УДК 519.632.4

РАСПАРАЛЛЕЛИВАНИЕ УНИВЕРСАЛЬНОЙ МНОГОСЕТОЧНОЙ ТЕХНОЛОГИИ

С. И. Мартыненко[1]

Предлагается новый способ распараллеливания многосеточных алгоритмов, предназначенных для решения дифференциальных уравнений в частных производных эллиптического типа. Подход основан на взаимной адаптации архитектуры многопроцессорного компьютера и цикла универсальной многосеточной технологии. Адаптация архитектуры к многосеточной технологии позволяет получить оптимальную загрузку процессоров. С другой стороны, адаптация многосеточного цикла к выбранной архитектуре приводит к уменьшению объема пересылаемых данных. Распараллеливание многосеточной технологии на уровнях с грубыми сетками не зависит от выбора сглаживающей процедуры. Показано, что эффективность распараллеливания возрастает по мере увеличения объема вычислений при решении более сложных задач.

Ключевые слова: численные алгоритмы, математическое моделирование, распараллеливание вычислительных методов, дифференциальные уравнения эллиптического типа, многопроцессорные вычислительные системы, многосеточные алгоритмы.

1. Введение. Прогресс вычислительной техники привел к повышению роли математического моделирования при решении научно-технических задач. Развитие современного программного обеспечения имеет четкую тенденцию к разработке продуктов, устроенных по принципу "черного ящика", для снижения времени отладки программ и других издержек вычислительного эксперимента. Подобные продукты уже получили широкое распространение для решения задач механики жидкости и газа и других приложений. Немаловажную роль играет также возможность эффективного распараллеливания вычислений.

Однако развитие программного обеспечения привело к необходимости пересмотра основных принципов построения вычислительных алгоритмов. В качестве примера рассмотрим некоторую краевую задачу для уравнения

$$\mathcal{L}u = f,$$

где $\mathcal{L}$ есть линейный эллиптический оператор. Дополнительным ограничением является требование "плавности" первых собственных функций оператора, которое для эллиптических задач обычно выполнено.

Базовый алгоритм численного решения поставленной краевой задачи представим в виде следующей последовательности действий:

1) построение вычислительной сетки,

2) аппроксимация краевой задачи на построенной сетке,

3) упорядочивание неизвестных и сведение сеточных уравнений к виду $Ax = b$,

4) решении полученной системы $Ax = b$ (например, методом Якоби или Зейделя).

Методы Якоби и Зейделя не содержат каких-либо проблемно-зависимых компонент и позволяют эффективно распараллеливать вычисления. Единственным недостатком базового алгоритма является его медленная скорость сходимости. Как правило, более быстро сходящиеся методы не обладают достаточным уровнем формализации и не позволяют эффективно распараллеливать вычисления.

Выход из столь противоречивой ситуации был предложен Р. П. Федоренко в 1961 г. Позднее эти идеи получили развитие в работах Н. С. Бахвалова и Г. П. Астраханцева, а также в работах западных математиков. Использование особенностей методов Якоби и Зейделя позволило в значительной мере сохранить преимущества базового алгоритма и существенно уменьшить объем вычислений, необходимых для получения численного решения. Однако первая алгоритмическая реализация основополагающей идеи Р. П. Федоренко, которая получила название "классические многосеточные методы" (КММ), была основана на адаптации алгоритма к задаче и поэтому не могла быть формализована. Фактически повышение скорости сходимости в КММ достигалось за счет снижения универсальности (робастности).

[1] Центральный институт авиационного моторостроения им. П. И. Баранова, ул. Авиамоторная, д. 2, 111250, Москва; e-mail: martyn_s@mail.ru

В [1] было показано, что подход Р. П. Федоренко допускает вторую алгоритмическую реализацию, которая получила название “универсальная многосеточная технология” (УМТ). Для рассматриваемой краевой задачи с указанным выше ограничением на оператор $\mathcal{L}$ технология УМТ содержит только один компонент, который может отсутствовать в базовом алгоритме, а именно более сложную аппроксимацию граничных условий. Фактически робастность УМТ достигается за счет незначительного снижения скорости сходимости по сравнению с оптимальной [1]. Однако высокая степень формализации УМТ позволяет использовать ее в современных программных продуктах.

В данной статье излагаются результаты исследований различных схем распараллеливания УМТ. Основное внимание уделяется получению оценок минимального ускорения и эффективности параллелизма при решении краевых задач.

Рассмотрим построение грубых сеток, используемых при решения N-мерной задачи, для иллюстрации распределения вычислительных сеток среди процессоров. Предполагается, что область Ω является N-мерным кубом. Равномерная сетка $G(0;1)$ для последующей дискретизации методом контрольного объема состоит из двух систем сеточных точек $G^{\mathrm{v}}(0;1)$ и $G^{\mathrm{f}}(0;1)$ определяемых как

$$G^{\mathrm{v}}(0;1) = \{x_i^{\mathrm{v}} : x_i^{\mathrm{v}} = h(i-1),\ \ i = (i_1, i_2, \ldots, i_N),\ \ h = (h_1, h_2, \ldots, h_N),$$
$$i_\alpha = 1, 2, \ldots, \mathrm{H}_\alpha + 1,\ \ h_\alpha = \mathrm{H}_\alpha^{-1},\ \ \alpha = 1, 2, \ldots, N\},$$
$$G^{\mathrm{f}}(0;1) = \{x_i^{\mathrm{f}} : x_i^{\mathrm{f}} = 0.5(x_i^{\mathrm{v}} + x_{i+1}^{\mathrm{v}}),\ \ i = (i_1, i_2, \ldots, i_N),\ \ i_\alpha = 1, 2, \ldots, \mathrm{H}_\alpha,\ \ \alpha = 1, 2, \ldots, N\}.$$

Самая мелкая сетка $G(0;1)$ может быть представлена в виде совокупности 3^N грубых сеток $G(1;1)$, $G(1;2)$, …, $G(1;3^N)$. Потребуем, чтобы все грубые сетки не имели общих сеточных точек и каждая сеточная точка на грубой сетке совпадала с одной точкой самой мелкой сетки

$$G(0;1) = \bigcup_{k=1}^{3^N} G(1;k) \quad \text{и} \quad G(1;n) \cap G(1;m) = \varnothing \quad \text{при} \quad n \neq m.$$

Самая мелкая сетка $G(0;1)$ образует нулевой сеточный уровень, а 3^N грубых сеток $G(1;1)$, $G(1;2)$, …, $G(1;3^N)$ образуют первый сеточный уровень. Пример построения грубых сеток для одномерной задачи приведен в [1]. Очевидно, что сеточные уравнения на сетках первого уровня могут быть решены параллельно. Следует заметить, что все грубые сетки $G(1;k)$, $k = 1, \ldots, 3^N$, имеют приблизительно одинаковое число сеточных точек $3^{-N}\bar{\mathbf{N}}$, где $\bar{\mathbf{N}}$ есть число узлов самой мелкой сетки. Затем построение грубых сеток продолжается рекуррентным образом: каждая сетка $G(L;k)$ $(k = 1, \ldots, 3^{NL})$ L-го уровня рассматривается как самая мелкая сетка для грубых сеток $G(L+1;j)$ $(j = 1, \ldots, 3^{N(L+1)})$ следующего $(L+1)$-го уровня. Полученные из 3^N сеток первого уровня 3^{2N} более грубых сеток образуют второй сеточных уровень и т.д. Сетки $G(L;k)$ $(k = 1, \ldots, 3^{NL})$ имеют $3^{-NL}\bar{\mathbf{N}}$ сеточных точек, каждый контрольный объем на грубой сетке $G(L;k)$ есть объединение 3^{NL} на самой мелкой сетке. Нетрудно видеть, что самая мелкая сетка может быть представлена в виде совокупности всех сеток каждого уровня L:

$$G(0;1) = \bigcup_{k=1}^{3^{NL}} G(L;k) \quad \text{и} \quad G(L;n) \cap G(L;m) = \varnothing \quad \text{при} \quad n \neq m, \quad L = 1, \ldots, L^+,$$

где L^+ есть номер уровня с самыми грубыми сетками.

С точки зрения распараллеливания вычислений технология УМТ обладает следующими особенностями.

1) Грубые сетки каждого уровня не имеют общих точек

$$G(L;n) \cap G(L;m) = \varnothing, \quad n \neq m, \quad n, m = 1, \ldots, 3^{NL}, \quad L = 1, \ldots, L^+.$$

Следовательно, сглаживающие итерации на этих сетках могут проводиться параллельно.

2) Отсутствие общих сеточных точек уменьшает обмен данными между процессорами.

3) Фиксированное число сеток (3^{NL}) на L-ом уровне позволяет заранее предсказать необходимое число процессоров.

4) Одинаковое число точек $(3^{-NL}\bar{\mathbf{N}})$ на каждой сетке L-ого уровня позволяет добиться оптимальной загрузки процессоров.

5) Самая мощная стратегия отыскания поправок снижает требования, которые предъявляются к выбору сглаживающей процедуры в многосеточных итерациях [1]. Поэтому при распараллеливании УМТ

возможен выбор тех упорядочиваний неизвестных и сглаживателей, которые позволят получить почти полный параллелизм для многих приложений.

В данной работе приводится архитектура, адаптированная к УМТ, многосеточный цикл, адаптированный к выбранной архитектуре, оценки минимального ускорения и эффективности, а также результаты вычислительных экспериментов.

2. Архитектура многопроцессорного компьютера. До настоящего времени развитие архитектур и способов распараллеливания вычислительных алгоритмов для решения систем линейных уравнений происходило почти независимыми путями. Многочисленные конфигурации процессоров были получены посредством различных комбинаций процессоров. О достоинствах и недостатках наиболее популярных архитектур было опубликовано большое число работ [2, 4]. Кроме того, различные численные методы имеют разную степень параллелизма [5].

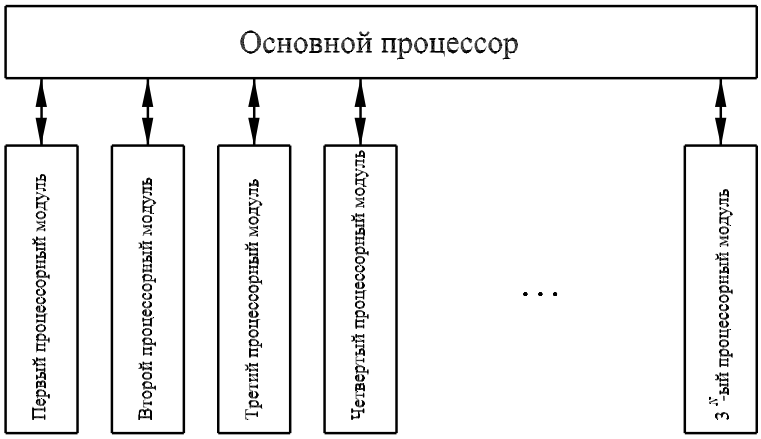

Рис. 1. Межпроцессорные связи для распараллеливания универсальной многосеточной технологии

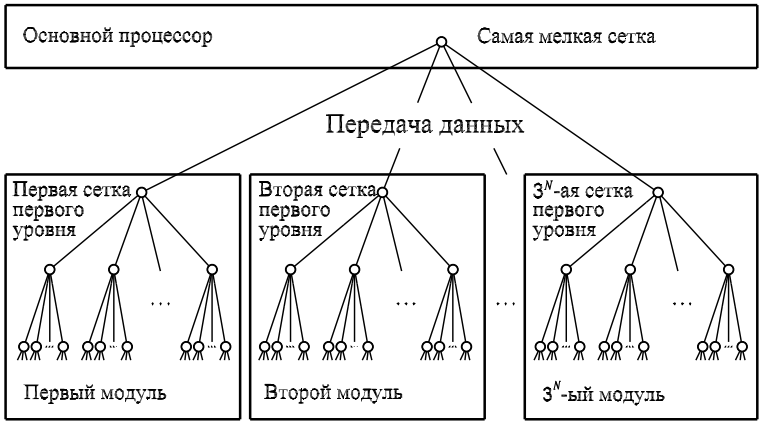

Рис. 2. Распределение вычислительных сеток среди процессорных модулей

Предположим, что для распараллеливания УМТ будет использован многопроцессорный компьютер с локальной памятью. Система состоит из блоков, каждый из которых включает в себя процессорный модуль (PM) и собственную локальную память. Поскольку все грубые сетки $G(L;n)$ не имеют общих узлов, каждый процессорный модуль может быть связан только с основным процессором (MP). Любое общение между MP и PM осуществляется посредством передачи сообщений. Так как число сеток на L-ом уровне есть 3^{NL}, то число PM (p) должно быть равно 3^{Nk}, где $k = 1, \ldots, L^+$. В дальнейшем будет рассмотрен простейший случай: $k = 1 \Rightarrow p = 3^N$, $N = 2, 3$. Схема соединения процессоров для распараллеливания УМТ показана на рис. 1. Основной процессор содержит дискретную формулировку распараллеливаемой задачи на самой мелкой сетке. Данная конфигурация процессоров использует минимальное число связей. Распределение вычислительных сеток среди процессорных модулей показано на рис. 2.

Распараллеливание УМГ в рамках данной процессорной конфигурации ($p = 3^N$) может быть разделено на два этапа.

Этап 1. Уровни с грубыми сетками ($0 < L \leqslant L^+$). Поправки на данных уровнях могут быть вычислены независимо на p процессорах. Распараллеливание на уровнях с грубыми сетками не зависит от выбора сглаживающей процедуры.

Этап 2. Самая мелкая сетка ($L = 0$). Только хорошо распараллеливаемые сглаживатели могут быть использованы на самой мелкой сетке для избежания простоя $p - 1$ процессоров.

3. Оценка минимального ускорения и эффективности. Ускорением $\mathbf{S}_p$ параллельного алгоритма называется отношение

$$\mathbf{S}_p = \frac{T(1)}{T(p)},$$

где $T(1)$ — время, необходимое для выполнения программы на однопроцессорном компьютере, и $T(p)$ — время счета на p процессорах. Ускорение $\mathbf{S}_p$ позволяет сравнить поведение данного алгоритма для одного и p процессоров. Эффективностью параллельного алгоритма $\mathbf{E}_p$ называется величина

$$\mathbf{E}_p = \frac{\mathbf{S}_p}{p}.$$

Для оценки минимального ускорения и эффективности предположим, что $p-1$ процессоров простаивают на самой мелкой сетке и время пересылки данных между MP и PM пренебрежимо мало.

С учетом принятых допущений минимальное ускорение $(\mathbf{S}_p)_m$ и эффективность $(\mathbf{E}_p)_m$ оценивается как

$$(\mathbf{S}_p)_m = \frac{T_0 + \sum\limits_{l=1}^{L^+} T_l}{T_0 + \dfrac{1}{p}\sum\limits_{l=1}^{L^+} T_l}, \qquad (\mathbf{E}_p)_m = \frac{1}{p}\frac{T_0 + \sum\limits_{l=1}^{L^+} T_l}{T_0 + \dfrac{1}{p}\sum\limits_{l=1}^{L^+} T_l}, \tag{1}$$

где T_l — время решения задачи на l-ом уровне.

Оценки минимального ускорения и эффективности параллельного исполнения для данной задачи могут быть получены из вычислительных экспериментов, выполненных на однопроцессорном компьютере.

Тест 1 (пилообразный цикл). Следуя [1], рассмотрим первую краевую задачу для уравнения Пуассона

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + F(x,y) = 0 \tag{2}$$

в $\Omega = (0,1) \times (0,1)$. Предположим, что для решения данной задачи построена равномерная сетка 730×730 ($L^+ = 5$). Функция F выбрана таким образом, что точное решение есть $U_e(x,y) = f(x)f(y)$, где $f(\varrho) = 10\big(e^{\varrho} + (1-e)\varrho - 1\big)$. Погрешность численного решения определим в виде

$$\mathrm{E} = \max_{ij} \big| U_e(x_i^{\mathrm{v}}, y_j^{\mathrm{v}}) - U_{ij} \big|.$$

Осредненные факторы уменьшения невязки ρ_q и $\rho_{\hat{\nu}}$ определяются в виде

$$\rho_q = \left(\frac{\|R^{(q)}\|}{\|R^{(0)}\|}\right)^{1/q}, \qquad \rho_{\hat{\nu}} = \left(\frac{\|R^{(\hat{\nu})}\|}{\|R^{(0)}\|}\right)^{1/\hat{\nu}},$$

где $\|\cdot\|$ есть L_2-норма, R — невязка, вычисленная на самой мелкой сетке, $R^{(0)}$ — невязка начального приближения, q — число многосеточных циклов и $\hat{\nu}$ — эквивалентное число сглаживающих итераций.

Таблица 1

Скорость сходимости в первом тесте

L^+	q	$\hat{\nu}_\Sigma$	ρ_q	$\rho_{\hat{\nu}}$	E
5	3	60	0,019	0,820	$1{,}63 \cdot 10^{-6}$

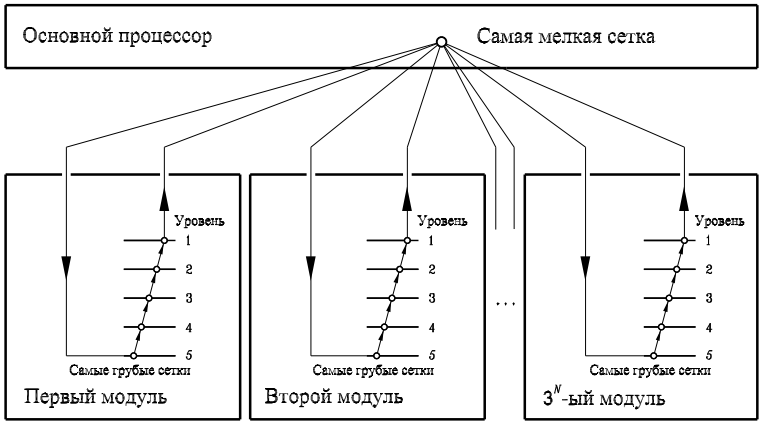

Рис. 3. Распараллеливание пилообразного цикла на грубых сетках

Таблица 2

Значения $(\mathbf{S}_p)_m$ и $(\mathbf{E}_p)_m$ по результатам первого теста

T_0	T_1	T_2	T_3	T_4	T_5	$(\mathbf{S}_p)_m$	$(\mathbf{E}_p)_m$
13,053 %	14,701 %	14,971 %	14,357 %	14,808 %	28, 110 %	4,41	0,49

Предполагаемое распараллеливание пилообразного цикла показано на рис. 3. В табл. 1 и 2 представлены скорость сходимости и полученные значения ускорения $(\mathbf{S}_p)_m$ и $(\mathbf{E}_p)_m$ эффективности.

Предполагая, что $T_0 = T_1 = \cdots = T_{L^+}$, получим грубые оценки минимального ускорения $(\mathbf{S}_p)_m$ и эффективности $(\mathbf{E}_p)_m$ в первом тесте

$$(\mathbf{S}_p^N)_m \approx 3^N \frac{L^+ + 1}{L^+ + 3^N}, \qquad (\mathbf{E}_p^N)_m \approx \frac{L^+ + 1}{L^+ + 3^N}. \tag{3}$$

Таблица 3

Оценка минимального ускорения и эффективности (статический цикл)

L^+	$(\mathbf{S}_p^2)_m$	$(\mathbf{E}_p^2)_m$	Самая мелкая сетка	$(\mathbf{S}_p^3)_m$	$(\mathbf{E}_p^3)_m$	Самая мелкая сетка
0	1,00	0,11	4×4	1,00	0,04	$4 \times 4 \times 4$
1	1,80	0,20	10×10	1,93	0,07	$10 \times 10 \times 10$
2	2,45	0,27	28×28	2,79	0,10	$28 \times 28 \times 28$
3	3,00	0,33	82×82	3,60	0,13	$82 \times 82 \times 82$
4	3,46	0,38	244×244	4,35	0,16	$244 \times 244 \times 244$
5	3,86	0,43	730×730	5,06	0,19	$730 \times 730 \times 730$
6	4,20	0,47	2188×2188	5,73	0,21	$2188 \times 2188 \times 2188$

В табл. 3 представлены значения $(\mathbf{S}_p^N)_m$ и $(\mathbf{E}_p^N)_m$ при распараллеливании первой краевой задачи для уравнения Пуассона, полученные из (3). Очевидно, что ускорение $\mathbf{S}_p$ и эффективность $\mathbf{E}_p$ параллельного алгоритма могут быть оценены как

$$3^N \frac{L^+ + 1}{L^+ + 3^N} \approx (\mathbf{S}_p^N)_m \; б \; \mathbf{S}_p < 3^N, \qquad \frac{L^+ + 1}{L^+ + 3^N} \approx (\mathbf{E}_p^N)_m \; б \; \mathbf{E}_p < 1.$$

4. Адаптация многосеточного цикла к архитектуре многопроцессорного компьютера. Для дальнейшего повышения эффективности распараллеливания необходимо уменьшить число арифметических операций, выполняемых на самой мелкой сетке. Согласно (1), асимптотическое поведение ускорения

и эффективности выражается как

$$\mathbf{S}_p \to p - (p-1)\,\frac{T_0}{\sum\limits_{l=1}^{L^+} T_l} \quad \text{и} \quad \mathbf{E}_p \to 1 - \frac{p-1}{p}\,\frac{T_0}{\sum\limits_{l=1}^{L^+} T_l} \qquad \text{при} \quad \sum_{l=1}^{L^+} T_l \gg T_0.$$

Ускорение и эффективность могут быть улучшены посредством адаптации многопроцессорного цикла к выбранной архитектуре многопроцессорного компьютера ($p = 3^N$). Для этого в процессе решения задачи будем рассматривать грубые сетки первого уровня как самые мелкие. В дальнейшем грубые сетки первого уровня будут называться *динамическими мелкими сетками*. Пилообразный цикл с динамическими мелкими сетками будет называться *динамическим пилообразным циклом*. Предполагаемое распараллеливание динамического цикла показана на рис. 4.

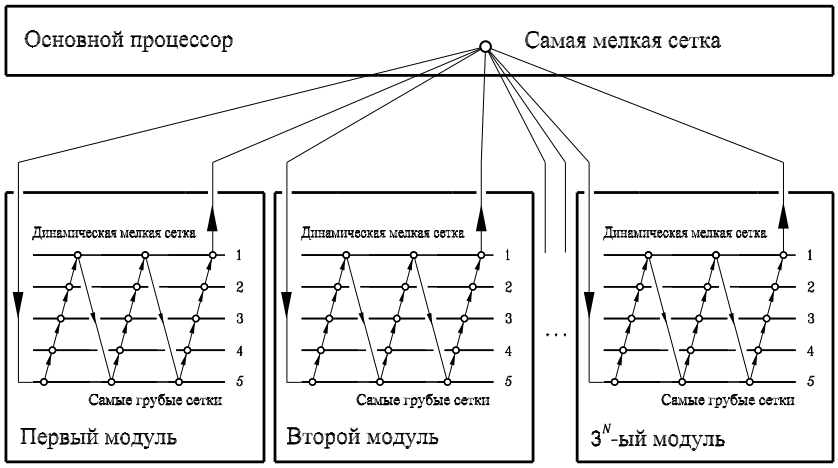

Рис. 4. Распараллеливание динамического пилообразного цикла на грубых сетках

Тест 2 (динамический пилообразный цикл). Во втором тесте для решения задачи (2) использован динамический цикл. В табл. 4 и 5 представлены скорость сходимости и значения ускорения $(\mathbf{S}_p)_m$ и эффективности $(\mathbf{E}_p)_m$ во втором тесте.

Таблица 4

Скорость сходимости во втором тесте

L^+	q^*	$\hat{\nu}_\Sigma$	ρ_q	$\rho_{\hat{\nu}}$	E
5	3	54	—	0,805	$8{,}13 \cdot 10^{-7}$

Таблица 5

Значения $(\mathbf{S}_p)_m$ и $(\mathbf{E}_p)_m$ по результатам второго теста

T_0	T_1	T_2	T_3	T_4	T_5	$(\mathbf{S}_p)_m$	$(\mathbf{E}_p)_m$
4,653 %	15,902 %	16,596 %	15,814 %	16,358 %	30,677 %	6,56	0,73

Оценка (3) для динамического цикла может быть записана в виде

$$(\mathbf{S}_p^N)_m \approx 3^N\,\frac{q^*L^+ + 1}{q^*L^+ + 3^N}\,, \qquad (\mathbf{E}_p^N)_m \approx \frac{q^*L^+ + 1}{q^*L^+ + 3^N}\,, \tag{4}$$

где q^* — число многосеточных итераций, которые выполняются на грубых сетках. В табл. 6 представлены значения $(\mathbf{S}_p^N)_m$ и $(\mathbf{E}_p^N)_m$, полученные при распараллеливании первой краевой задачи для уравнения Пуассона.

Таблица 6

Оценка минимального ускорения и эффективности (динамический цикл)

L^+	q^*	$(\mathbf{S}_p^2)_m$	$(\mathbf{E}_p^2)_m$	$(\mathbf{S}_p^3)_m$	$(\mathbf{E}_p^3)_m$
3	2	4,20	0,47	5,73	0,21
	3	5,00	0,56	7,50	0,28
	4	5,57	0,62	9,00	0,33
	5	6,00	0,67	10,29	0,38
	6	6,33	0,70	11,40	0,42
4	2	4,76	0,53	6,94	0,26
	3	5,57	0,62	9,00	0,33
	4	6,12	0,68	10,67	0,40
	5	6,52	0,72	12,06	0,45
	6	6,82	0,76	13,24	0,49
5	2	5,21	0,58	8,03	0,30
	3	6,00	0,67	10,29	0,38
	4	6,52	0,72	12,06	0,45
	5	6,88	0,76	13,50	0,50
	6	7,15	0,79	14,68	0,54
6	2	5,57	0,62	9,00	0,33
	3	6,33	0,70	11,40	0,42
	4	6,82	0,76	13,24	0,49
	5	7,15	0,79	14,68	0,54
	6	7,40	0,82	15,86	0,59

Согласно (4), асимптотическое поведение эффективности имеет вид

$$(\mathbf{E}_p^N)_m \to 1 - \frac{3^N - 1}{q^* L^+} \quad \text{при} \quad \frac{3^N}{q^* L^+} \to 0.$$

Очевидно, что увеличение объема вычислений, необходимого для решения некоторой задачи, более сложной чем уравнение Пуассона, приведет к дальнейшему увеличению эффективности параллелизма: $q^* \uparrow \Rightarrow (\mathbf{E}_p^N)_m \to 1$.

5. Заключение. Многопроцессорный компьютер, предназначенный для параллельного исполнения УМГ, имеет простейшую архитектуру (фиксированное число процессоров при минимальном числе межпроцессорных связей). Особенности построения грубых сеток позволяют получить оптимальную загрузку процессоров. Эффективность параллелизма оценивается как

$$\frac{q^* L^+ + 1}{q^* L^+ + 3^N} < \mathbf{E}_p^N < 1.$$

Дальнейшее развитие связано с распараллеливанием сглаживающих итераций на самой мелкой сетке, однако в этом случае эффективность сильно зависит от времени обмена данными между процессорами.

СПИСОК ЛИТЕРАТУРЫ

1. *Мартыненко С.И.* Универсальная многосеточная технология для численного решения дифференциальных уравнений в частных производных на структурированных сетках // Вычислительные методы и программирование. 2000. **1**, № 1. 83–102.
2. *Voevodin V.V.* Mathematical foundations of parallel computing. Singapore: World Scientific Publ. Co. 1992.
3. *Немнюгин С.А., Стесик О.Л.* Параллельное программирование для многопроцессорных вычислительных систем. СПб.: БХВ-Петербург, 2002.
4. *Ortega J.M.* Introduction to parallel and vector solution of linear systems. New York–London: Plenum Press, 1988.
5. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. СПб: БХВ-Петербург, 2002.

Поступила в редакцию
14.11.2002