

УДК 519.684.4

МЕТОД РАСПАРАЛЛЕЛИВАНИЯ ПРОГОНКИ НА ГИБРИДНЫХ ЭВМ

А. Н. Быков¹, А. М. Ерофеев¹, Е. А. Сизов¹, А. А. Федоров¹

Рассматривается метод распараллеливания прогонки на гибридных ЭВМ с использованием ускорителей. Приводятся результаты численного исследования эффективности распараллеливания на задаче о бегущей тепловой волне. Статья рекомендована к публикации Программным комитетом форума “Суперкомпьютерные технологии в образовании, науке и промышленности” (НРС-2012; <http://agora.guru.ru/hpc2012>).

Ключевые слова: методика РАЗЭС-КП, метод прогонки, гибридные ЭВМ, арифметические ускорители.

1. Введение. Современные тенденции в разработке и создании суперкомпьютеров состоят в том, что качественно более высокая производительность достигается путем перехода к ЭВМ с множественными вычислительными процессами на основе параллельной обработки информации. Успешное использование большой потенциальной мощности таких ЭВМ для решения одной задачи возможно лишь после разработки прикладного программного обеспечения, учитывающего параллельную обработку информации. В Институте теоретической и математической физики (ИТМФ) РФЯЦ-ВНИИЭФ для решения научно-технических задач разработаны различные прикладные программы. Одна из них основана на эйлерово-лагранжевой методике РАЗЭС-КП численного решения трехмерных нестационарных задач газовой динамики с учетом теплопроводности [1]. Конечно-разностная аппроксимация дифференциальных уравнений, используемая в данной методике, приводит задачу нахождения численного решения к необходимости решения систем линейных алгебраических уравнений (СЛАУ) трехдиагонального вида. Для нахождения решения таких СЛАУ используется метод прогонки [2, 3].

Появление параллельных ЭВМ с арифметическими ускорителями сделало актуальной задачу реализации методов решения СЛАУ с помощью прогонки на параллельных ЭВМ с такой архитектурой. В настоящей статье обсуждается разработка и программная реализация на гибридных параллельных ЭВМ одного из способов реализации метода прогонки на параллельных ЭВМ, предложенного Н. Н. Яненко [2].

2. Метод прогонки. Для полноты изложения приведем сначала известный метод прогонки. В общем случае системы линейных алгебраических уравнений с трехдиагональной матрицей [3] имеют вид

$$a_i y_{i-1} + b_i y_i + c_i y_{i+1} = d_i, \quad i = 1, 2, \dots, N - 1, \tag{1}$$

$$y_0 = \kappa_1 y_1 + \mu_1, \quad y_N = \kappa_2 y_{N-1} + \mu_2. \tag{2}$$

Для численного решения систем с трехдиагональными матрицами применяется метод прогонки, который представляет собой вариант метода последовательного исключения неизвестных. Метод состоит в следующем. Решение системы (1), (2) ищется в виде

$$y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = 0, 1, \dots, N - 1, \tag{3}$$

где α_{i+1} и β_{i+1} — неизвестные коэффициенты, которые последовательно находятся от α_1, β_1 до α_N, β_N :

$$\alpha_{i+1} = \frac{-c_i}{b_i + a_i \alpha_i}, \quad \beta_{i+1} = \frac{d_i - a_i \beta_i}{b_i + a_i \alpha_i}, \quad i = 1, \dots, N - 1, \tag{4}$$

$$\alpha_1 = \kappa_1, \quad \beta_1 = \mu_1. \tag{5}$$

Вычисление коэффициентов $\alpha_{i+1}, \beta_{i+1}$ по формулам (4) и (5) называется прямым ходом прогонки. После того как прогоночные коэффициенты α_{i+1} и $\beta_{i+1}, i = 0, \dots, N - 1$, найдены, решение системы (1), (2) находится по рекуррентной формуле

$$y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = N - 1, N - 2, \dots, 0, \quad y_N = \frac{\kappa_2 \beta_N + \mu_2}{1 - \kappa_2 \alpha_N}. \tag{6}$$

¹ Всероссийский научно-исследовательский институт экспериментальной физики (ФГУП “РФЯЦ-ВНИИЭФ”), пр. Мира, 37, 607188, г. Саров; А. Н. Быков, нач. отдела, e-mail: ban3101@mail.ru, А. М. Ерофеев, ст. науч. сотр., e-mail: amekrym@gmail.com, Е. А. Сизов, мл. науч. сотр., e-mail: sizove@mail.ru, А. А. Федоров, математик, e-mail: fedoroffaa1@rambler.ru

Этот процесс называется обратным ходом прогонки.

На параллельных ЭВМ возникают трудности с “распараллеливанием” прогонки из-за рекуррентности формул. В настоящее время в методике РАМЗЕС-КП при счете на параллельных ЭВМ с распределенной памятью (без ускорителей) используется специальная версия параллельно-конвейерного метода распараллеливания прогонки по множеству линий сетки с автоматической адаптацией к быстродействию коммуникаций [1]. Такой подход прошел апробацию в течение ряда лет на параллельных ЭВМ с распределенной памятью. Известен другой способ распараллеливания прогонки, предложенный Н. Н. Яненко [2]. По ряду соображений было принято решение для гибридных ЭВМ попробовать способ Яненко. Ниже описан один из вариантов этого алгоритма.

Пусть каждому процессору выделено одинаковое количество точек $m = M/N$, где M — число точек, N — число процессоров, индексация глобальная. Таким образом, на отдельном процессоре с номером j решается лишь часть уравнений системы (1) с номерами от $(j-1)*m+1$ до $j*m$, где j — номер процессора. Обозначим y_{j*m} через z_j , $j = 0, \dots, N$, и представим искомые значения в виде

$$y_i = z_{j-1}u_i + z_jv_i + w_i, \quad i = 0, \dots, m, \quad (7)$$

где u, v, w — решения следующих систем уравнений:

$$\begin{cases} a_i u_{i-1} + b_i u_i + c_i u_{i+1} = 0, & \begin{cases} a_i v_{i-1} + b_i v_i + c_i v_{i+1} = 0, \\ v_{(j-1)*m} = 0, \quad v_{j*m} = 1, \end{cases} & \begin{cases} a_i w_{i-1} + b_i w_i + c_i w_{i+1} = d_i, \\ w_{(j-1)*m} = 0, \quad w_{j*m} = 0, \end{cases} \end{cases} \quad (8)$$

$$i = (j-1)*m+1, \dots, j*m-1, \quad j = 1, \dots, N.$$

Решения этих трех систем можно найти методом прогонки, причем независимо на каждом из процессоров. Будем называть эти решения предрешениями, а этот этап решения задачи — этапом нахождения предрешений.

В уравнения с номерами $j*m$ (т.е. в $a_{j*m}y_{j*m-1} + b_{j*m}y_{j*m} + c_{j*m}y_{j*m+1} = d_{j*m}$, $j = 0, \dots, N$) подставляем вместо y комбинации (7). Таким образом, получаем систему трехточечных уравнений для вычисления z_j , имеющую следующий вид:

$$\begin{aligned} A_j z_{j-1} + B_j z_j + C_j z_{j+1} &= D_j, \quad j = 1, \dots, N-1, \\ B_0 z_0 + C_0 z_1 &= D_0, \quad A_N z_{N-1} + B_N z_N = D_N, \quad B_0 = b_0 + c_0 u_1, \quad C_0 = c_0 v_1, \quad D_0 = d_0 - c_0 w_1, \\ A_j &= a_{j*m} u_{j*m-1}, \quad B_j = a_{j*m} v_{j*m-1} + b_{j*m} + c_{j*m} u_{j*m+1}, \quad C_j = c_{j*m} v_{j*m+1}, \\ D_j &= d_{j*m} - a_{j*m} w_{j*m-1} - c_{j*m} w_{j*m+1}, \quad j = 1, \dots, N-1, \\ A_N &= a_{N*m} u_{N*m-1}, \quad B_N = b_{N*m} + a_{N*m} v_{N*m-1}, \quad D_N = d_{N*m} - a_{N*m} w_{N*m-1}. \end{aligned} \quad (9)$$

Назовем этот этап решения задачи этапом нахождения решений на границе процессоров. Решение системы (9) и составляет нераспараллеливаемую часть данного алгоритма. Размерность этой системы уравнений равна количеству процессоров, а не количеству точек задачи. Систему будем решать методом встречных прогонок на всех процессорах. Окончательное решение восстанавливаем по формуле (7).

Алгоритм реализован в методике РАМЗЕС-КП на языке программирования ФОРТРАН с использованием функций библиотеки MPI. На рис. 1 представлена блок-схема, отражающая основные моменты реализации алгоритма.

3. Метод распараллеливания прогонки на ЭВМ с распределенной и с общей памятью. Необходимо учитывать различие в архитектуре центральных процессоров (ЦП) и арифметических ускорителей (АрУ) для этих типов ЭВМ. Программу необходимо разбивать на такие части, для которых лучше всего использовать традиционные процессоры, и на части, расчет которых оптимальнее всего проводить на АрУ [4]. Единый процесс прогонки достаточно сложно распараллелить из-за рекуррентности формул, поэтому мы будем распараллеливать множество прогонок. На гибридных ЭВМ каждая нить будет решать одну систему трехточечных уравнений.

На рис. 2 представлена блок-схема, отражающая основные моменты реализации алгоритма на гибридных ЭВМ, содержащих АрУ. Программа, реализующая этот алгоритм на гибридных ЭВМ, написана в методике РАМЗЕС-КП на языке программирования ФОРТРАН с использованием функций библиотеки MPI и библиотеки CUDA для языка ФОРТРАН. Эта программа отличается от исходного прототипа для ЦП тем, что в ней распараллеливается на АрУ этап нахождения предрешений и восстановления решения исходной системы по формулам (7).

4. Результаты тестирования. Тестирование разработанной программы проводилось на ядрах ЦП Intel Xeon X5670 и АрУ NVIDIA Tesla C2050. Характеристики ЦП и АрУ представлены в табл. 1.

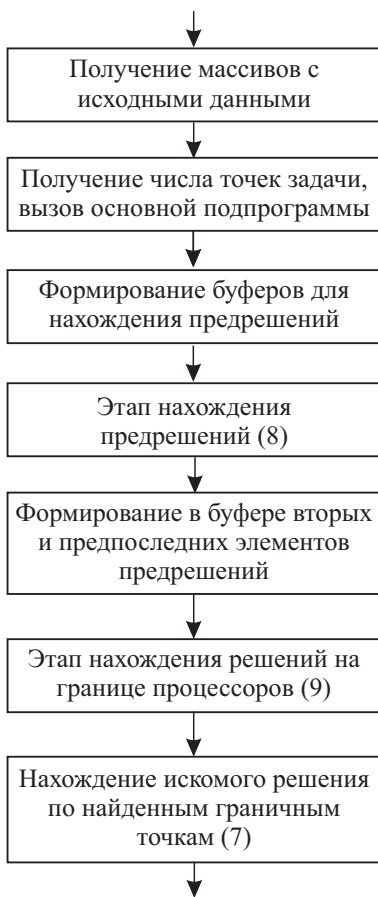


Рис. 1. Блок-схема алгоритма распараллеливания прогонки

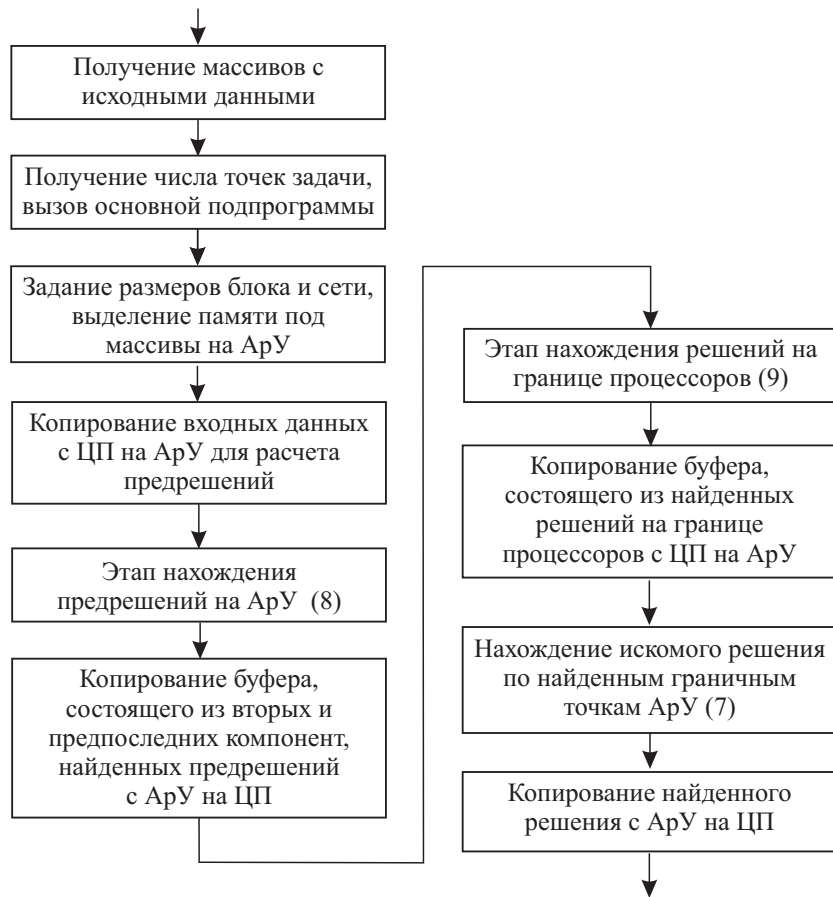


Рис. 2. Блок-схема алгоритма распараллеливания прогонки на АрУ

Характеристики ЦП и АрУ

Таблица 1

В	Параметр	Значение
1.	Центральный процессор Intel Xeon X5670:	
	— тактовая частота, ГГц	2.93
	— количество вычислительных ядер, шт.	6
2.	АрУ Nvidia Tesla C2050:	
	— интерфейс подключения	PCI-Express 2.0 x16
	— количество векторных ядер, шт.	14
	— тактовая частота, ГГц	1.15
	— теоретическая пиковая производительность (64-bit), Гфлоп/с	515.2
	— тип оперативной памяти	GDDR5
	— емкость оперативной памяти, Гбайт	3
	— тактовая частота оперативной памяти, МГц	750
— пропускная способность памяти, Гбайт/с	144	

Рассмотрим задачу о бегущей тепловой волне. На отрезке $0 \leq x \leq 1$ см, на котором находятся два вещества, в начальный момент времени задана температура $T(x, 0) = 0$ КэВ.

На границах формулируются следующие условия:

$$T(x, t) = \begin{cases} 10\sqrt{t}, & \text{при } x = 0 \text{ см} \\ 0, & \text{при } x = 1 \text{ см.} \end{cases}$$

Процесс описывается нелинейным уравнением теплопроводности

$$\frac{\partial E}{\partial t} = \frac{1}{\rho} \operatorname{div}(\chi \cdot \operatorname{grad} T)$$

со следующими характеристиками:

$$E = T, \quad \chi = \begin{cases} 0.5T^2, & \text{при } 0 \leq x \leq 0.5 \text{ см} \\ 5T^2, & \text{при } 0.5 \leq x \leq 1 \text{ см.} \end{cases}$$

Задача имеет аналитическое решение:

$$T(x, t) = \begin{cases} \sqrt{20(5t - x)} & \text{при } 5t \geq x & \text{и } 0 \leq x \leq 0.5 \text{ см,} \\ 0 & \text{при } 05t < x & \text{и } 0 \leq x \leq 0.5 \text{ см,} \\ \sqrt{2(50t - x - 4.5)} & \text{при } 50t - x - 4.5 \geq 0 & \text{и } 0.5 \leq x \leq 1 \text{ см,} \\ 0 & \text{при } 50t - x - 4.5 < 0 & \text{и } 0.5 \leq x \leq 1 \text{ см.} \end{cases}$$

Будем решать двумерную задачу в одномерной постановке с числом точек 10000*1600 и числом временных шагов 440. Задача решалась последовательно на 1, 15, 45, 90 АрУ. В этой задаче имеем 1600 систем трехточечных уравнений. Как уже отмечалось ранее, на АрУ каждая нить будет решать только одну систему.

Решение тестовой задачи с помощью этого алгоритма иллюстрируется на рис. 3.

Численное исследование эффективности распараллеливания проводилось в двух режимах: с использованием АрУ (CUDA + MPI) и без использования АрУ (MPI). Для обоих режимов в качестве характеристик расчета будем выделять:

- время работы алгоритма,
- ускорение $S_p = t_1/t_p$. Здесь t_1 – время решения задачи на одном процессоре, t_p – время решения задачи на p процессорах,
- эффективность $E_p = \frac{t_1}{pt_p} 100\%$.

Результаты представлены в табл. 2 и 3.

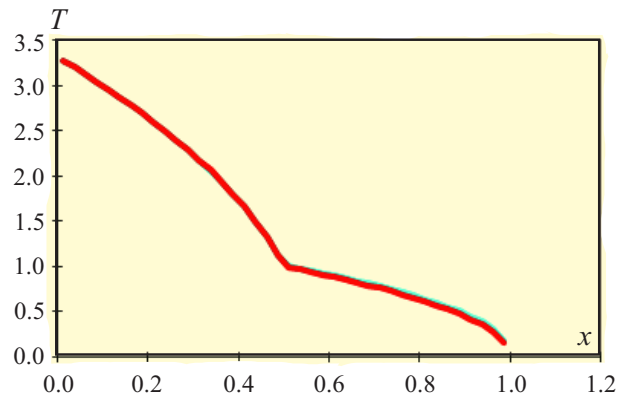


Рис. 3. Графики решений тестовой задачи на разном числе ускорителей и точного решения на момент времени 0.111 сек.

Таблица 2
Эффективность работы алгоритма на ЭВМ без использования ускорителей

	p (число ядер)			
	1	15	45	90
t , сек.	4208.12	352.01	164.01	123.81
S_p	1	11.95	25.66	33.99
E_p , %	100	79.7	57.02	37.77

Таблица 3
Эффективность работы алгоритма с использованием АрУ

	p (число ускорителей)			
	1	15	45	90
t , сек.	1139.69	82.83	33.38	27.72
S_p	1	13.75	34.12	41.09
E_p , %	100	91.67	75.83	45.65

Ускорение программы в расчетах с использованием АрУ по сравнению с расчетами без использования АрУ представлено в табл. 4, где t_{GPU} и t_{CPU} – время работы программы с и без АрУ. Отношение этих времен показывает преимущество использования АрУ.

Из представленных данных можно сделать вывод о том, что использование ускорителей позволяет уменьшить время работы программы до 5 раз. Для исследования масштабируемости программы было проведено профилирование программ, результаты которого представлены в табл. 5. Разобьем программу на части: арифметика ЦП (вычисления, проводимые на процессоре), коммуникации между процессорами, передача данных на АрУ и обратно и арифметика АрУ (вычисления на ускорителе). Длительность

Таблица 4
Сравнение результатов реализации на ускорителях и без них

p	t_{GPU} , сек.	t_{CPU} , сек.	t_{CPU}/t_{GPU}
1	1139.69	4208.12	3.69
15	82.83	352.01	4.25
45	33.38	164.01	4.91
90	27.72	123.81	4.47

Таблица 5
Длительность работы частей программы

Работа частей программы, %	p (число ускорителей)		
	15	45	90
Общее время	100	100	100
Арифметика ЦП	0.36	0.95	1.37
Коммуникации MPI	1.60	12.01	34.23
Обмен данными между ЦП и АрУ	54.10	47.25	35.25
Арифметика АрУ	43.94	39.80	28.85

работы частей программы на разном числе ускорителей представлена в табл. 5. Основное время занимает обмен данными между ЦП и АрУ, но при включении данного алгоритма в РАМЗЕС-КП отпадет необходимость большинства перекачек данных, так как они уже будут храниться в памяти АрУ. Кроме того, с увеличением числа ускорителей возрастает время коммуникаций между процессорами, что приводит к снижению эффективности распараллеливания.

5. Заключение. В данной работе выполнена программная реализация на гибридных многопроцессорных ЭВМ одного из способов реализации метода прогонки на параллельных ЭВМ, предложенного Н. Н. Яненко. Работоспособность программы была проверена на задаче о бегущей тепловой волне, получено хорошее согласие с аналитическим решением. Разработанная программа позволила получить ускорение до 5 раз при использовании АрУ.

В дальнейшем предполагается

- реализовать параллельно-конвейерный алгоритм для метода прогонки на этапе вычисления решения на границах процессоров;
- применить двумерную декомпозицию, что уменьшит число процессоров в прогонке и должно повысить эффективность распараллеливания;
- внедрить разработанные программы в РАМЗЕС-КП, что позволит перенести расчет коэффициентов трехточечных уравнений на АрУ и избавиться от двух этапов обменов данными между ЦП и АрУ;
- реализовать на гибридных ЭВМ параллельно-конвейерный метод, используемый сейчас в методике РАМЗЕС-КП.

СПИСОК ЛИТЕРАТУРЫ

1. Быков А.Н., Веселов В.А., Воронин Б.Л., Ерофеев А.М. Методика РАМЗЕС-КП для расчета пространственных движений многокомпонентных теплопроводных сред в эйлерово-лагранжевых координатах // Труды РФЯЦ-ВНИИЭФ. Вып. 13. Саров, 2008. 50–57.
2. Яненко Н.Н., Коновалов А.Н., Бугров А.Н., Шустов Г.В. Об организации параллельных вычислений и “распараллеливании” прогонки // Численные методы механики сплошной среды. 9, № 7. Новосибирск, 1978. 139–146.
3. Самарский А.А. Введение в численные методы. М.: Наука, 1982.
4. Рыбакин Б.П. Параллельное программирование для графических ускорителей. М.: НИИСИ РАН, 2011.

Поступила в редакцию
26.04.2013