УДК 519.6

ИСПОЛЬЗОВАНИЕ ИНТЕРФЕЙСА OpenMP ДЛЯ РАСПАРАЛЛЕЛИВАНИЯ МЕТОДИКИ ТИМ

А. А. Воропинов, И. Г. Новиков, С. С. Соколов
(РФЯЦ-ВНИИЭФ)

Описывается распараллеливание методики ТИМ в модели общей памяти с использованием интерфейса OpenMP. Распараллеливание осуществлено путем добавления директив OpenMP для каждого цикла, итерации которого могут выполняться независимо друг от друга. Реализованные алгоритмы проверены на ряде расчетов.

Введение

Методика ТИМ предназначена для решения нестационарных многомерных задач механики сплошной среды на лагранжевых сетках произвольной структуры. По данной методике могут рассчитываться двумерные задачи (методика ТИМ-2D [1]) в цилиндрической и декартовой системах координат и трехмерные задачи (методика ТИМ-3D [2]) в декартовой системе координат. В настоящее время в методике ТИМ реализованы приближения для решения следующих классов задач: газовой динамики, нестационарной упругопластичности, магнитной гидродинамики, теплопроводности, двухтемпературности и многопотоковой газодинамики.

Для повышения точности расчетов и более адекватного численного моделирования сложных задач необходимо использовать сетки с большим количеством ячеек (счетных точек). Расчеты с большим количеством точек требуют значительного календарного времени. Один из путей сокращения этого времени — выполнение расчетов в параллельном режиме счета.

Наиболее распространенными для задач механики сплошной среды в настоящее время являются распараллеливание в модели распределенной памяти (чаще всего используется интерфейс передачи сообщений MPI [3]) и распараллеливание в модели общей памяти (интерфейс OpenMP [4]). Более универсальным считается MPI, однако для методик, использующих нерегулярные сетки, реализация распараллеливания с его помощью имеет ряд трудностей. Поэтому было решено первоначально осуществить распараллеливание методики ТИМ в модели общей памяти с использованием интерфейса OpenMP.

В настоящее время как для двумерной методики ТИМ-2D, так и для трехмерной методики ТИМ-3D, несмотря на сложность, активно ведутся работы в направлении распараллеливания в модели распределенной памяти. В дальнейшем для вычислительных систем, использующих смешанную систему памяти, будет реализован режим смешанного распараллеливания. В этом случае интерфейс OpenMP будет применяться на нижнем уровне для распараллеливания счета на одном вычислительном узле при расчете фрагмента задачи, отнесенного к этому вычислительному узлу, в рамках MPI-распараллеливания.

Принципы распараллеливания в модели общей памяти

Сформулируем основные принципы распараллеливания методики ТИМ в модели общей памяти, которых придерживались авторы:

1. *Распараллеливание итераций счетных циклов.* Такое распараллеливание наиболее просто реализуется в модели общей памяти и позволяет использовать OpenMP совместно с распараллеливанием в модели распределенной памяти.
2. *Независимое распараллеливание каждого счетного блока.* Это означает, что каждый счетный блок распараллеливается независимо от другого, а фрагменты программы между этими блоками выполняются последовательно. При этом внутри каждого счет-

ного блока количество параллельных областей OpenMP должно быть по возможности минимизировано. Такое распараллеливание выполняется несколько проще, чем *сквозное* распараллеливание всей программы, и в случае небольшого количества счетных блоков и небольших последовательных участков оказывается вполне приемлемым.

Тем не менее отказ от принципа независимого распараллеливания каждого счетного блока может быть одним из первых шагов для устранения последовательных участков и повышения эффективности распараллеливания. В этом случае создается одна параллельная область OpenMP, которая включает все счетные блоки. Таким образом, все счетные блоки распараллеливаются *насквозь*. Заметим, что сквозное распараллеливание достаточно трудоемко с точки зрения программной реализации: необходимо выявить точки синхронизации и предотвратить возможные конфликты между счетными блоками.

3. *Использование высокопараллельных алгоритмов.* При разработке методики ТИМ был использован ряд алгоритмов, которые не являются оптимальными в последовательном режиме, однако в модели общей памяти значительно лучше распараллеливаются. В результате при увеличении количества процессоров такие алгоритмы выполняются быстрее, чем "хорошие" последовательные алгоритмы. То есть используются алгоритмы, максимально "быстрые" в параллельном режиме счета.
4. *Использование профилирования.* Перед выполнением распараллеливания оценивался временной вклад каждого счетного блока в общее время счета. Соответственно распараллеливание осуществлялось сначала для самых дорогостоящих по времени счета блоков. Такой подход позволяет на первоначальном этапе выполнить распараллеливание участков программы, на которые приходится более 95 % счетной нагрузки.
5. *Минимизация дополнительной памяти.* В процессе анализа кодов программ, при определении параллельных областей, было решено не описывать "большие" массивы (ячеечные, узловые, для описания структуры сетки) как частные для каждой нити (класс PRIVATE). Такие массивы описывались как общие (класс SHARED). Это позволило сэкономить значительные объемы памяти, требуемые для организации параллельных областей во время счета, так как при увеличении числа процессоров объем затрачиваемой памяти возрастал бы прямо пропорционально их числу.

Разделение методики ТИМ на счетные блоки

На начальном этапе работ по распараллеливанию методики ТИМ были получены оценки времени работы каждого счетного блока, отвечающего за те или иные вычисления.

В табл. 1 приведены доли времени исполнения отдельных блоков.

Как видно, относительное время выполнения последовательных участков, т. е. тех, которые не были распараллелены, для двумерного и трехмерного случаев практически одинаково и составляет примерно 1,5—2 % от общего времени расчета. Стоит отметить, что в данной таблице представлены замеры для первой тестовой задачи, которая будет рассмотрена ниже.

Исследование эффективности распараллеливания

Тестирование распараллеленных программ методики ТИМ в модели общей памяти с использованием интерфейса OpenMP было проведено на модельных задачах газодинамики и упругопластичности. Расчеты выполнялись в параллельном и последовательном режимах на параллельных вычислительных системах, у которых на одном вычислительном узле имеется несколько процессоров, работающих на общей памяти. Получено совпадение по всем контролируемым величинам: числу счетных шагов, временному шагу, суммарной кинетической, внутренней, полной энергии и ряду других величин.

В качестве характеристик эффективности распараллеливания использовались следующие функции:

$S_p = \dfrac{t_1}{t_p}$ — ускорение счета; $E_p = \dfrac{t_1}{p\,t_p} \times$ $\times\ 100\,\%$ — эффективность распараллеливания, где t_1 — время расчета на одном процессоре используемой параллельной вычислительной системы, t_p — время счета на p процессорах;

Таблица 1

Процентное соотношение времени выполнения отдельных счетных блоков для задачи о сильном взрыве

Рассчитываемый процесс	Доля от полного времени исполнения (в скобках — доля распараллеленного участка счетного блока, %)	
	ТИМ-2D	ТИМ-3D
Счет уравнения движения	33,2 (100)	35,1 (100)
Счет уравнения энергии	39,8 (100)	36,9 (100)
Коррекция сетки	20,7 (95)	20,3 (94)
Локальные перестройки	6,1 (96)	7,2 (95)
Прочие	0,2 (0)	0,5 (0)
Итоговый последовательный участок	1,479	2,078

$E_{\max} = \dfrac{1}{1+d(p-1)} \cdot 100\%$ — максимальная эффективность распараллеливания, которую можно достичь на p процессорах при доле d времени выполнения нераспараллеленного участка программы от общего времени ее исполнения в последовательном режиме (максимально достижимая эффективность).

Для исследования эффективности были использованы две тестовые задачи: *о сильном взрыве* и *о плоской волне.*

Методическая задача о сильном взрыве. Данная задача широко известна как задача о взрыве Седова [5].

Рассматривается сферически-симметричное движение газа, возникающее в результате точечного взрыва в однородном веществе без противодавления. Начальная геометрия задается шаром радиусом $R = 0{,}05$ с центром в начале координат.

Во всей области находится идеальный газ с уравнением состояния $P = (\gamma - 1)\rho E$, $\gamma = 1{,}4$. Начальная плотность газа равна 1.

На всех внешних границах установлено граничное условие типа *жесткой стенки.*

Расчет задачи о сильном взрыве выполнялся на нерегулярной сетке в однообластной постановке. Число ячеек (счетных точек) в области составляло 120 000. Внутренняя энергия в центральной ячейке в начальный момент времени задавалась равной 10^5, в остальных ячейках — равной нулю. Данная задача использует все счетные блоки из табл. 1.

В табл. 2, 3 представлены замеры времени счета задачи с числом ячеек 120 000 на различном числе процессоров как по методике ТИМ-2D, так и по методике ТИМ-3D.

На рис. 1,*а,б* приведены графики эффективности распараллеливания в зависимости от числа процессоров в целом по задаче в сравнении с максимально достижимой эффективностью при наличии последовательного участка ($E_{\max}$).

Методическая задача о плоской волне. Задача описывает распространение плоской ударной волны по области, заполненной газом.

Начальная геометрия для трехмерного расчета задается прямоугольным параллелепипедом (высота — 5, в основании — квадрат с ребром 1), для двумерного расчета — цилиндром (радиус — 1, высота — 5).

Во всей области находится идеальный газ с уравнением состояния $P = (\gamma - 1)\rho E$, $\gamma = 3$. Начальная плотность газа равна 1, внутренняя энергия равна нулю. Все поверхности области представляют собой жесткие стенки, кроме одного из оснований, которое является свободной границей. На свободной границе области задается постоянное давление, равное 5.

Расчет задачи о плоской волне в газодинамической постановке выполнялся на шестигранной (в двумерном случае на четырехугольной) счетной сетке. Число ячеек в области варьировалось и составляло для разных расчетов 52 тыс., 105 тыс., 210 тыс., 420 тыс., 840 тыс., 1 260 тыс.

В данной задаче движение вещества происходит вдоль одного направления, совпадающего с линиями сетки, поэтому искажения счетной сетки не происходит и блоки коррекции и перестройки не вносят вклада в счетную нагрузку. Для таких расчетов доля последовательного (нераспараллеленного) участка программы сводится к минимуму. Для методики ТИМ-2D она составила менее 0,0002 %, а для ТИМ-3D — менее 0,0001 %.

Таблица 2

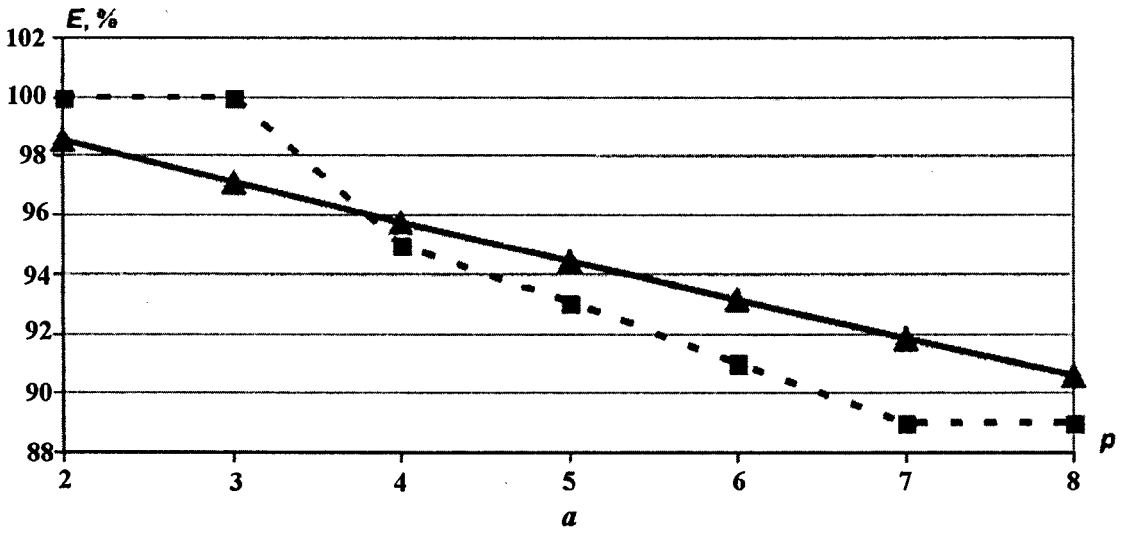
Задача о сильном взрыве. Ускорение счета (в число раз) на различном числе процессоров

Методика	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$
ТИМ-2D	2,01	3,00	3,80	4,69	5,44	6,21	7,14
ТИМ-3D	1,99	2,92	3,73	4,55	5,27	6,08	6,79

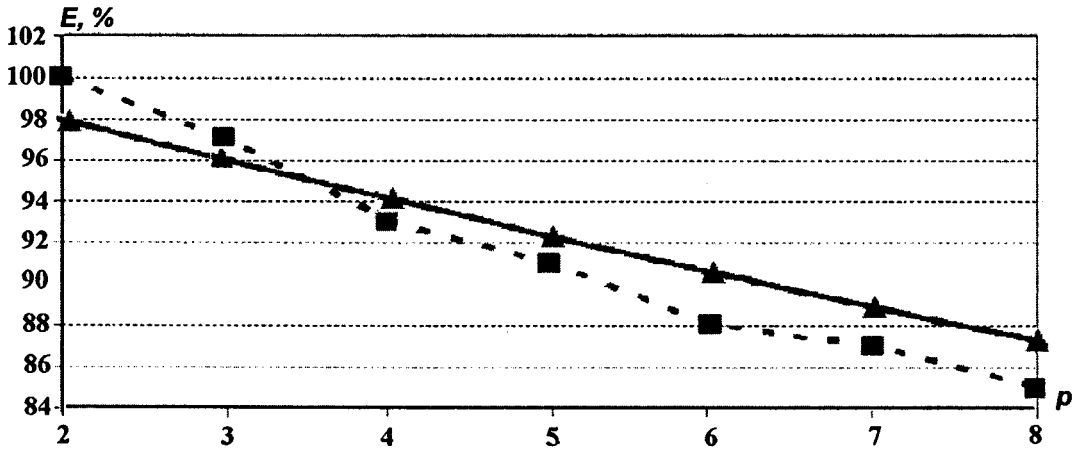
Таблица 3

Задача о сильном взрыве. Эффективность распараллеливания (в %) на различном числе процессоров

Методика	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$
ТИМ-2D	100	100	95	93	91	89	89
ТИМ-3D	100	97	93	91	88	87	85



а



б

Рис. 1. Задача о сильном взрыве. Зависимость эффективности распараллеливания счета от числа процессоров: *а* — для методики ТИМ-2D; *б* — для методики ТИМ-3D; - ■ - — расчет по методике; –▲– — $E_{\max}$

В табл. 4 приведены замеры времени исполнения отдельных блоков в процентном соотношении ко времени счета всей задачи.

В табл. 5, 6 представлены ускорение и эффективность распараллеливания при расчете задачи с различным числом счетных ячеек на различ-

Таблица 4

Процентное соотношение времени выполнения отдельных счетных блоков для задачи о плоской волне

Рассчитываемый процесс	Доля от полного времени исполнения (в скобках — доля распараллеленного участка счетного блока, %)	
	ТИМ-2D	ТИМ-3D
Счет уравнения движения	41,4 (100)	53,6 (100)
Счет уравнения энергии	58,6 (100)	46,4 (100)
Итоговый последовательный участок	< 0,0002	< 0,0001

Таблица 5

Задача о плоской волне. Ускорение счета (в число раз) на различном числе процессоров

Число ячеек	Методика	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$
52 тыс.	ТИМ-2D	1,92	2,82	3,66	4,32	5,01	5,66	6,27
	ТИМ-3D	1,97	2,69	3,56	4,40	5,19	6,08	6,79
105 тыс.	ТИМ-2D	1,92	2,61	3,31	4,35	5,10	5,80	6,43
	ТИМ-3D	1,95	2,81	3,71	4,57	5,49	6,19	7,16
210 тыс.	ТИМ-2D	1,94	2,82	3,51	4,28	4,97	5,74	6,30
	ТИМ-3D	1,97	2,90	3,83	4,73	5,65	6,61	7,44
420 тыс.	ТИМ-2D	1,94	2,81	3,58	4,32	4,91	5,73	6,29
	ТИМ-3D	1,97	2,94	3,86	4,79	5,75	6,64	7,57
840 тыс.	ТИМ-3D	1,97	2,94	3,88	4,84	5,79	6,73	7,64
1 260 тыс.	ТИМ-3D	1,96	2,96	3,90	4,78	5,78	6,72	7,61

Таблица 6

Задача о плоской волне. Эффективность распараллеливания (в %) на различном числе процессоров

Число ячеек	Методика	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$
52 тыс.	ТИМ-2D	96,1	94,0	91,4	86,4	83,6	80,9	78,3
	ТИМ-3D	98,4	89,6	89,0	88,0	86,5	86,9	84,8
105 тыс.	ТИМ-2D	96,1	87,0	82,7	86,9	85,0	82,9	80,4
	ТИМ-3D	97,3	93,7	92,7	91,3	91,5	88,5	89,5
210 тыс.	ТИМ-2D	97,2	94,0	87,7	85,6	82,9	82,1	78,8
	ТИМ-3D	98,6	96,7	95,9	94,6	94,1	94,4	93,0
420 тыс.	ТИМ-2D	97,0	93,8	89,5	86,3	81,8	81,8	78,6
	ТИМ-3D	98,6	97,9	96,6	95,9	95,8	94,9	94,6
840 тыс.	ТИМ-3D	98,7	98,1	97,1	96,7	96,6	96,2	95,4
1 260 тыс.	ТИМ-3D	98,0	98,7	97,5	95,6	96,3	96,0	95,2

ном числе процессоров как по методике ТИМ-2D, так и по методике ТИМ-3D. На рис. 2 приведены графики эффективности распараллеливания в зависимости от числа процессоров для расчетов на различном количестве точек. Исходя из данных, представленных в таблицах, можно сделать вывод, что для методики ТИМ-2D ускорение не зависит от количества точек. Для методики ТИМ-3D с увеличением количества точек наблюдается постепенное увели-

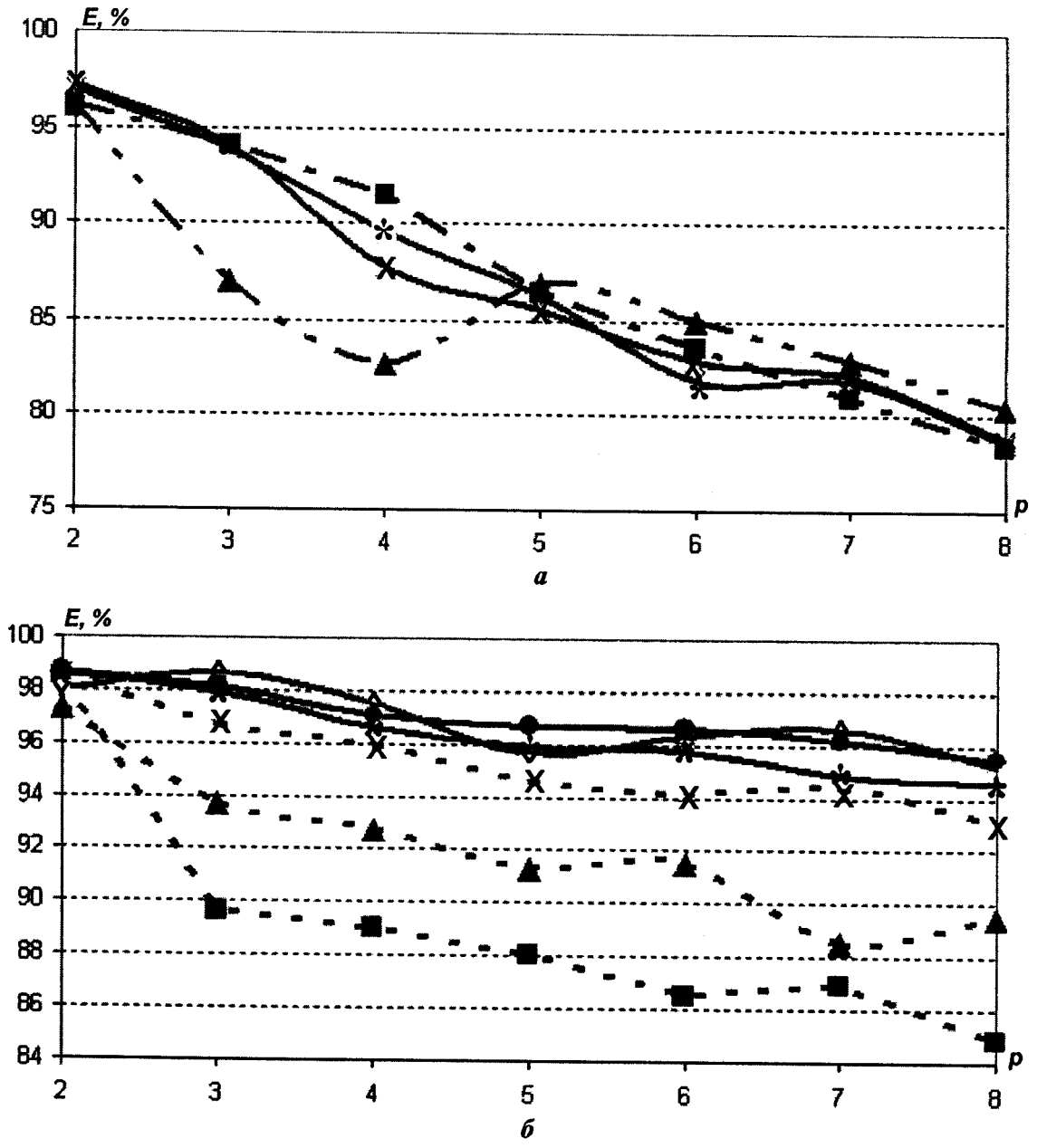


Рис. 2. Задача о плоской волне. Зависимости эффективности распараллеливания от числа процессоров для различного количества точек: *а* — при расчете по методике ТИМ-2D; *б* — при расчете по методике ТИМ-3D; - ■ - — 52 тыс.; - ▲ - — 105 тыс.; - × - — 210 тыс.; -∗- — 420 тыс.; -●- — 840 тыс.; -◇- — 1 260 тыс.

чение эффективности, которая стабилизируется при числе ячеек 840 тыс.

Как видно из таблиц, эффективность распараллеливания двумерной методики оказалась ниже, чем эффективность распараллеливания трехмерной методики. Это связано, во-первых, со значительными различиями в расчетах уравнения энергии и движения по той и другой методикам. Так, для методики ТИМ-3D наибольший объем вычислений приходится на расчет уравнения движения (из-за значительно большего количества узлов по сравнению с ячейками), а в методике ТИМ-2D расчет уравнения движения является менее дорогостоящей операцией. Во-вторых, циклы с небольшим объемом вычислений (*малые* циклы) хуже поддаются распараллеливанию из-за возрастающего отношения накладных расходов к полезной нагрузке. В двумерной методике при расчете уравнения движения имеется два малых цикла (для узлов, лежащих на границах), которые распараллеливаются с эффективностью менее 50 %. Для повыше-

ния эффективности распараллеливания необходимо пересмотреть их с алгоритмической точки зрения. Отметим, что в трехмерной методике подобные циклы распараллеливаются с эффективностью не менее 80 %, это связано с большим количеством граничных узлов.

Анализируя графики зависимости эффективности распараллеливания, можно сделать вывод, что для задачи о плоской волне получены хорошие результаты по ускорению счета при расчетах по трехмерной методике ТИМ-3D на большом количестве точек. При счете по двумерной методике ТИМ-2D на данной задаче результаты хуже, чем на первой тестовой задаче. Это связано, по всей видимости, с типом расчетной сетки. В данной задаче количество узлов примерно равно количеству ячеек сетки, а в предыдущей задаче их было примерно в 2 раза больше. Эффективность распараллеливания на шестиугольной сетке получилась выше ввиду большей вычислительной нагрузки, приходящейся на расчет одной счетной точки (вычислительная нагрузка выше, а количество синхронизаций примерно одинаково).

Некоторые особенности распараллеливания в модели общей памяти

Общеизвестно, что разработка параллельных программ связана с дополнительными трудностями, не присущими реализации последовательных программ. Это сложность отладки, наличие дополнительных накладных расходов на синхронизацию и обмен данными и ряд других проблем. В этой связи необходимо отметить ряд особенностей распараллеливания в модели общей памяти с использованием интерфейса OpenMP. Данные замечания следуют из опыта, полученного авторами при распараллеливании методик ДМК [6], ТИМ-2D и ТИМ-3D.

Сложности отладки программ. Эта проблема присуща также распараллеливанию в модели распределенной памяти, однако при распараллеливании с использованием интерфейса OpenMP она является более острой. При таком распараллеливании большая часть данных — общая для процессоров, и, оказывается, очень сложно выяснить, какой именно процессор и по каким причинам занес неправильные данные в ту или иную ячейку памяти.

Как правило, решать такую проблему приходится не за счет средств отладки, а за счет детального анализа алгоритмов (текстов программ, используемых переменных и т. д.).

Различия во времени первоначального распараллеливания и его окончательной доработки. Несомненным преимуществом распараллеливания с использованием интерфейса OpenMP является возможность получения быстрого результата — работающей параллельной программы. При этом для небольшого количества процессоров может быть получено вполне приемлемое ускорение. Однако дальнейшая доработка может занимать значительно большее время.

Эта особенность связана с тем, что первоначально выполняется распараллеливание самых дорогостоящих по количеству вычислений блоков. Однако при необходимости повышения эффективности приходится распараллеливать средние и небольшие счетные блоки. Это не так просто, так как для небольших блоков время на организацию параллельных областей может оказаться сопоставимым со временем счета самого блока или даже превышать его. В таких случаях приходится искать пути объединения мелких блоков в более крупные или даже объединять все счетные блоки в одну параллельную область. Использование же больших параллельных областей хотя и снижает накладные расходы (повышая тем самым эффективность), однако требует дополнительной работы по выявлению точек синхронизации, конфликтов между счетными блоками и т. д.

Низкая мобильность программ. К недостаткам распараллеливания с использованием интерфейса OpenMP относится привязка к конкретной вычислительной системе. При переходе к другой вычислительной системе на *идеально распараллеленной программе* эффективность иногда может резко упасть (программа даже может выполняться медленнее, чем в последовательном режиме счета). Объясняется это тем, что при распараллеливании с использованием интерфейса OpenMP большую роль играют самые различные параметры вычислительной системы: системное программное обеспечение, скорость процессора, каналы доступа к памяти, наличие и объем кэш-памяти и др. В результате при переходе на другую вычислительную систему всю работу по выявлению и устранению

"узких мест" приходится начинать сначала. По этим причинам распараллеливание с использованием интерфейса OpenMP оказывается очень "чувствительным" к любому изменению конфигурации вычислительной системы.

Эта особенность приводит к тому, что для повышения эффективности зачастую приходится значительно модифицировать программы, подстраивая их под конкретную вычислительную систему. При этом на другой вычислительной системе данная программа уже, возможно, будет неоптимальной. В результате программу, распараллеленную с использованием интерфейса OpenMP, достаточно сложно поддерживать сразу на нескольких различных вычислительных системах.

Ограничения по масштабируемости. Одна из самых существенных проблем распараллеливания в модели общей памяти связана с ограничениями по масштабируемости, т. е. появлением трудностей при увеличении количества задействованных процессоров.

Это, с одной стороны, ограничения со стороны вычислительных систем: в системах на общей памяти с большим количеством процессоров скорость доступа к различным участкам оперативной памяти оказывается неоднородной, что снижает эффективность распараллеливания, выполненного в предположении, что скорость доступа одинакова.

С другой стороны, сложную программу редко удается распараллелить полностью (т. е. так, чтобы она всегда выполнялась в параллельной области). Как правило, некоторая небольшая часть программного кода на каждом счетном шаге выполняется последовательно. Однако при увеличении количества процессоров доля таких последовательных участков становится все больше. В результате максимальная эффективность оценивается следующим образом: $E^p_{\max} = \frac{1}{1 + d\,(p-1)}$ (d — доля времени выполнения нераспараллеленного участка в последовательном режиме счета; p — количество процессоров). Так, для методики ТИМ в последовательном режиме $d \approx 2{,}08\,\%$. В результате в параллельном режиме на 32 процессорах максимальная достижимая эффективность — всего 60,8%. Реальная эффективность оказывается еще ниже.

Решить данную проблему можно путем сокращения доли последовательных участков (что не всегда возможно). Другой путь решения — использование смешанного распараллеливания. В этом случае p равно количеству процессоров в составе одного вычислительного узла.

Заключение

Осуществлено распараллеливание программ, реализующих методику ТИМ (для решения как двумерных, так и трехмерных задач) в модели общей памяти с использованием интерфейса OpenMP.

Анализируя результаты выполненных расчетов, можно сделать следующие выводы:

- реализованные алгоритмы выполняются правильно;
- получены приемлемые значения ускорения и эффективности;
- возможно использование OpenMP-распараллеливания на нижнем уровне в рамках одного вычислительного узла параллельной вычислительной системы совместно с MPI-распараллеливанием.

В дальнейшем OpenMP-распараллеливание планируется развивать по следующим направлениям:

- для повышения эффективности необходимо уменьшить долю последовательных участков в кодах программ;
- пересмотреть некоторые алгоритмы для уменьшения количества циклов с небольшим количеством вычислений;
- усовершенствовать алгоритмы для получения наибольшей эффективности распараллеливания программ на общей памяти с использованием интерфейса OpenMP.

Список литературы

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. *Соколов С. С., Панов А. И., Воропинов А. А. и др.* Методика ТИМ расчета трехмерных задач механики сплошных сред на неструктурированных многогранных лагранжевых сетках // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2005. Вып. 3. С. 37—52.
3. MPI Documents (http://www.mpi-forum.org/docs/docs.html, 14.07.2007).
4. OpenMP specifications (http://www.openmp.org/drupal/node/view/8, 14.07.2007).
5. *Седов Л. И.* Методы подобия и размерности в механике. М.: Физматгиз, 1962.
6. *Воропинов А. А., Мотлохов В. Н., Рассказова В. В.* Распараллеливание счета по программе ДМК на многопроцессорных машинах с общей памятью с использованием интерфейса OpenMP // Молодежь в науке. Сб. докл. науч.-тех. конф. Саров, 11—13 марта 2002 г. Саров: РФЯЦ-ВНИИЭФ, 2002. С. 47—51.

Статья поступила в редакцию 22.06.07.