УДК 532.546; 533.15

Об одном эффективном конвейерном параллельном алгоритме для решения задач механики сплошной среды

К. И. Михайленко, С. Ф. Хизбуллина

Институт механики УНЦ РАН, Уфа

Аннотация. В работе предлагается техника конвейеризации вычислительного процесса при пространственной декомпозиции расчётной области для построения эффективных параллельных алгоритмов численного решения задач гидродинамики, ориентированных на кластерные вычислительные системы. Показаны методы достижения высокой эффективности параллельного приложения, основанного на конечно-разностной явной или полу-явной численной схеме. Записано выражение для определения мимнимального размера расчётной области, при котором эффективность параллельной программы приближается к единице.

Ключевые слова: параллельный алгоритм, кластерная вычислительная система, пространственная декомпозиция, математическое моделирование, гидродинамика

1 Введение

Задачи вычислительной гидродинамики представляют собой особый класс задач, решение которых с желаемой точностью может потребовать значительных вычислительных ресурсов. Это связано с тем, что любое увеличение точности используемой математической модели, например, более подробная пространственная дискретизация, более адекватное описание физических, химических и иных параметров исследуемых процессов, приводит к резкому ужесточению требований, предъявляемых к используемой вычислительной технике. Именно поэтому задачи вычислительной гидродинамики стоят на первом месте в списке так называемых «задач Большого Вызова» [1], содержащем перечень научных и научно-технических

проблем, решение которых требует использования высокопроизводительных вычислительных систем.

Термином «высокопроизводительная» принято называть вычислительную систему, производительность и ресурсы которой значительно превышают соответствующие средние показатели для вычислительной техники на текущий момент. Простейший путь повышения производительности системы заключается в дублировании ее функциональных устройств (процессоров, блоков оперативной и дисковой памяти и т.д.), поэтому многопроцессорность является характерной особенностью современных высокопроизводительных вычислительных систем.

В настоящее время существует несколько направлений развития высокопроизводительной вычислительной техники, из которых наиболее популярно создание кластерных вычислительных систем (вычислительных кластеров). Основная отличительная особенность вычислительного кластера — объединение нескольких одно- или двухпроцессорных компьютеров (узлов кластера) с помощью стандартной коммуникационной среды, что, с одной стороны, значительно снижает стоимость построения вычислительной системы, но с другой стороны, позволяет использовать лишь весьма небольшие скорости обмена данными между отдельными узлами кластера по сравнению с полноценным суперкомпьютером. На практике это приводит к существенному ужесточению требований на используемый для решения задачи численный алгоритм, который должен в этом случае обеспечивать минимальный обмен данными между различными процессорами системы. Кроме того, эффективность алгоритма во многом будет определяться тем, насколько равномерно и полно он обеспечивает загрузку всех процессоров системы [2, 3].

Следует отметить, что хотя большинство вычислительных алгоритмов, используемых в вычислительной гидродинамике, обладают так называемым внутренним параллелизмом [4], не каждый из них может быть распараллелен для использования на многопроцессорных вычислительных системах. Более того, класс алгоритмов, допускающих эффективное распараллеливание для кластерных вычислительных систем еще более узок. Тем не менее, для большинства практических задач гидродинамики удается построить численные алгоритмы, достаточно эффективно работающие на кластерных вычислительных системах.

В статье рассматривается техника построения эффективной схемы распараллеливания алгоритмов решения явных и полу-явных конечно-разностных численных схем.

2 Распараллеливание численных алгоритмов

Одним из наиболее распространенных в настоящее время подходов к построению параллельных алгоритмов является подход, основанный на пространственной декомпозиции расчетной области (domain decomposition method). Суть его достаточно проста и наглядна: расчетная область разбивается на отдельные подобласти, число которых согласовано с количеством процессоров вычислительной системы. Далее расчет всех подобластей проводится по одному и тому же алгоритму. При этом для расчета каждой отдельной подобласти отводится свой процессор.

Несмотря на очевидную простоту приведенного принципа, его практическое применение зависит от конкретных особенностей распараллеливаемого алгоритма.

Многие численные методы решения задач гидродинамики основаны на методе конечных разностей. Хорошо известно, что свойства численного алгоритма во многом определяются видом используемой в нем конечно-разностной схемы. При решении задач на обычных однопроцессорных вычислительных системах предпочтение обычно отдается алгоритмам, основанным на неявных конечно-разностных схемах. Это объясняется, прежде всего, устойчивостью таких схем, что позволяет производить вычисления с произвольными шагами дискретизации по временной и пространственным переменным. Кроме того, аппроксимация производных по пространству в неявной конечно-разностной схеме оказывается более адекватна физике моделируемого процесса.

При использовании неявных конечно-разностных алгоритмов решение задачи сводится, в конечном итоге, к решению системы линейных алгебраических уравнений, размерность которой зависит от количества узлов конечно-разностной сетки. Однако при распараллеливании подобные алгоритмы приводят к весьма значительным обменам данными между отдельными узлами системы, что делает их практически непригодными для использования на кластерных вычислительных системах. Поэтому параллельные алгоритмы, основанные на неявных численных схемах, применяются, в основном, на суперкомпьютерах, особенно на машинах с общей памятью.

Алгоритмы, построенные на основе явных конечно-разностных схем, допускают достаточно простое распараллеливание, применимое и для кластерных систем. Однако основным недостатком явных схем является их условная устойчивость, что значительно снижает эффективность соответствующих алгоритмов.

Улучшить описанную ситуацию позволяет переход на алгоритмы, основанные на полу-явных конечно-разностных схемах [5]. Полу-явность в данном случае означает, что для расчета некоторой искомой величины u

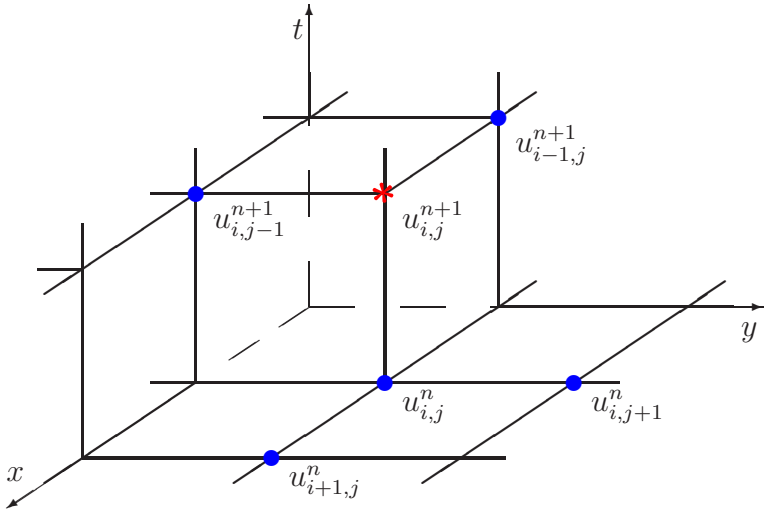


Рис. 1. Схема расположения узловых точек, используемых для расчета нового значения

в любой внутренней точке расчетной области используются значения этой величины в соседних узлах сетки не только на предыдущем временном слое, но и уже вычисленные значения этой величины на текущем:

$$u^{n+1}_{i,j} = f(u^{n}_{i,j}, u^{n}_{i+1,j}, u^{n}_{i,j+1}, u^{n+1}_{i-1,j}, u^{n+1}_{i,j-1}, \Delta x, \Delta y)\,. \qquad (1)$$

Как видно из Рис. 1, в результате счёта с использованием организованного по описанному принципу полу-явного численного метода возникает «бегущая ступенька», осложняющая процесс распараллеливания.

Возможные способы распараллеливания явных и полу-явных алгоритмов различаются, прежде всего, схемами декомпозиции расчетной области. Далее приводится анализ нескольких таких возможных схем.

3 Распараллеливание явных и полу-явных численных схем

В основу распараллеливания алгоритма положена идея геометрического параллелизма, которая заключается в декомпозиции исходной геометрической области на ряд подобластей, количество которых зависит от числа процессоров используемой для расчета вычислительной системы. Для упрощения рассуждений будем предполагать, что каждый вычислитель-

ный процесс параллельного алгоритма выполняется на отдельном процессоре, то есть понятия «процесс» и «процессор» будут синонимичны.

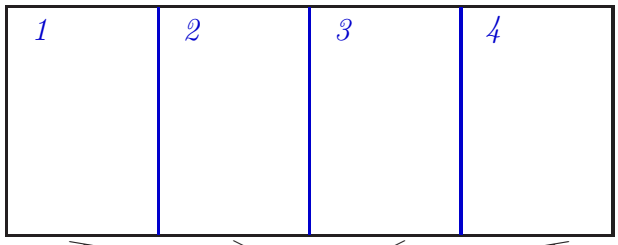
3.1 Простая декомпозиция

Изначально простая декомпозиция, отвечающая требованию минимального объема пересылок, предусматривает разбиение расчетной области только в направлении одной из координатных осей, как это показано на Рис. 2(a). На Рис. 2(b, c) представлена пространственно-временная диаграмма, описывающая активность процессоров и их взаимодействие для случаев использования явной и описанной выше полу-явной численных схем. Здесь горизонтальные полоски демонстрируют время, затрачиваемое каждым процессором на вычисление своей подобласти. Обмен сообщениями между процессорами обозначен стрелками.

Типичная схема распараллеливания для явной конечно-разностной схемы выглядит следующим образом. Вся расчетная область разбивается на число частей, равных количеству процессоров, отведенных для вычислений. Все части имеют максимально близкие размеры. При этом конечно-разностные сетки граничащих частей перекрываются на два слоя (так называемые «теневые грани»). В силу явности расчетной схемы на каждом временном шаге расчет каждой подобласти производится независимо и параллельно. При этом вычисления проводятся только во внутренних узлах сетки. После завершения расчета на временном слое результаты с приграничных («теневых») слоев сетки пересылаются в соответствующие граничные слои соседних областей. Значения на границах всей расчетной области находятся из краевых условий. После этого расчет повторяется на новом временном слое.

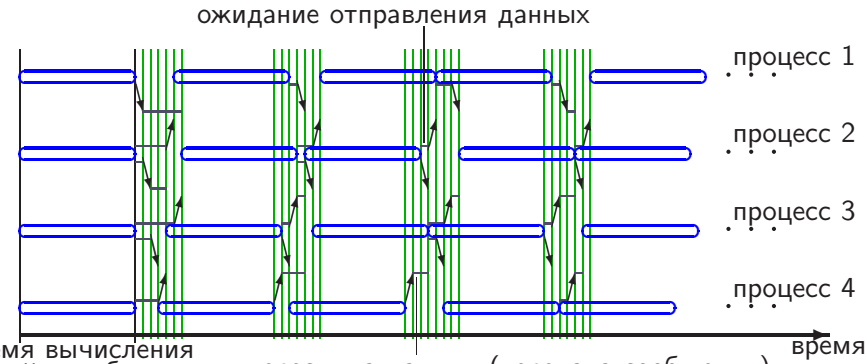
Рис. 2(b) демонстрирует наличие неизбежных пауз в работе процессоров. Указанные паузы возникают из-за ограничений, накладываемых средой передачи данных вычислительной системы. В вычислительном кластере такие ограничения приводят к заметным затратам времени на передачу отдельного сообщения, а также не позволяют организовать одновременную передачу всех требуемых сообщений. Как результат, работа процессоров многократно прерывается для передачи и получения необходимых для дальнейших вычислений данных, а времена простоев оказываются тем бóльшими, чем чаще и чем бóльшего объёма данными необходимо обмениваться в процессе вычислений.

Последовательность вычислений для полу-явной конечно-разностной схемы можно описать следующим образом. Первый процессор рассчитывает свою подобласть на первом шаге по времени, после чего передает необходимые для продолжения расчета результаты («бегущую ступеньку») второму процессору. После получения указанных результатов, второй

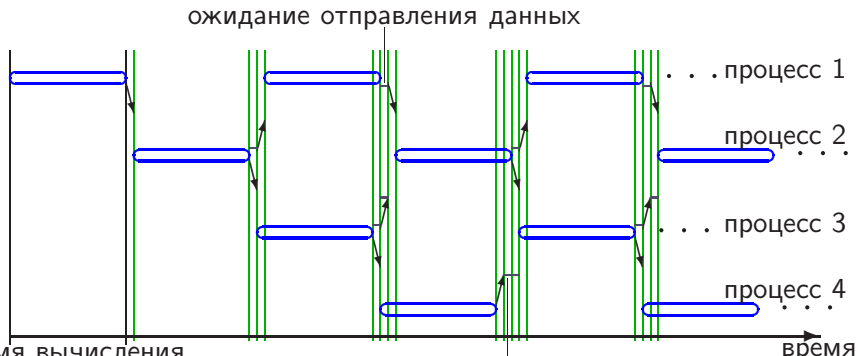


a.

подобласти



b. (один шаг по времени)



c. (один шаг по времени)

Рис. 2. Расчетная область (a) и пространственно-временная диаграмма для случая простой декомпозиции по топологии «линейка» для явной (b) и полу-явной численных схем (c)

процессор начинает вычисления для своей подобласти. В это время первый процессор ожидает результаты вычислений второй подобласти от второго процессора, так как они необходимы ему для продолжения расчетов на втором временном слое.

Для остальных процессоров реализуется аналогичная схема работы.

Из Рис. 2(c) видно, что в каждый момент времени работает только половина процессоров, что объясняется полу-явностью используемого численного алгоритма и принятой схемой декомпозиции. В этом случае, если каждый процессор на любом шаге по времени обсчитывает свою подобласть целиком, полу-явность алгоритма приводит к чередующейся работе процессоров, то есть в каждый момент времени половина процессоров будет простаивать.

3.2 Бинарная декомпозиция подобласти

Улучшить загрузку процессоров позволяет такая организация вычислительного процесса, когда подобласть рассчитывается не целиком, а по частям, с промежуточной пересылкой данных.

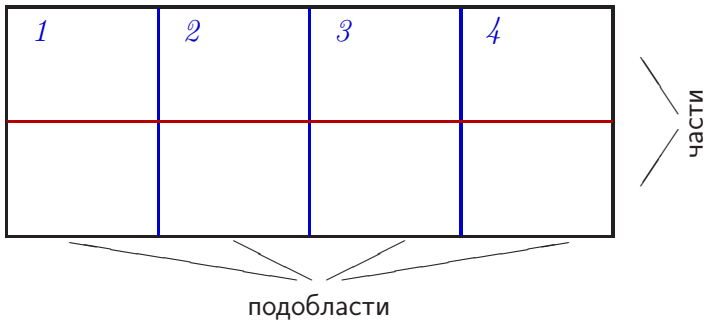
Рассмотрим случай разбиения подобласти на две части, как это показано на Рис. 3(a). Пространственно-временная диаграмма, иллюстрирующая вычислительный процесс в этом случае, изображена на Рис. 3(b, c). Заштрихованные горизонтальные полоски на диаграмме описывают процесс вычисления процессором первой части своей подобласти, а не заштрихованные — вычисление второй части.

На Рис. 3(b) видно, что если время расчета одной части подобласти будет меньше суммарного времени всех пересылок, то для непрерывной работы всех процессоров в этом случае количество частей, на которые разбивается подобласть, должно быть не менее двух.

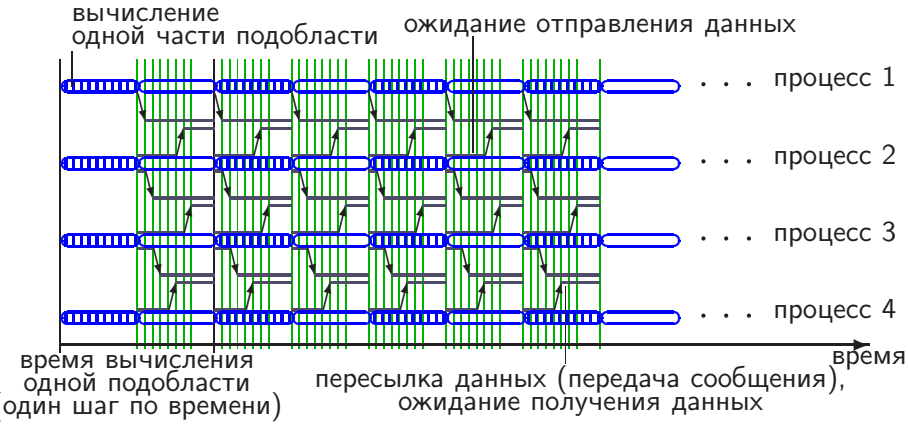
Действительно, после расчёта первой части своей подобласти каждый процесс отправляет информацию со своих теневых граней соседям и немедленно приступает к вычислению второй части подобласти. По окончании второго этапа снова пересылаются теневые грани и, если уже получена информация о теневых гранях для первой части подобласти, необходимая для продолжения вычислений, процесс повторяется на следующем временно́м слое.

Из приведенной диаграммы на Рис. 3(c) можно видеть, что в случае дополнительного разбиения подобласти на две части для полу-явной конечно-разностной схемы время простоя процессора значительно меньше по сравнению с ранее описанным случаем простой декомпозиции (Рис. 2(c)). Для объяснения этого факта рассмотрим ход вычислительного процесса.

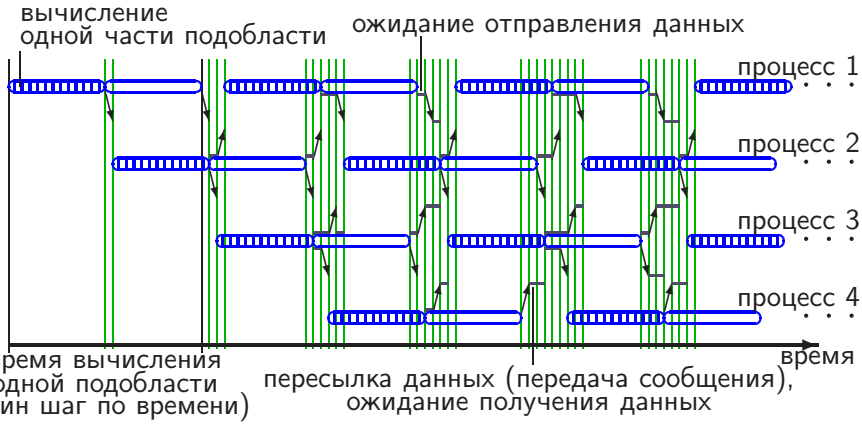
После окончания расчета первой части первой подобласти первый процессор отсылает необходимые результаты второму. В следующий момент



a.



b.



c.

Рис. 3. Расчетная область (a) и пространственно-временная диаграмма для случая декомпозиции по топологии «линейка» с дополнительным разбиением подобластей на две части для явной (b) и полу-явной (c) численных схем

времени работают два процессора одновременно. Первый процессор продолжает расчет второй части своей подобласти, а второй — начинает расчет первой части своей.

Для продолжения расчета на втором временном слое первый процессор нуждается в результатах расчета только первой части второй подобласти. Благодаря этому и удается сократить время простоя процессора. Более того, время простоя оказывается равным времени передачи всех сообщений и в идеальном случае мгновенной передачи данных оно равно нулю.

Конечно, любая реальная кластерная система имеет отличное от нуля вполне определенное время передачи сообщения, в связи с чем представленная схема декомпозиции оказывается недостаточно эффективной.

3.3 Множественная декомпозиция подобласти

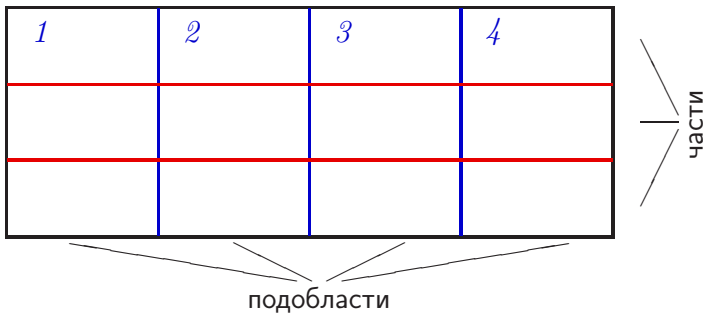
Для непрерывной работы всех процессоров при использовании полу-явной численной схемы количество частей, на которые разбивается подобласть, должно быть не менее трех [6]. Диаграмма работы параллельной программы при таком разбиении представлена на Рис. 4(b). Здесь полосками, заштрихованными вертикальными линиями, показано время расчета первой части подобласти, заштрихованными крестиком — второй и не заштрихованными — третьей части.

Как продемонстрировано на диаграмме Рис. 4(b) дополнительное разбиение каждой расчетной подобласти не менее чем на три части позволяет подобрать условия, приводящие к полной загрузке участвующих в расчете узлов кластерной вычислительной системы.

Теперь можно записать последовательность расчета. При этом надо учесть, что необходимость сохранения больших объемов промежуточных результатов расчета (на различных, но далеко не всех временных шагах) требует выделения одного процесса для сбора данных со всех процессов и сохранения их на диске. Для определенности будем считать этот выделенный процесс нулевым. Такая процедура позволяет освободить остальные процессы от выполнения операции обмена данными с диском и улучшить тем самым равномерность их загрузки вычислительной работой. Все процедуры обмена данными с периферийными устройствами осуществляются через нулевой процесс, именно он осуществляет первоначальную рассылку данных, содержащих требуемую для них информацию о свойствах среды, начальных и граничных условиях и прочих исходных данных по остальным процессам.

- Нулевой процесс считывает с диска исходные данные и распределяет их по рабочим процессам. При этом можно производить действия над большими массивами данных, не помещающимися целиком в оперативной памяти, отведенной данному процессу.

a.



b.

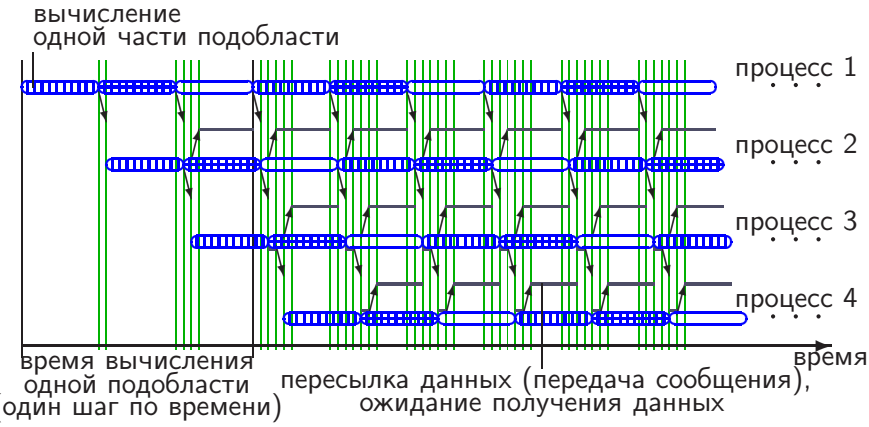


Рис. 4. Расчетная область (a) и пространственно-временная диаграмма (b) для случая декомпозиции по топологии «линейка» с дополнительным разбиением подобластей на три части для полу-явной численной схемы

- Первый процесс производит расчет первой части своей подобласти на первом временном слое. В это время остальные процессы системы находятся в состоянии ожидания. По окончании расчета первый процесс отсылает требуемые результаты второму.

- На следующем этапе работают уже два процесса: первый процесс рассчитывает вторую часть своей подобласти, а второй — первую часть своей. По окончании расчета первый процесс пересылает второму необходимые ему для расчета второй части подобласти данные. Второй процесс передает третьему данные, необходимые для расчета первой части третьей подобласти. Кроме того, второй процесс передает первому данные, необходимые для расчета первой части первой подобласти на втором временном шаге.

- Подобная последовательность действий продолжается и на следующих этапах, в том числе и после вступления в работу всех процессоров системы.

- Если результаты расчета некоторого временного шага должны быть сохранены, то каждый процесс по окончании расчета всей подобласти на указанном шаге передает результаты нулевому процессу для сохранения их на диске.

4 Оценка эффективности параллельного алгоритма

Размеры пространственной подобласти и количество частей могут быть выбраны исходя из того, что время расчета подобласти должно быть не меньше суммарного времени всех пересылок данных. Последнее условие предполагает, что в любой момент времени может производиться только одна пересылка.

Формально указанное условие может быть записано в виде неравенства, связывающего время расчета одной части подобласти t_p и время передачи одного сообщения t_s:

$$t_p \geq 2(p-1)t_s + \frac{pt_\Delta}{n}\,, \tag{2}$$

где p — число процессоров, участвующих в вычислениях; n — количество частей на которые разбивается подобласть; t_Δ — добавка, учитывающая периодическую передачу результатов расчета нулевому процессу для сохранения в файле.

При выполнении условия (2) все обмены данными между процессорами оказываются разнесенными во времени и не будут пересекаться. Эффективность вычислительного процесса можно повысить, если пересылки данных осуществлять посредством неблокирующих функций прием-передачи.

Оценим входящие в выражение (2) параметры.

Время вычисления одной части подобласти может быть оценено по формуле

$$t_p = \frac{IJm}{\nu}\,. \tag{3}$$

Здесь I и J — размеры части подобласти в точках (узлах вычислительной сетки); m — количество операций, требующееся для расчета всех параметров в одной точке на одном временном слое (для случая системы уравнений Навье—Стокса в двумерной области эта величина может быть оценена в $m \approx 10^2$); ν — производительность процессора в Mflops.

Время пересылки оценивается как

$$t_s = t_\ell + \frac{Jk}{u}\,, \tag{4}$$

где t_ℓ — время латентности среды передачи данных; k — размер числа в байтах; u — скорость передачи данных по сети.

Добавка, учитывающая время периодической передачи результатов расчета нулевому процессу для сохранения на диске

$$t_\Delta = \frac{1}{\mu}\left(\frac{IJnk}{u} + t_\ell\right)\,. \tag{5}$$

Здесь μ — частота пересылки промежуточных результатов.

Пусть $I = J$ для упрощения оценки. В этом случае на основе приведенных уравнений записывается квадратное уравнение относительно I, которое имеет положительный вещественный корень только при выполнении условия

$$\left[(p-1)\frac{k}{u}\right]^2 \geq \left(\frac{pk}{\mu u} - \frac{m}{\nu}\right)\left[2(p-1)t_\ell + \frac{pt_\ell}{n\mu}\right]\,. \tag{6}$$

На практике данное условие можно упростить:

$$\frac{pk}{\mu u} \leq \frac{m}{\nu}\,. \tag{7}$$

Приведённое условие не является слишком жёстким даже для кластерных вычислительных систем, использующих в качестве среды передачи данных сеть Ethernet. Так, для 12-узлового вычислительного кластера на базе процессоров Alpha, установленного в Башкирском региональном центре высокопроизводительных вычислений, и для плоской задачи о течении несжимаемой вязкой жидкости оценка требуемых размеров вычислительной области даёт величину $I \geq 100$.

Данный вывод подтверждается проведёнными вычислительными экспериментами. В частности, с сипользованием описанной параллельной техники решены тестовые задачи о нестационарном распределении температуры в плоской пластине и о течении несжимаемой вязкой жидкости в плоском канале.

5 Заключение

Таким образом, в работе исследована возможность параллельной реализации различных алгоритмов решения задач механики сплошной среды в

зависимости от используемого вида конечно-разностных схем. Показано что алгоритмы, базирующиеся на полу-явных численных схемах, которые объединяют некоторые достоинства как явных, так и неявных схем, могут быть эффективно распараллелены для использования на кластерных вычислительных системах. Кроме того, приводится соотношение, позволяющее оценить минимальные необходимые для эффективного вычислительного процесса размеры расчетной области.

Список литературы

[1] Grand Challenges 1993: High performance computing and communications // A report by the Committee on Physical, Mathematical and Engineering Sciences. — The FY 1993. U.S. Research and Development program. — 75 p.

[2] Foster I. Designing and Building Parallel Programs. — Addison-Wesley, 1995. — 430 p.

[3] Воеводин В. В. Математические модели и методы в параллельных процессах. — М.: Наука, 1986. — 296 с.

[4] Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. — 608 с.

[5] Griebel M., Dornseifer T., Neunhoeffer T. Numerical Simulation in Fluid Dynamics. — SIAM, 1998. — 217 p.

[6] Gazizov R. K., Khizbullina S. F., Lukashuk S. Yu., Mikhaylenko C. I. Numerical Solving of Fluid Dynamics Equations on Cluster Computing Systems: a Technique Using Domain Decomposition [Электронный ресурс] // Proc. of the 4th International Workshop on Computer Science and Information Technologies, CSIT'2002. — Patras, Greece. 2002. — 6 p. (CD-ROM)