки итераций, т.е. в цикле от 1 до 1000, 101 итерация может быть выполнена раньше 100. Нарушение последовательности итераций происходит из-за их распределения между нитями OpenMP.

Выходом из положения стало разделение счетных циклов на пары. Первый цикл выполняется по параграничным и приграничным элементам сетки. Вторым циклом по внутренним элементам. При этом оба цикла заключены в одну параллельную область OpenMP. Между циклами выполняются обмены. При этом обмены выполняются внутри параллельной области OpenMP также в параллельном режиме.

5.3.2 Обращение к MPI процедурам в параллельных областях OpenMP

Вызов процедур асинхронных обменов производится в параллельных областях OpenMP между двумя счетными циклами по приграничным и внутренним узлам или ячейкам. При этом несколькими нитями (в рамках одного процесса) одновременно могут вызываться MPI процедуры для обмена данными о параобластях с соседями. В некоторых случаях это приводило к автостам в MPI, если на вычислительной системе используется библиотека MPI, не являющаяся «безопасной для нитей» (thread safe). Чтобы избежать одновременного обращения, вызов обменов был модифицирован таким образом, чтобы вызов MPI обменов производила в один момент только одна нить. Для этого все обращения к MPI процедурам заключены в критическую область OpenMP с одинаковым именем MPI. Такой подход решает описанную проблему, практически не снижая скорости выполнения, при условии, что количество обменов невелико. Также его использование гарантирует, что в других блоках подобные проблемы не возникнут.

6. Различные режимы параллельного счета

Три уровня распараллеливания могут использоваться различным образом: по отдельности, в различных попарных сочетаниях, и все три одновременно. Вместе с последовательным режимом возможно 8 разных режимов счета. Выбор конкретного режима проведения расчета определяется задачей, моделируемыми приближениями и характеристиками вычислительной системы. Рассмотрим режимы счета более подробно.

Последовательный режим счета. Этот режим, как правило, используется на персональном компьютере для различных обработок расчетов, а также для методических расчетов. Этот режим часто используется разработчиками при начале разработки алгоритмов, а также для устранения проблем несвязанных с распараллеливанием.

6.1 Режимы счета с одним уровнем распараллеливания

Возможны следующие режимы счета с использованием одного уровня распараллеливания:

Распараллеливание по счетным областям в модели распределенной памяти (крупноблочное распараллеливание). Этот режим может использоваться для расчета небольших задач на малом количестве ядер (обычно менее 10). Ограничение данного режима распараллеливания обусловлено ограниченным количеством матобластей и сложностями в балансировке нагрузки, поскольку в областях используются различное количество точек, материалы, моделируемые приближения, модели и др. Этот метод распараллеливания применяется, когда использование общей памяти невозможно (например, из-за характеристик вычислительной системы).

Распараллеливание по параобластям в модели распределенной памяти (мелкозернистое распараллеливание). В отдельности такой режим может использоваться для расчета однообластных задач (в этом случае уровень управления задачами и областью фактически совмещаются). Из-за малого количества таких задач применяется такой режим редко.

Распараллеливание итераций счетных циклов в модели общей памяти с использованием интерфейса OpenMP. Использование данного режима ограничено тем, что в современных кластерных системах количество ядер, а следовательно и количество нитей, работающих на общей памяти, невелико. По этой причине распараллеливания в модели общей памяти оказывается недостаточно для проведения расчетов с большим количеством точек. Этот режим используется при выполнении некоторых сервисных операций, при проведении расчетов с небольшим количеством точек (не более 100 тысяч ячеек) и для вычислительных систем, использующих только общую память.

Как видно из описания, каждый из этих режимов имеет существенные ограничения по применимости. И каждый из них по отдельности не может быть использован как универсальный подход. Поэтому основные способы распараллеливания – смешанные режимы.

6.2 Режимы счета с использованием двух уровней распараллеливания

Попарное использование методов распараллеливания включает следующие режимы:

Смешанное распараллеливание по областям и распараллеливание итераций циклов при помощи OpenMP. В этом режиме счет задачи осуществляется так же, как в режиме распараллеливания по областям, но на одном вычислительном узле кластера запускается только один MPI процесс. Каждый MPI процесс является главной нитью OpenMP в рамках своего вычислительного узла. Такой режим активно используется при проведении расчетов для задач средних размеров (от нескольких сотен тысяч до миллиона ячеек).

Смешанное мелкозернистое и распараллеливание итераций счетных циклов. Режим работы аналогичен предыдущему, за исключением того, что распараллеливание счета в общей памяти осуществляется внутри параобласти. Этот режим может применяться для больших однообластных задач. Применение такого распараллеливания также ограничено из-за малого количества задач из одной матобласти.

Смешанное по областям и мелкозернистое распараллеливание в модели распределенной памяти. В этом режиме для каждой матобласти определяется управляющий процесс, который отвечает за ее расчет. А сама матобласть разделяется на некоторое количество параобластей. Такой режим распараллеливания может применяться для расчёта многообластных задач в случаях, когда необходимо задействовать большое количество процессов. Этот режим используется для вычислительных систем без общей памяти.

6.3 Режим счета с трехуровневым распараллеливанием

Последний режим параллельного счета – использование всех трех базовых режимов одновременно – *режим трехуровневого распараллеливания.* Этот режим наиболее сложный с точки зрения реализации и управления, но в таком режиме полностью решаются проблемы запуска на произвольном количестве ядер. Поэтому этот режим является основным для решения больших задач из нескольких миллионов точек. Работа в таком режиме аналогична работе в смешанном режиме распараллеливания по областям и мелкозернистому, но на одном вычислительном узле

запускается по одному MPI процессу, как в режиме смешанного распараллеливания по областям и распараллеливание итераций циклов.

Необходимо отметить, что выделение этих 8 режимов счета является отчасти условным. В реальных расчетах могут возникать ситуации, когда, например, для маленькой области не будет использоваться мелкозернистое распараллеливание, если размер области не превышает размера компакта. В случае же совсем маленькой области, состоящей из нескольких сотен ячеек, может отключаться и OpenMP-распараллеливание, т.к. накладные расходы на организацию параллельной области выше выигрыша от расчета этой области в параллельном режиме. Таким образом, хотя глобально на уровне задачи может быть использован режим из всех трех уровней распараллеливания, но на локальном уровне конкретной матобласти могут отключаться уровни мелкозернистого и/или OpenMP-распараллеливания.

7. Исследование эффективности распараллеливания

В качестве характеристик эффективности распараллеливания использовались следующие функции: $S_p = \frac{t_1}{t_p}$ – ускорение счета; $E_p = \frac{t_1}{p \cdot t_p} \cdot 100\%$ – эффективность распараллеливания, где t_1 – время расчёта на одном узле используемой параллельной машины, t_p – время счёта на p узлах.

Для исследования эффективности распараллеливания использовалась задача об обжатии цилиндра в многообластной постановке. Размеры геометрии 10×1 с разбиением на 10 матобластей размером 1×1 . Между матобластями решается задача контактного взаимодействия, на боковой поверхности цилиндра задано постоянное по времени давление $P_{sp} = 5$. Остальные границы являются жесткими стенками. Вещество: Идеальный газ с параметрами: $\gamma = 3$, $\rho_0 = 1$. В каждой области использовалась сетка диаграммы Вороного из 1 миллиона ячеек со случайной расстановкой центров. Фрагмент сетки приведен на рисунке 4. При расчете на одном вычислительном узле использовался только уровень распараллеливания на общей памяти. Это значение использовалось в качестве базового для оценки эффективности распараллеливания. При расчете на 10 узлах использовалось два уровня распараллеливания – «по областям» и OpenMP. При большем количестве узлов каждая матобласть разбивалась на параобласти. При расчете на 20 узлах – каждая матобласть на 2 параобласти, на 40 – на 4, и так далее. Примеры декомпозиций для одной области для различного количества процессов приведены на рис. 5.

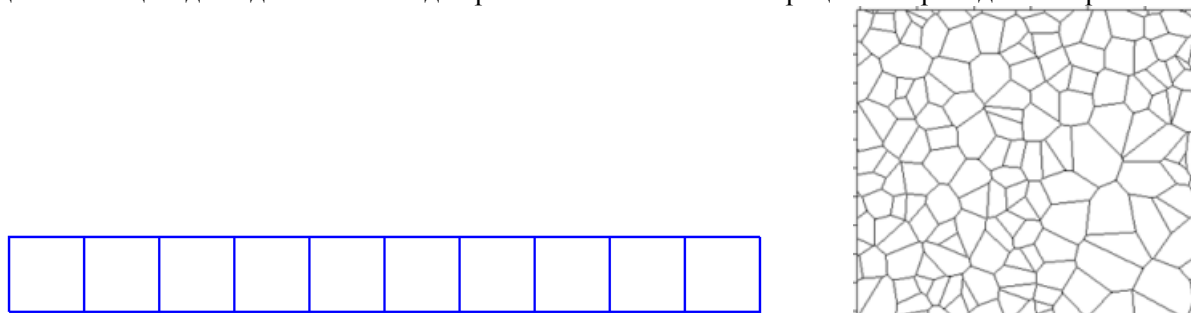


Рис. 4. Вид геометрии с разделением на области и фрагмент начальной сетки диаграммы Вороного

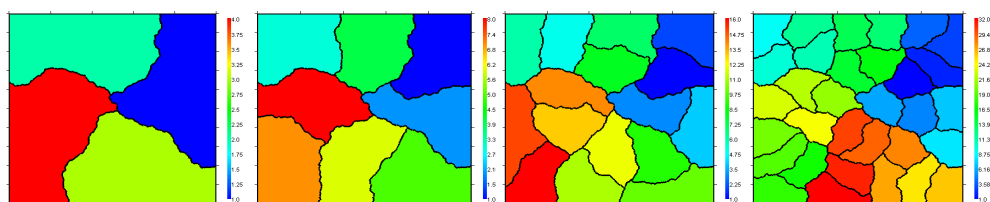


Рис. 5. Вид декомпозиции одной области для количества ядер 40, 80, 160 и 320

В таблице 1 приведены значения ускорения и эффективности распараллеливания. Расчеты проводились в модели смешанной памяти: по одному MPI процессу на каждый узел вычислительной системы, в каждом узле использовалось 8 процессорных ядер. В строке «MPI × OpenMP» первое число указывает количество MPI процессов, а второе – количество OpenMP нитей.

Таблица 1. Мелкозернистое распараллеливание с наложением

Режим	OpenMP	По областям + OpenMP	Трехуровневое распараллеливание			
			160	240	320	400
Количество ядер	8	80	160	240	320	400
MPI × OpenMP	1 × 8	10 × 8	20 × 8	30 × 8	40 × 8	50 × 8
Ускорение, разы	1	8,92	16,08	21,61	23,18	24,5
Эффективность, %	100	89	80	72	58	49

8. Тестовый расчет

В качестве тестового расчета была выбрана газодинамическая задача о полете несферической оболочки [10]. Начальная геометрия задачи представлена на рисунке 6. Начальная форма оболочки $R(t=0, \theta) = 7 + 0.875 \cos^3 \theta - 0.525 \cos \theta$, толщина $\Delta R = 0.16$. Материал – свинец: $\rho_0 = 11.4$. Оболочка имеет скорость, направленную к центру: $v = 5706860$. Давление на наружной поверхности оболочки $P_e = 0$, давление на внутренней поверхности зависит от времени $P_i = 3.495 \cdot 10^6 \cdot (V(t))^{-2/3}$, где $V(t)$ – объём внутренней полости.

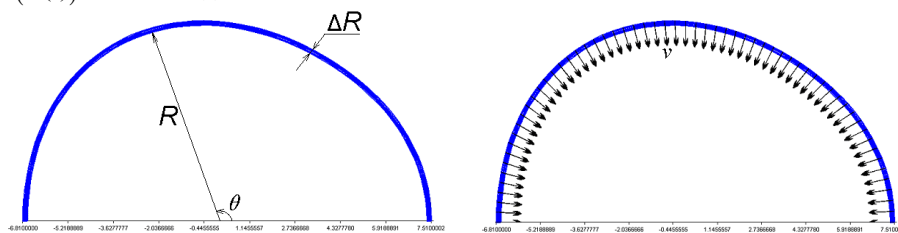


Рис. 6. Начальная форма и скорость задачи о полете несферической оболочки

Результаты расчетов приведены на рисунках 7–8. Во всех расчетах получено полное совпадение результатов между собой. Сравнение формы оболочки в последовательном и параллельном режимах показано на рисунке 7.

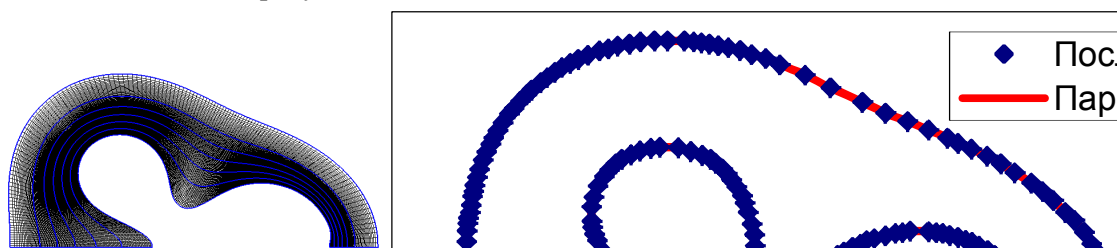


Рис. 7. Вид расчетной сетки на момент времени $t = 1.1$ и сравнение формы оболочки в последовательном и параллельном расчетах

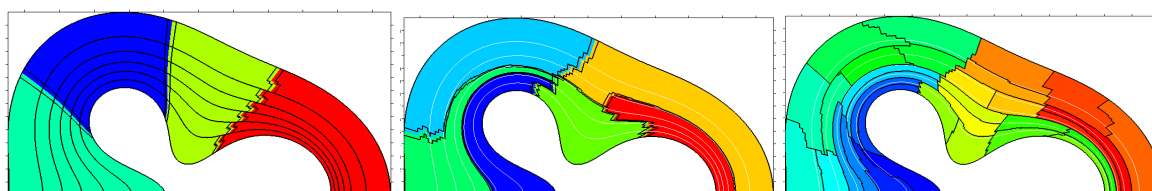


Рис. 8. Примеры результатов расчетов, в различных режимах распараллеливания

9. Заключение

В статье описывается метод трехуровневого распараллеливания, применяемый в методике ТИМ-2D. На первом уровне осуществляется распараллеливание счета по математическим областям. На втором уровне производится распараллеливание счета внутри математической области по параобластям. На первых двух уровнях используется модель распределенной памяти и интерфейс MPI. На третьем уровне осуществляется распараллеливание итераций счетных циклов в модели общей памяти с использованием интерфейса OpenMP. Уровни могут использоваться как по отдельности, так и в различных сочетаниях при решении одной задачи. При этом результаты расчетов в последовательном и параллельном режимах полностью совпадают. Использование метода трехуровневого распараллеливания позволяет гибко настроиться под конфигурацию вычислительной системы и решать задачи с использованием нескольких сотен процессорных ядер с эффективностью не менее 50%.

Литература

1. Соколов С.С., Воропинов А.А., Новиков И.Г., и др. Методика ТИМ-2D для расчета задач механики сплошной среды на нерегулярных многоугольных сетках с произвольным количеством связей в узлах // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2006. Вып. 4. С. 29-43.
2. Воропинов А.А., Новиков И.Г., Соболев И.В., Соколов С.С. Распараллеливание методики «ТИМ» в модели общей памяти с использованием интерфейса OpenMP // Вычислительные методы и программирование. 2007. Том 8. №1. С. 134-141.
3. Воропинов А.А., Соколов С.С., Новиков И.Г. Распараллеливание в модели смешанной памяти для расчета задач газодинамики в методике «ТИМ-2D» // Параллельные вычислительные технологии (ПаВТ'2008): Труды международной научной конференции (Санкт-Петербург, 28 января - 1 февраля 2008 г.). Челябинск: Изд. ЮУрГУ, 2008. С. 69 - 79.
4. Воропинов А.А., Соколов С.С., Новиков И.Г. Двухуровневое распараллеливание в модели смешанной памяти для расчета задач газодинамики в методике ТИМ-2D // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2008. Вып. 1. С. 51-59.
5. Воропинов А.А. Алгоритмы мелкозернистого распараллеливания в методике ТИМ-2D // Вычислительные методы и программирование. 2009. Том 10. С. 112-120.
6. Воропинов А.А., Новиков И.Г., Соколов С.С. Методы мелкозернистого распараллеливания в методике ТИМ-2D // Тезисы докладов XVIII Всероссийской конференции «Теоретические основы и конструирование численных алгоритмов решения задач математической физики», посвященной памяти К.И. Бабенко (Дюрсо, 13-17 сентября, 2010). М.: Институт прикладной математики им. М.В. Келдыша, 2010. С. 19.
7. Половникова Т.Н., Воропинов А.А. Опыт использования библиотек SCOTCH и MeTiS для декомпозиции неструктурированных сеток в методике ТИМ // Супервычисления и математическое моделирование. Труды XII международного семинара / под ред. Р. М. Шагалиев. Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2011. С. 282 - 288.
8. Воропинов А.А. Декомпозиция данных для распараллеливания методики ТИМ-2D и критерии оценки ее качества // Вестник ЮУрГУ, серия «Математическое моделирование и программирование». 2009. №37(170), вып. 4. С. 40-50.
9. Воропинов А.А., Новиков И.Г., Соколов С.С. Расчет контактного взаимодействия между счетными областями в методике ТИМ-2D // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2008. Вып. 2. С. 5-20.
10. Sofronov I.D., Rasskazova V.V., Nesterenko L.V. The use of nonregular nets for solving two-dimensional nonstationary problems in gas dynamics // Numerical Methods in Fluid Dynamics / под ред. N. N. Yanenko, Ju. I. Shokin. Moscow: Mir Publishers, 1984. С. 82 - 121.