Уфа: УГАТУ, 2003 Вестник УГАТУ Т. 4, № 1. С. 100–107

УДК 519.6; 532

Р. К. ГАЗИЗОВ, С. Ю. ЛУКАЩУК, К. И. МИХАЙЛЕНКО

РАЗРАБОТКА ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧ МЕХАНИКИ СПЛОШНОЙ СРЕДЫ НА ОСНОВЕ ПРИНЦИПА ПРОСТРАНСТВЕННОЙ ДЕКОМПОЗИЦИИ

Обсуждаются некоторые особенности использования принципа пространственной декомпозиции расчетной области для построения параллельных алгоритмов численного решения задач механики сплошной среды, ориентированных на кластерные вычислительные системы. *Параллельный алгоритм; кластерная вычислительная система; пространственная декомпозиция; математическое моделирование; механика сплошной среды*

ВВЕДЕНИЕ

Задачи вычислительной механики сплошной среды относятся к тому классу задач, которые стимулируют развитие как эффективных численных алгоритмов, так и вычислительной техники. Чем более адекватна используемая математическая модель реальному процессу, чем более детально описываются геометрические, физические, химические, иные свойства моделируемого объекта, тем больший объем вычислений требуется для получения конечного результата и тем большими ресурсами и производительностью должна обладать используемая вычислительная система. Поэтому неслучайно задачи механики сплошной среды занимают верхние позиции в списке так называемых «задач Большого Вызова» [1], содержащем перечень научных и научно-технических проблем, решение которых требует использования высокопроизводительных вычислительных систем.

Под термином «высокопроизводительная» понимается такая вычислительная система, производительность и ресурсы которой значительно превышают соответствующие средние показатели для вычислительной техники на текущий момент времени. 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 большинства современных высокопроизводительных вычислительных систем.

Наиболее простым и относительно недорогим способом создания высокопроизводительной вычислительной техники является объединение нескольких одно- или двухпроцессорных компьютеров единой коммуникационной средой. Именно по такому принципу строятся компьютерные системы, называемые вычислительными кластерами, популярность которых в научном мире возрастает с каждым годом. От полноценного суперкомпьютера кластерную систему отличает значительно меньшая скорость обмена данными между отдельными узлами системы. 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 далеко не каждый последовательный численный алгоритм может быть распараллелен, а класс алгоритмов, допускающих эффективное распараллеливание для кластерных систем, оказывается еще у́же. Тем не менее для большинства практических задач механики сплошных сред удается построить численные алгоритмы, достаточно эффективно работающие на кластерных вычислительных системах. Данная работа посвящена некоторым вопросам построения таких алгоритмов.

1. 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 [4]). Суть его достаточно проста: 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 во многом определяется конкретными особенностями распараллеливаемого алгоритма.

Большинство численных методов решения задач механики сплошной среды 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 адекватной 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. Типичная схема распараллеливания для двухмерной расчетной области 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. 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 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.

Основным недостатком явных схем является их условная устойчивость, что значительно снижает эффективность соответствующих алгоритмов. В самом деле, пусть кластерная система из 10 процессоров используется с целью проведения расчетов на более мелкой сетке. Тогда уменьшение среднего размера сетки в 10 раз для большинства явных алгоритмов механики сплошной среды приведет к необходимости уменьшения шага по времени в 100 раз. В результате даже в пренебрежении временем, затрачиваемым на передачу данных, мы получаем увеличение в 100 раз общего времени расчета по сравнению с однопроцессорным вариантом расчета на грубой сетке. Справедливости ради необходимо отметить, что расчет на мелкой сетке на однопроцессорной машине в этом случае, скорее всего, просто невозможен. Однако эффективность такого параллельного алгоритма оказывается не очень высокой.

Улучшить описанную ситуацию позволяет переход на алгоритмы, основанные на полуявных конечно-разностных схемах [5]. Полуявность в данном случае означает, что для расчета некоторой искомой величины u 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 слое (рис. 1):

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

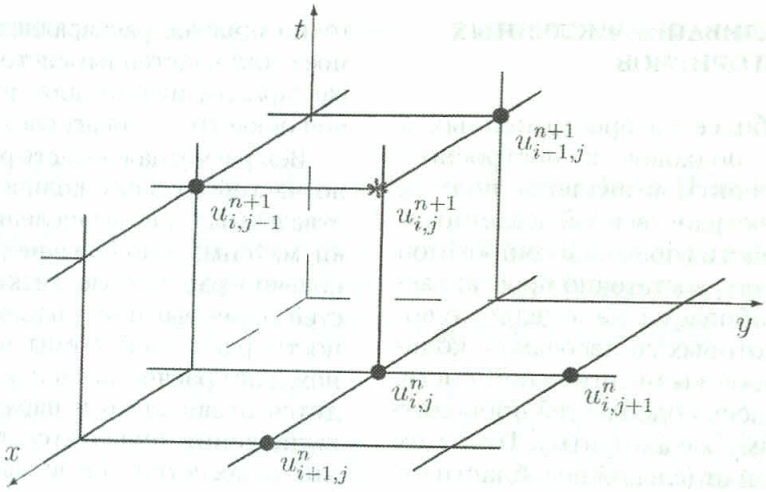

Рис. 1. Схема расположения используемых для расчета нового значения узловых точек

Возможные способы распараллеливания полуявного алгоритма различаются, прежде всего, схемами декомпозиции расчетной области. Далее приводится анализ нескольких таких возможных схем.

2. РАСПАРАЛЛЕЛИВАНИЕ ПОЛУЯВНОЙ ЧИСЛЕННОЙ СХЕМЫ

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. Будем предполагать, что каждый вычислительный процесс параллельного алгоритма выполняется на отдельном процессоре.

Преимуществом выбранного алгоритма является то, что он допускает реализацию в виде ряда подпрограмм, которые могут выполняться и над подобластями. Пространственную декомпозицию облегчает также тот факт, что расчетная область всегда является прямоугольником (для трехмерных областей — параллелепипедом), описанным вокруг исследуемого объекта.

2.1. 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*. Однако, если каждый процессор на любом шаге по времени обсчитывает свою подобласть целиком, полуявность алгоритма приводит к чередующейся работе процессоров, т. е. в каждый момент времени половина процессоров будет простаивать.

На рис. 2, *b* представлена пространственно-временная диаграмма, описывающая активность процессов и их взаимодействие. Здесь горизонтальные полоски демонстрируют время, затрачиваемое каждым процессом на вычисление своей области. Обмен сообщениями между процессами обозначен стрелками. Хорошо видно, что в каждый момент времени работает только половина процессоров, что объясняется полуявностью используемого численного алгоритма и принятой схемой декомпозиции.

Действительно, последовательность вычислений в рассматриваемой области можно описать следующим образом. Первый процесс расчитывает свою подобласть на первом шаге по времени, после чего передает необходимые для продолжения расчета результаты второму процессу. После получения указанных результатов, второй процесс начинает вычисления для своей подобласти. В это время первый процесс ожидает результаты вычислений второй подобласти от второго процесса, так как они необходимы ему для продолжения расчетов на втором временном слое.

Для остальных процессов реализуется аналогичная схема работы.

2.2. Бинарная декомпозиция подобласти

Улучшить загрузку процессоров позволяет такая организация вычислительного про-

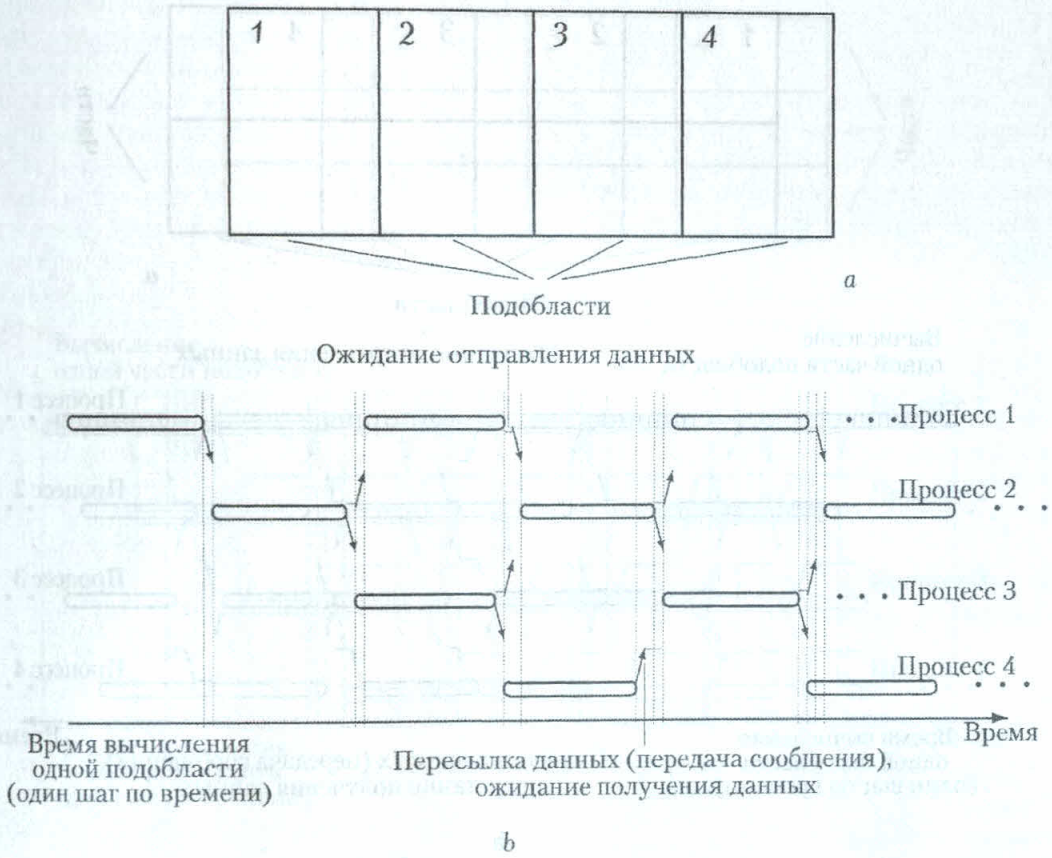

Рис. 2. Расчетная область (*a*) и пространственно-временная диаграмма (*b*) для случая простой декомпозиции по топологии «линейка»

цесса, 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*. Заштрихованные горизонтальные полоски на диаграмме описывают процесс вычисления процессом первой части своей подобласти, а незаштрихованные — вычисление второй части.

Из приведенной диаграммы хорошо видно, что в случае дополнительного разбиения подобласти на две части время простоя процессора значительно меньше по сравнению с ранее описанным случаем (рис. 2, *b*). Для объяснения этого факта рассмотрим ход вычислительного процесса.

После окончания расчета первой части первой подобласти первый процесс отсылает необходимые результаты второму. В следующий момент времени работают два процесса одновременно. Первый процесс продолжает расчет второй части своей подобласти, а второй — начинает расчет первой части своей.

Для продолжения расчета на втором временном слое первый процесс 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 сообщения 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.

2.3. Множественная декомпозиция подобласти

Для непрерывной работы всех процессов количество частей, на которые разбивается подобласть, должно быть не менее трех [6].

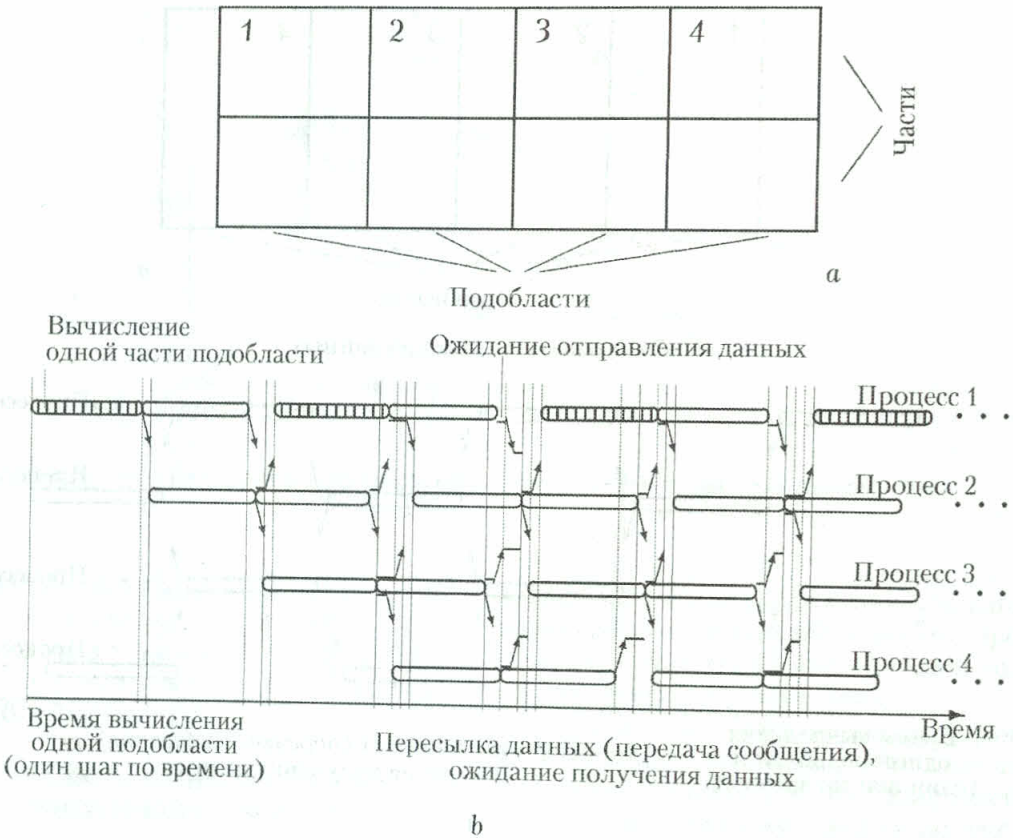

Рис. 3. Расчетная область (*a*) и пространственно-временная диаграмма (*b*) для случая декомпозиции по топологии «линейка» с дополнительным разбиением подобластей на две части

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 незаштрихованными — третьей части.

Как хорошо видно из диаграммы на 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.
- Первый процесс производит расчет первой части своей подобласти на первом временном слое. В это время остальные процессы системы находятся в состоя-

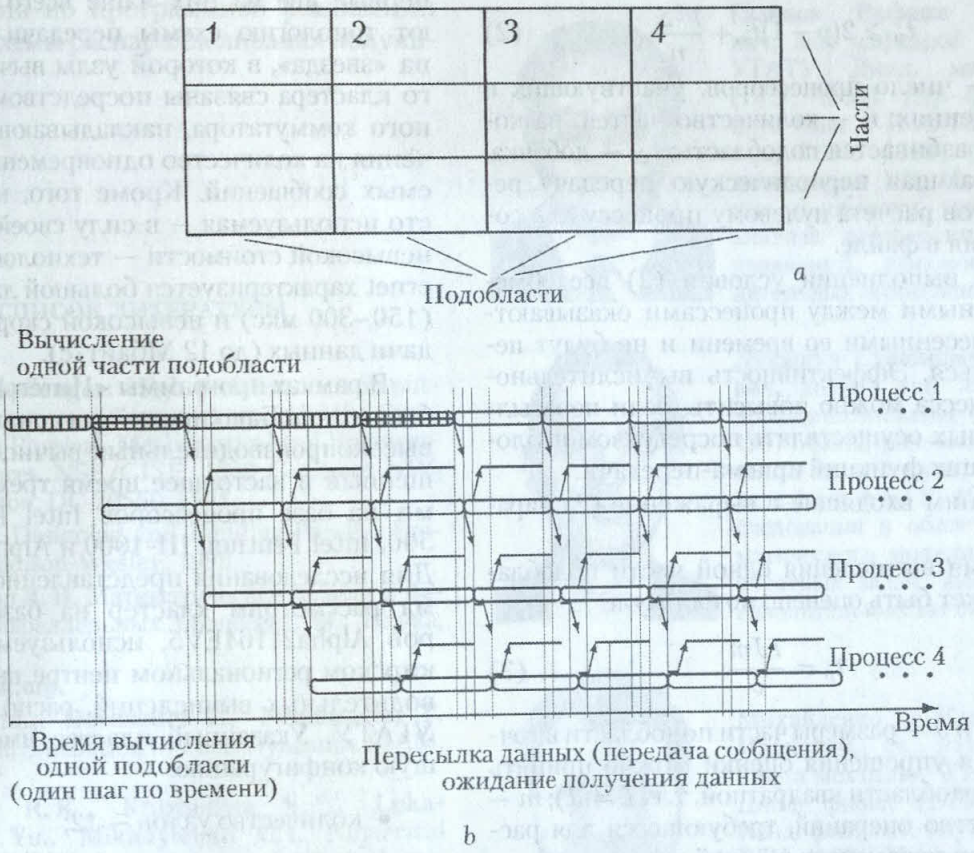

Рис. 4. Расчетная область (*a*) и пространственно-временная диаграмма (*b*) для случая декомпозиции по топологии «линейка» с дополнительным разбиением подобластей на три части

нии ожидания. По окончании расчета первый процесс отсылает требуемые результаты второму процессу.

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

3. ОЦЕНКИ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА

3.1. Теоретическая оценка

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:

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

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 процессами оказываются разнесенными во времени и не будут пересекаться. Эффективность вычислительного процесса можно повысить, если пересылки данных осуществлять посредством неблокирующих функций приема-передачи.

Оценим входящие в выражение (2) параметры.

Время вычисления одной части подобласти может быть оценено по формуле

$$t_p = \frac{IJm}{\nu}. \qquad (3)$$

Здесь I и J — размеры части подобласти в точках (для упрощения оценки можно принять часть подобласти квадратной, т. е. $I = J$); m — количество операций, требующееся для расчета всех параметров в одной точке на одном временном слое (для случая системы уравнений Навье—Стокса в двухмерной области эта величина может быть оценена в $m \approx 10^2$); ν — производительность процессора в Mflops.

Время пересылки оценивается как

$$t_s = t_\ell + \frac{Jk}{u}, \qquad (4)$$

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:

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

Здесь μ — частота пересылки промежуточных результатов.

3.2. Численная оценка

Эффективность представленного параллельного алгоритма существенно зависит от архитектуры вычислительной кластерной системы. Подавляющее большинство имеющихся в России кластеров в качестве коммуникационной среды используют технологии FastEthernet, Myrinet и SCI. Отметим, что первые две из них чаще всего предполагают топологию схемы передачи данных типа «звезда», в которой узлы вычислительного кластера связаны посредством единственного коммутатора, накладывающего ограничения на количество одновременно передаваемых сообщений. Кроме того, наиболее часто используемая — в силу своей простоты и невысокой стоимости — технология FastEthernet характеризуется большой латентностью (150–300 мкс) и невысокой скоростью передачи данных (до 12 Мбайт/с).

В рамках программы «Интеграция» в Уфе был создан Башкирский региональный центр высокопроизводительных вычислений, оснащенный в настоящее время тремя кластерами на базе процессоров Intel Pentium III-500, Intel Pentium III-1000 и Alpha21164EV5. Для исследования представленного алгоритма рассмотрим кластер на базе процессоров Alpha21164EV5, используемый в Башкирском региональном центре высокопроизводительных вычислений, расположенном в УГАТУ. Указанный кластер имеет следующую конфигурацию:

- количество узлов — 12;
- процессор на узле — Alpha21164EV5 с тактовой частотой 533 МГц;
- память на узле — 128 Мбайт;
- коммуникационная среда — FastEthernet;
- коммутатор - HP ProCurve 1600M.

Оценка минимального размера части подобласти, обеспечивающей непрерывную работу всех процессоров указанного кластера, дает нижнюю границу для величины $I \approx 100$. В соответствии с приведенной оценкой, минимальный размер расчетной области, необходимый для эффективной загрузки всех процессоров указанного кластера, равен 300×1100 узловых точек.

ЗАКЛЮЧЕНИЕ

В работе дан анализ возможности параллельной реализации различных алгоритмов решения задач механики сплошной среды в зависимости от используемого вида конечно-разностных схем. Показано что алгоритмы, базирующиеся на полуявных численных схемах, которые объединяют некоторые достоинства как явных, так и неявных схем, могут быть эффективно распараллелены для использования на кластерных вычислительных

системах. В настоящее время проводится активная работа по программной реализации описанной схемы распараллеливания полуявной схемы.

СПИСОК ЛИТЕРАТУРЫ

1. **Grand Challenges:** High performance computing and communications // A report by the Committee on Physical, Mathematical and Engineering Sciences. NSF/CISE, 1800 G. Street NW, Washington, DC 20550, 1991.
2. **Foster I.** Designing and Building Parallel Programs. Addison-Wesley, 1995.
3. **Воеводин В. В.** Математические модели и методы в параллельных процессах. М.: Наука, 1986.
4. **www.ddm.org.**
5. **Griebel M., Dornseifer T., Neunhoeffer T.** Numerical Simulation in Fluid Dynamics. SIAM, 1998.
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 // Proc. of the 4th Int. Workshop on Computer Science and Information Technologies, CSIT'2002. Patras, Greece. 2002.

ОБ АВТОРАХ

Газизов Рафаил Кавыевич, зав. кафедрой ВВТиС УГАТУ. Дипл. математик (БГУ, 1983). Д-р физ.-мат. наук (защ. в ИММ Уральск. отд. РАН, 1999). Заслуж. деят. науки РБ. Исследования в области группового анализа дифференциальных уравнений, высокопроизводительных вычислений.

Лукащук Станислав Юрьевич, доцент той же кафедры. Дипл. инженер (УГАТУ, 1997). Канд. физ.-мат. наук по тепло- и молекулярной физике (защ. в БГУ, 1999). Исследования в области математического моделирования, обратных задач, высокопроизводительных вычислений.

Михайленко Константин Иванович, ст. науч. сотр. Ин-та механики УНЦ РАН. Дипл. физик (БГУ, 1992). Канд. физ.-мат. наук (защ. в БГУ, 1999). Исследования в области мат. моделирования, динамики многофазных систем, высокопроизводительных вычислений.

Разное

СЛОВО О НАУКЕ И ОБРАЗОВАНИИ

Многознание не научает быть умным . . . *Гераклит.*
Разум растет у людей в соответствии с мирапознанием. *Эмпедокл.*
Главным свойством учителя должна быть щедрость. *П. Л. Капица.*
Единственный источник научного знания есть опыт. *К. Ф. Рулье.*
Лучше изучить лишнее, чем ничего не изучить. *Сенека.*
Думаю, что все сколько-нибудь ценное, чему я научился, приобретено мною путем самообразования. *Чарльз Дарвин.*
Чем больше у меня работы, тем больше я учусь. *Фарадей.*
Я учился, творя . . . *К. Э. Циолковский.*
Моя единственная сила — это мое упорство. *Пастер.*
Чем крупнее достижения ученого, тем короче и точнее их можно описать. *П. Л. Капица.*

[Слово о науке: Афоризмы. Изречения. Литературные цитаты. Кн. 2. М.: Знание, 1981.]