

УДК 519.6

РАСПАРАЛЛЕЛИВАНИЕ МЕТОДИКИ “ТИМ” В МОДЕЛИ ОБЩЕЙ ПАМЯТИ С ИСПОЛЬЗОВАНИЕМ ИНТЕРФЕЙСА OpenMP

А. А. Воропинов¹, И. Г. Новиков¹, И. В. Соболев¹, С. С. Соколов¹

Методика ТИМ предназначена для расчета нестационарных многомерных задач механики сплошной среды на лагранжевых сетках произвольной структуры. По методике ТИМ могут проводиться расчеты двумерных задач в цилиндрической и декартовой системах координат (ТИМ-2D) и трехмерных задач в декартовой системе координат (ТИМ-3D). Для сокращения времени расчетов в методике ТИМ реализовано распараллеливание в модели общей памяти с использованием интерфейса OpenMP. Распараллеливание программ осуществлено путем добавления директив распараллеливания OpenMP для каждого цикла, итерации которого могут выполняться независимо друг от друга. Для каждого цикла распараллеливание выполнено независимо. Распараллелены блоки расчета процессов газовой динамики, упругопластичности, магнитной гидродинамики, двухпотокости, двухтемпературности, поддержания счетной сетки, а также ряд вспомогательных процедур. В ряде случаев пришлось пересмотреть алгоритмы, использованные в последовательном режиме счета. Распараллеливание выполнено для счетных блоков, в последовательном режиме занимающих около 99 % времени счета. Реализованные алгоритмы проверены на ряде тестовых и методических расчетов. Эффективность на восьми процессорах составила 85 %. Статья подготовлена по материалам доклада авторов на международную научную конференцию “Параллельные вычислительные технологии” (ПаВТ-2007; <http://agora.guru.ru/pavt>).

1. Введение. Методика ТИМ предназначена для расчета нестационарных многомерных задач механики сплошной среды на лагранжевых сетках произвольной структуры. По методике ТИМ могут проводиться расчеты двумерных задач в цилиндрической и декартовой системах координат ТИМ-2D [1] и трехмерных задач в декартовой системе координат ТИМ-3D [2]. В настоящее время в методике ТИМ реализованы приближения для расчета нескольких классов задач, таких как: задачи газовой динамики, нестационарной упругопластичности, магнитной гидродинамики, теплопроводности, двухтемпературности и многопоточковой газодинамики.

Для повышения точности проводимых расчетов и более адекватного численного моделирования сложных задач необходимо производить численное моделирование на сетках с большим количеством ячеек (счетных точек). Проведение расчетов с большим количеством точек требуют значительного календарного времени. Один из путей сокращения календарных сроков проведения расчетов — это реализация распараллеливания счета.

Наиболее распространенными методами распараллеливания задач механики сплошной среды в настоящее время являются модель распределенной памяти (чаще всего используется интерфейс передачи сообщений MPI [3]) и модель общей памяти (интерфейс OpenMP [4]). Наиболее универсальным считается MPI, однако реализация распараллеливания с использованием передачи сообщений MPI очень сложна для нерегулярных методик. Поэтому было решено первоначально реализовать распараллеливание методики ТИМ в модели общей памяти с использованием интерфейса OpenMP.

Несмотря на сложность, работы в направлении распараллеливания в модели распределенной памяти в настоящее время активно ведутся как для двумерной методики ТИМ-2D, так и для трехмерной методики ТИМ-3D. В дальнейшем для вычислительных систем, работающих в модели смешанной памяти, может быть реализован режим смешанного распараллеливания. В этом случае интерфейс OpenMP может использоваться на нижнем уровне для распараллеливания счета на одном вычислительном узле внутри параобласти.

2. Принципы распараллеливания в модели общей памяти. Сформулируем основные принципы распараллеливания методики ТИМ в модели общей памяти, которых придерживались авторы.

¹ Всероссийский научно-исследовательский институт экспериментальной физики, пр. Мира, д. 37, 607190, г. Саров; e-mail: voron@md08.vniief.ru, novikov@vniief.ru, I.V.Sobolev@vniief.ru, ssokolov@vniief.ru

1. Распараллеливание итераций счетных циклов. Такое распараллеливание наиболее легко реализуется в модели общей памяти и позволяет использовать OpenMP совместно с другими способами распараллеливания.

2. Независимое распараллеливание каждого счетного блока, т.е. каждый счетный цикл распараллеливается независимо от другого, а блоки программы между этими циклами выполняются последовательно. При этом внутри каждого счетного блока количество параллельных областей все равно должно быть по возможности минимизировано. Такое распараллеливание также выполняется несколько проще, чем “сквозное” распараллеливание всех счетных блоков. Однако в случае небольшого количества счетных блоков и небольших последовательных участков такое распараллеливание оказывается вполне приемлемым. Тем не менее отказ от этого правила может быть одним из первых действий в случае необходимости повысить эффективность распараллеливания. Заметим, что “сквозное” распараллеливание достаточно трудоемко, так как 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 эти алгоритмы значительно лучше распараллеливаются. В результате при увеличении количества процессоров такой алгоритм работает быстрее, чем первоначальный, “хороший” последовательный алгоритм.

4. Использование профилирования. Перед выполнением распараллеливания оценивается временной вклад каждого счетного блока в общее время счета.

5. Минимизация дополнительной памяти. В процессе анализа кодов программ при определении параллельных областей было решено не описывать “большие” массивы (ячеечные, узловые, “контейнерные”) как частные для каждой нити (клауза PRIVATE). Такие массивы описывались как общие (клауза 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.

Таблица 1

Процентное соотношение выполнения отдельных блоков методики ТИМ

Название процесса	Процент от полного времени исполнения (в скобках: процент исполнения программного кода отдельного блока в параллельной области от времени исполнения всего 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)
Неучтенные	0.2 (0)	0.5 (0)
В итоге доля (в %) последовательного участка	1.479	2.078

3. Распараллеливание методики ТИМ по физическим процессам. На начальном этапе работ по распараллеливанию методики ТИМ были сделаны оценки времени работы каждого блока методики, отвечающего за те или иные вычисления. В табл. 1 приведены (в процентном соотношении к времени расчета всей задачи) замеры времени исполнения отдельных блоков методики. Как видно из табл. 1, доли выполнения последовательных участков кодов программ методик, т.е. участков кодов программ, которые не были распараллелены для двумерного и трехмерного случаев, практически одинаковы и составляют примерно 1.5–2%. Следует отметить, что в данной таблице представлены замеры для тестовой задачи о сильном взрыве, которая будет рассмотрена ниже.

4. Исследование эффективности распараллеливания. Тестирование распараллеленных программ методики ТИМ в модели общей памяти с использованием интерфейса OpenMP было проведено

на модельных задачах газодинамики и упругопластичности. При проведении расчетов получено совпадение результатов между параллельными и последовательными расчетами на параллельных машинах, на одном вычислительном узле которых имеется более одного процессора на общей памяти. Совпадение получено по всем контролируемым величинам: число счетных шагов, величина временного шага, суммарная кинетическая, внутренняя, полная энергия на каждом счетном шаге и ряд других величин.

В качестве характеристик эффективности распараллеливания использовались следующие функции: $S_p = \frac{t_1}{t_p}$ — ускорение счета; $E_p = \frac{t_1}{pt_p} \cdot 100\%$ — эффективность распараллеливания, где t_1 — время расчета на одном процессоре используемой параллельной машины, t_p — время счета на p процессорах; $E_{\text{MAX}} = \frac{1}{1+d(p-1)} \cdot 100\%$ — максимальная эффективность распараллеливания на p процессорах при наличии доли d последовательного участка программ от общего времени исполнения.

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

4.1. Методическая задача о “сильном взрыве”. Расчет на нерегулярной сетке. Данная задача широко известна как задача о “седовском взрыве” [5].

Физическая постановка задачи. Рассматривается задача о сферически-симметричном движении газа, возникающем в результате точечного взрыва в однородном веществе без противодействия.

Начальная геометрия: шар радиуса $R = 0.05$ с центром в начале координат.

Вещество: во всей области находится идеальный газ с уравнением состояния $p = (\gamma - 1)\rho E$, $\gamma = 1.4$.

Начальные условия: начальное время $t_0 = 0$; в центральной ячейке области задана внутренняя энергия. Все поверхности области представляют собой жесткие стенки.

Математическая постановка задачи. Задачу считали в однообластной постановке. Число ячеек в области равно 120 000. Внутренняя энергия задавалась в начальный момент равной 10^5 . Данная задача использует все процессы, описанные в табл. 1.

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

На рис. 1 и 2 приведены графики зависимости эффективности распараллеливания в зависимости от числа процессоров в целом по задаче в сравнении с максимально возможной эффективностью при наличии доли последовательного участка.

4.2. Методическая задача о “плоской волне”. Рассмотрим расчет задачи о “плоской волне” в газодинамической постановке на шестигранной (в двумерном случае на четырехугольной) счетной сетке.

Физическая постановка задачи. Задача описывает распространение плоской ударной волны по области, заполненной газом.

Начальная геометрия: параллелепипед (в двумерном случае цилиндр).

Вещество: во всей области находится идеальный газ с уравнением состояния $p = (\gamma - 1)\rho E$, $\gamma = 3$.

Начальные условия: начальное время $t_0 = 0$. В начальный момент времени на свободной границе области задается постоянное давление. Все поверхности области представляют собой жесткие стенки,

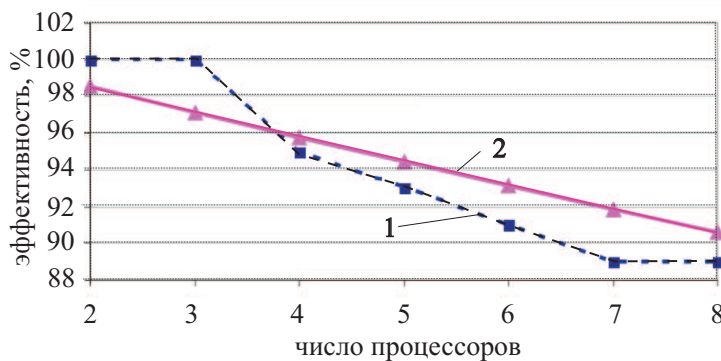


Рис. 1. Зависимости ускорения счета от числа процессоров методики ТИМ-2D: E_p (1), E_{MAX} (2)

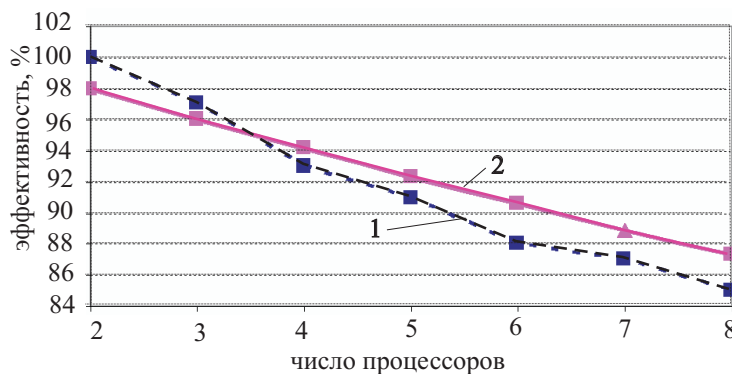


Рис. 2. Зависимости эффективности распараллеливания от числа процессоров методики ТИМ-3D: E_p (1), E_{MAX} (2)

Таблица 2

Ускорение счета всей задачи на различном числе процессоров по методике ТИМ

“Имя” блока	Ускорение счета, число раз						
	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

Эффективность распараллеливания всей задачи на различном числе процессоров по методике ТИМ

“Имя” блока	Эффективность распараллеливания, %						
	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

кроме поверхности приложения давления.

Математическая постановка задачи. Задачу считали в однообластной постановке. Число ячеек в области варьировалось и составляло для разных расчетов: 52 тыс., 105 тыс., 210 тыс., 420 тыс., 840 тыс. и 1260 тыс. Давление на свободной границе задавалось равным 5.

В данной задаче движение вещества происходит вдоль одного направления, совпадающего с линиями сетки, поэтому искажения счетной сетки не происходило и функциональной нагрузки на блоки коррекции с перестройкой счетной сетки нет. Для таких расчетов доля последовательного (не распараллеленного) участка программы сводится до минимума, а именно: в ТИМ-2D она составила менее 0.0002%, а в методике ТИМ-3D — менее 0.0001%. В табл. 4 приведены (в процентном соотношении к времени расчета всей задачи) замеры времени исполнения отдельных блоков методики. Как видно из таблицы, доли участков кодов программ, которые не были распараллелены для двумерного и трехмерного случаев, практически одинаковы и составляют менее 0.0002%.

Таблица 4

Процентное соотношение выполнения отдельных блоков методики ТИМ для второй тестовой задачи

Название процесса	Процент от полного времени исполнения (в скобках: процент исполнения программного кода отдельного блока в параллельной области от времени исполнения всего блока)	
	ТИМ-2D	ТИМ-3D
Счет уравнения движения	41.4 (100)	53.6 (100)
Счет уравнения энергии	58.6 (100)	46.4 (100)
В итоге доля (в %) последовательного участка	< 0.0002	< 0.0001

В табл. 5 и 6 представлены ускорение и эффективность распараллеливания для рассматриваемой задачи с различным числом счетных ячеек на различном числе процессоров как по методике ТИМ-2D, так и по методике ТИМ-3D. На рис. 3 и 4 приведены графики зависимости эффективности распараллеливания от числа процессоров в целом по задаче в сравнении с максимально возможной эффективностью при наличии доли последовательного участка.

Исходя из данных в представленных таблицах, можно сделать вывод, что для методики ТИМ-2D наблюдается насыщение ускорения, начиная с 52 тыс. ячеек, а для методики ТИМ-3D насыщение наблюдается при числе ячеек 840 тыс. и более.

Как видно из таблиц, эффективность распараллеливания двумерной методики оказалась ниже, чем эффективность распараллеливания трехмерной методики. Это связано, во-первых, со значительными различиями в процентном соотношении между расчетом уравнения энергии и движения в обеих методиках. Так, для методики ТИМ-3D более дорогим, с точки зрения объема вычислений, является расчет уравнения движения (из-за значительно большего количества узлов по сравнению с ячейками), а в методике ТИМ-2D расчет уравнения движения является менее дорогой операцией. Во-вторых, циклы с небольшим объемом вычислений (“малые” циклы) хуже поддаются распараллеливанию из-за возрастающего отношения накладных расходов по отношению к полезной нагрузке. В двумерной методике при расчете уравнения движения имеется два таких “малых” цикла (для узлов, лежащих на границах), которые распараллеливаются с эффективностью менее 50%. Для улучшения эффективности распараллеливания необходимо пересмотреть их с алгоритмической точки зрения. В трехмерной методике подобные циклы распараллеливаются с эффективностью не менее 80%, что связано с большим количеством граничных узлов.

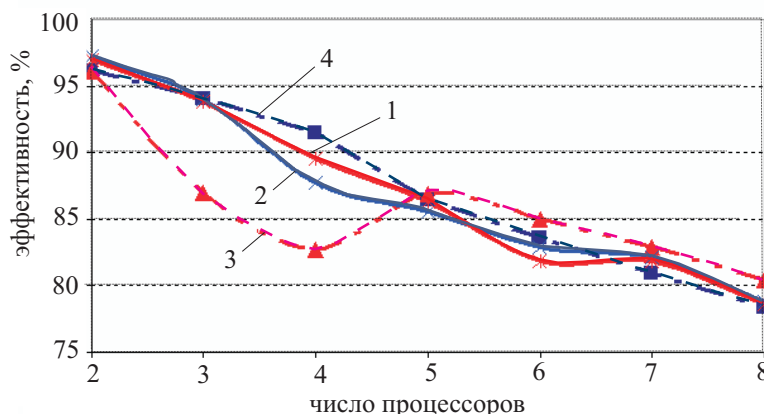


Рис. 3. Зависимости эффективности распараллеливания от числа процессоров методики ТИМ-2D: 52 тыс. (4), 105 тыс. (3), 210 тыс. (2) и 420 тыс. ячеек (1)

Таблица 5

Ускорение счета всей задачи на различном числе процессоров по методике ТИМ

Число ячеек	Методика	Ускорение счета (округленное), число раз						
		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
1260 тыс.	ТИМ-3D	1.96	2.96	3.90	4.78	5.78	6.72	7.61

При анализе графиков зависимости эффективности распараллеливания можно сделать вывод о том, что, по сравнению с теоретически достигаемой эффективностью при наличии последовательного участка программ, для данной задачи получены хорошие результаты по ускорению счета для трехмерной методики (ТИМ-3D). Двумерная методика ТИМ-2D показала 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 как минимум в два раза больше. Из-за этого эффективность распараллеливания на нерегулярной сетке получилась выше

вследствие большей сбалансированности вычислительной нагрузки на расчет одной ячейки сетки.

Таблица 6

Эффективность распараллеливания всей задачи на различном числе процессоров по методике ТИМ

Число ячеек	Методика	Эффективность распараллеливания (округленная), %						
		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
1260 тыс.	ТИМ-3D	98.0	98.7	97.5	95.6	96.3	96.0	95.2

5. Некоторые особенности распараллеливания в модели общей памяти. Общеизвестно, что разработка параллельных программ связана с дополнительными сложностями, не присущими последовательным программам. Это и сложность отладки, и наличие дополнительных накладных расходов на синхронизацию и обмен данными, и ряд других. В этой связи авторы считают необходимым отметить ряд особенностей, присущих распараллеливанию в модели общей памяти с использованием интерфейса OpenMP. Данные замечания следуют из опыта авторов, полученного при распараллеливании методик ДМК [6], ТИМ-2D и ТИМ-3D.

5.1. Сложности отладки программ. Эта проблема присуща и распараллеливанию в модели распределенной памяти, однако для распараллеливания с использованием интерфейса OpenMP она стоит еще острее. При распараллеливании в модели общей памяти большая часть данных разделяется между процессорами и оказывается очень сложно оценить — 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 и используемых переменных).

5.2. Быстрый результат — длительная доработка. Несомненным преимуществом распараллеливания с использованием интерфейса OpenMP является возможность получения быстрого результата: работающей параллельной программы. При этом для небольшого количества процессоров может быть получена вполне приемлемая эффективность. Однако дальнейшая отработка может занимать значительно больше время.

Эта особенность связана с тем, что первоначально выполняется распараллеливание для самых “дорогих” по количеству вычислений блоков. Однако при необходимости повысить эффективность приходится выполнять распараллеливание небольших и средних счетных блоков, что уже более проблематично, так как для небольших блоков накладные расходы на организацию параллельных областей могут оказаться сопоставимыми или даже превышать время счета самого блока. В этих случаях приходится искать пути объединения мелких блоков в более крупные или даже объединения всех счетных блоков 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 др.

5.3. “На новой машине — все сначала”. Другим недостатком для распараллеливания с использованием интерфейса OpenMP является привязанность к конкретной вычислительной системе. При переходе к другой вычислительной системе на “идеально распараллеленной программе” эффективность

вдруг может резко упасть (программа может даже выполняться медленнее, чем в последовательном режиме счета). Объясняется это тем, что для распараллеливания с использованием интерфейса OpenMP очень большую роль играют самые различные параметры вычислительной системы: от используемого системного ПО до скорости процессора, каналов доступа к памяти, наличия кеш-памяти и др. В результате распараллеливание с использованием интерфейса OpenMP оказывается очень чувствительным к любому изменению конфигурации вычислительной системы.

В результате при переходе на другую вычислительную систему всю работу по оптимизации программ иногда приходится начинать сначала. (Под оптимизацией мы понимаем доработку программ для предотвращения резкого снижения эффективности.) При этом выявляются области снижения эффективности, выясняется причина, ищутся пути обхода или решения возникшей проблемы.

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

5.4. Совместимость по ошибкам. В 2001 г. при реализации распараллеливания с использованием интерфейса OpenMP для методики ДМК [6] возник ряд трудностей, связанных с ошибками в компиляторе. К сожалению, ситуация с тех пор кардинально не улучшилась. В компиляторах, которые использовали авторы, в режиме генерации кода с таким распараллеливанием часто возникают самые различные ошибки.

Эти ошибки возникают и проявляются по-разному. Как правило, это недиагностируемые “вылеты” программы во время выполнения, режис “авосты” компилятора или неправильная работа программы. При этом наиболее трудным для обнаружения и диагностики оказывается именно последний случай, когда программа выдает неправильный результат.

Для решения этих проблем приходится модифицировать программы так, чтобы “проблемные” конструкции не использовались.

5.5. Ограничения по масштабируемости. Одной из самых существенных проблем для распараллеливания в модели общей памяти является ограничение по масштабируемости, т.е. по увеличению количества задействованных процессоров.

Это, с одной стороны, ограничения со стороны вычислительных систем: в вычислительных 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, сложные программы 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 и больше. В результате максимальная эффективность оценивается следующим образом: $E_{\text{MAX}}^P = \frac{1}{1 + \alpha(P - 1)}$, где α — доля последовательного участка, P — количество процессоров. Так, для методики ТИМ доля нераспараллеленного блока в последовательном режиме составляла 2.9%. Это привело к тому, что в параллельном режиме на 20 процессорах максимальная достижимая эффективность составляет уже всего $E_{\text{MAX}}^{20} \simeq 64.5\%$.

Решить данную проблему можно путем сокращения доли последовательных участков (что не всегда возможно сделать). Другой путь решения — использование смешанного распараллеливания.

6. Заключение. Осуществлено распараллеливание методики ТИМ (2D и 3D) в модели общей памяти с использованием интерфейса OpenMP.

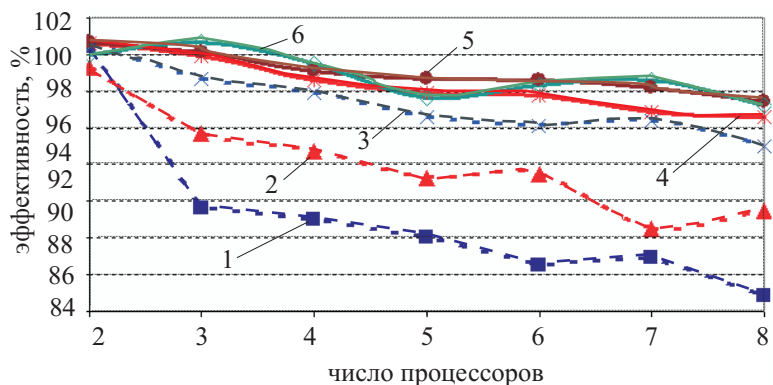


Рис. 4. Зависимости эффективности распараллеливания от числа процессоров методики ТИМ-3D: 52 тыс. (1), 105 тыс. (2), 210 тыс. (3), 420 тыс. (4), 840 тыс. (5) и 1260 тыс. ячеек (6)

Результаты проведенных расчетов показали:

— для повышения эффективности необходимо минимизировать долю последовательных участков кодов программ;

— необходимо пересмотреть некоторые алгоритмы для минимизации количества “малых” циклов;

— необходимо и далее усовершенствовать алгоритмы программ для получения наибольшей эффективности распараллеливания на общей памяти с использованием интерфейса OpenMP.

По результатам исследований для распараллеливания методики ТИМ (2D и 3D) решено использовать интерфейс OpenMP на процессорах одного вычислительного узла (нижний уровень) и интерфейс MPI между узлами параллельной машины (верхний уровень).

СПИСОК ЛИТЕРАТУРЫ

1. Соколов С.С., Воронинов А.А., Панов А.И. 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. № 4. 29–43.
2. Соколов С.С., Панов А.И., Воронинов 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. № 3. 37–52.
3. MPI-2: Extensions to the Message-Passing Interface (<http://www.mpi-forum.org/docs/mpi-20.ps>), 22.02.2004.
4. OpenMP Fortran Application Program Interface. Version 2.0, November 2000 (<http://www.openmp.org/spec/mp-documents/>), 13.04.2005.
5. Седов Л.И. Методы подобия и размерности в механике. М.: Физматгиз, 1962.
6. Воронинов 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. докладов научно-технической конференции. Саров: ВНИИЭФ, 2002. 47–51.

Поступила в редакцию
29.03.2007
