

3 Зюзьков В. М. Последовательность  $Fibonacci(n) \bmod n$  // Вестник Томского государственного университета. Математика и механика. 2013, № 4(24). С. 15–23.

4 Wolfram Mathematica [Электр. ресурс].– URL: <http://www.wolfram.com/mathematica>

## Реализация метода IADE-RB-CG в системе фрагментированного программирования LuNA

DOI 10.17223/978-5-7511-2389-5/10

С.Е. Киреев<sup>1</sup>, В.А. Перепелкин<sup>1</sup>, А.А. Ткачёва<sup>1,2</sup>

<sup>1</sup>Институт вычислительной математики и математической геофизики  
СО РАН, Новосибирск

<sup>2</sup>Новосибирский государственный университет, Новосибирск  
{kireev, perepelkin, tkacheva}@ssd.ssc.ru

*Целью работы является апробация системы фрагментированного программирования LuNA на конкретном численном алгоритме, реализующем решение модельной задачи с помощью модификации метода IADE. В работе представлен фрагментированный алгоритм решения задачи. Выполнено сравнительное тестирование нескольких реализаций алгоритма на кластере. Показано преимущество использования в системе LuNA средств прямого управления.*

Ключевые слова: технология фрагментированного программирования, система LuNA.

Разработка больших численных моделей для суперкомпьютеров является сложной задачей ввиду необходимости выполнить декомпозицию данных и вычислений, а также обеспечить корректную работу взаимодействующих процессов, настройку параллельной программы на доступные ресурсы, эффективную (по времени и ресурсам) работу программы во времени. Решение этих проблем требует квалификации в области системного параллельного программирования, что затрудняет реализацию больших численных моделей их разработчиками. Для преодоления этих трудностей прибегают к средствам автоматизации параллельного программирования, таким как системы параллельного программирования. В таких системах описание модели или численного алгоритма осуществляется на высоком уровне абстракции, а требуемая параллельная программа генерируется автоматически. Автоматическая генерация эффективных

(по времени, памяти и т.п.) программ в настоящий момент является сложной научной проблемой вследствие алгоритмической трудно-решаемости таких подзадач, как распределение ресурсов, планирование вычислений.

Одной из таких систем является система LuNA [1] (от Language for Numerical Algorithms), разрабатываемая в Институте вычислительной математики и математической геофизики СО РАН. Она ориентирована на эффективную параллельную реализацию больших численных моделей для суперкомпьютеров. Эта система поддерживает технологию фрагментированного программирования, теоретическим базисом которой является теория структурного синтеза параллельных программ на вычислительных моделях [2]. В основе технологии лежит подход, при котором прикладной численный алгоритм представляется в явной крупноблочной-параллельной форме и сохраняет такую фрагментированную структуру во время исполнения. Это позволяет динамически распределять данные и вычисления по вычислительным узлам мультикомпьютера, а также варьировать порядок выполнения операций алгоритма в рамках имеющихся информационных зависимостей. Таким образом, система обеспечивает динамические свойства исполнения алгоритма, настройку на доступные ресурсы и т.д. Существенной особенностью подхода является возможность влиять на ход исполнения алгоритма с помощью средств ручного управления исполнением алгоритма для достижения высокой эффективности реализации алгоритма.

В настоящей работе рассматривается реализация конкретного приложения – одной из модификаций метода IADE в системе фрагментированного программирования LuNA с использованием средств ручного управления исполнением алгоритма. Внимание уделяется вопросу эффективности полученной реализации.

Метод IADE (Iterative Alternating Decomposition Explicit) уже долгое время используется для решения многомерных параболических задач [3–5]. Имея 2-й порядок точности по времени и 4-й по пространству, он является более эффективным и имеет более высокую скорость сходимости, чем классические итеративные методы 4-го порядка [4]. Метод IADE использует схему с расщеплением, дающую в одномерном случае двухстадийный итерационный процесс. На первом полушаге  $i$ -го шага по времени вычисляется приближенное решение  $x^{i+\frac{1}{2}}$ , используя значения  $x^i$ , а на втором полу-

шаге на его основе вычисляются новые значения  $x^{i+1}$ . В базовом последовательном алгоритме значение  $x_j^{i+\frac{1}{2}}$  зависит от  $x_{j-1}^{i+\frac{1}{2}}$ , тогда как значение  $x_j^{i+1}$  зависит от  $x_{j+1}^{i+1}$  [3]. Чтобы устранить эти зависимости, были разработаны различные стратегии распараллеливания [5]. В данной работе используется красно-черное (Red-Black) упорядочение элементов, дающее на каждом шаге две прогонки для четных и нечетных элементов. Кроме того, чтобы ускорить сходимость, была использована модификация схемы на основе метода сопряженных градиентов (CG) [5].

В качестве примера реализации метода IADE-RB-CG рассматривается решение одномерного уравнения теплопроводности. Процесс решения представляет собой последовательность временных шагов, на каждом из которых решается система линейных уравнений с одними и теми же коэффициентами, но разными правыми частями.

Чтобы получить фрагментированный алгоритм, выполняется разделение данных и вычислений на фрагменты, причем размер фрагментов должен быть параметром алгоритма. Таким образом, вся область моделирования делится на подобласти одинакового размера. Каждая подобласть содержит красные и черные узлы сетки, которые сгруппированы в отдельные фрагменты данных с целью уменьшить в алгоритме количество зависимостей по данным. Например, вектор решения  $x$  размером  $M$ , разделенный на  $m$  частей, во фрагментированном алгоритме представлен фрагментами данных  $xR_0, xR_1, \dots, xR_{m-1}$ , содержащих красные элементы, и фрагментами данных  $xB_0, xB_1, \dots, xB_{m-1}$ , содержащих черные элементы. Размер каждого фрагмента данных  $S \approx \frac{M}{2m}$ .

Фрагментированный алгоритм метода IADE-RB-CG представлен на рис. 1 в виде графа зависимостей. Круги обозначают фрагменты данных, а закругленные прямоугольники – фрагменты вычислений. Черные круги – входные фрагменты данных, белые – выходные, а серые – промежуточные для данного шага по времени. Фрагмент кода «init\_f» вычисляет элементы вектора правых частей во фрагментах данных  $fR^n, fB^n$ . Фрагмент кода «solve» вычисляет решения для следующего шага по времени. Фрагменты данных в левой части рис. 1а содержат коэффициенты, используемые для решения. Фрагменты данных  $yR, yB$  содержат решение на полушаге  $x^{i+\frac{1}{2}}$ . Эlemen-

ты алгоритма, относящиеся к фазе инициализации, здесь и далее опущены. Рисунок 1б демонстрирует фрагментированный алгоритм, реализующий фрагмент кода «solve». Он содержит итеративный процесс, выполняющийся до сходимости. Фрагменты данных  $rR$ ,  $rB$  и  $zR$ ,  $zB$  содержат векторы, используемые в части модификации схемы на основе метода сопряженных градиентов.

На рисунке 2 представлен фрагментированный алгоритм одной итерации метода IADE-RB-CG (фрагмент кода «compute» на рис. 1б). Круги здесь обозначают множества из  $m$  фрагментов данных, соответствующих  $m$  подобластям. Прямоугольники также обозначают соответствующие множества фрагментов вычислений. Так, прямоугольник « $iadeR$ » обозначает множество из  $m$  фрагментов вычислений, реализованных одним и тем же фрагментом кода « $iadeR$ ».  $j$ -й фрагмент вычислений принимает на вход  $j$ -е фрагменты данных из входных множеств фрагментов данных и формируют на выходе  $j$ -е фрагменты данных из выходных множеств. То же самое верно и для « $iadeB$ », « $cgR$ » и « $cgB$ ». Взаимодействия между подобластями при этом не происходит, что позволяет выполнять фрагменты вычислений параллельно в случае наличия вычислительных ресурсов.

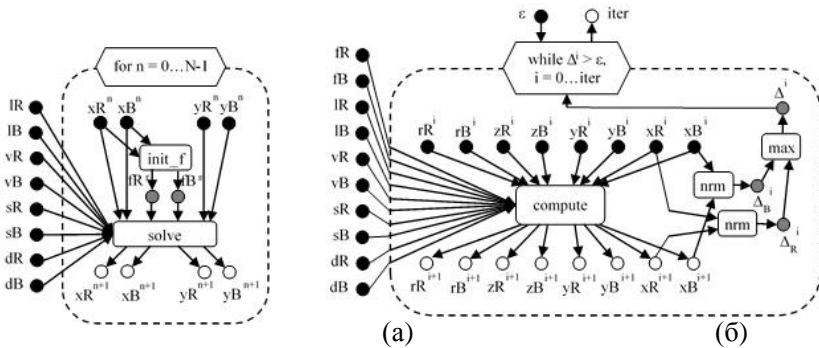


Рис. 1. Фрагментированный алгоритм метода IADE-RB-CG (а) и фрагментированный алгоритм, реализующий фрагмент кода "solve" (б)

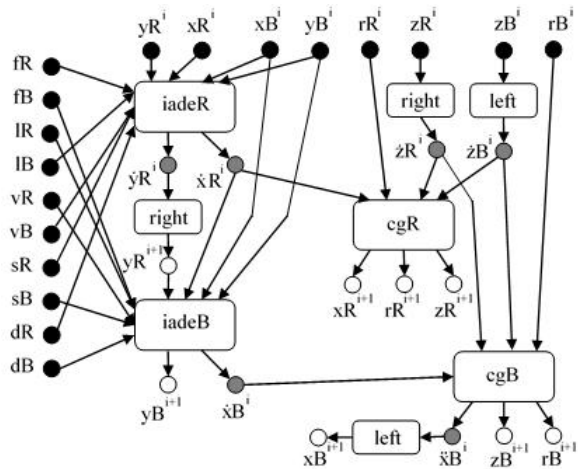


Рис. 2. Фрагментированный алгоритм  $i$ -й итерации метода IADE-RB-CG

Взаимодействие между подобластями происходит только во фрагментах кода «left» и «right», в которых происходит обмен границами (рис. 3). Так как все подобласти, кроме, возможно, последней, имеют четное число элементов, только по одному элементу серого цвета при обмене передается в правую подобласть (рис. 3а), и только по одному элементу черного цвета передается в левую подобласть (рис. 3б).

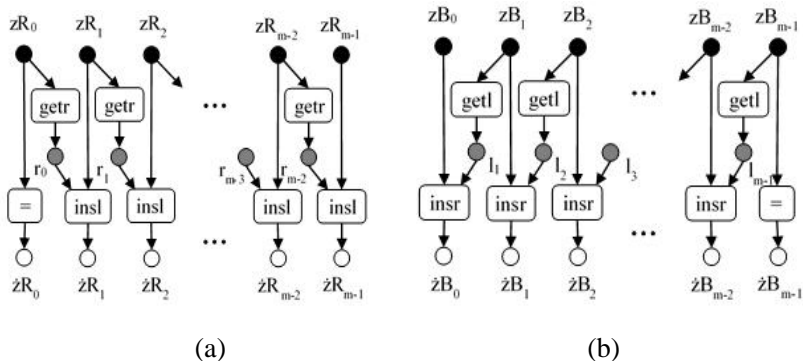


Рис. 3. Фрагментированные алгоритмы передачи границ подобластей: правой (а) и левой (б)

Разработанный алгоритм был реализован в системе фрагментированного программирования LuNA. Программа в системе LuNA – это буквально запись представленного в виде графа алгоритма в текстовой форме. Ниже приведен пример LuNA-подпрограммы, соответствующей алгоритму, представленному на рис. 3а.

```
sub send_right(#in int m, name sizeR, name zR_in, #out name
zR_out)
{ df r;
  for j=0..m-2
    get_R_right_border(#in sizeR[j],zR_in[j], #out r[j]);
    copy(#in zR_in[0], #out zR_out[0]);
  for j=1..m-1
    set_R_left_border(#in sizeR[j],zR_in[j],r[j-1], #out zR_out[j]);
}
```

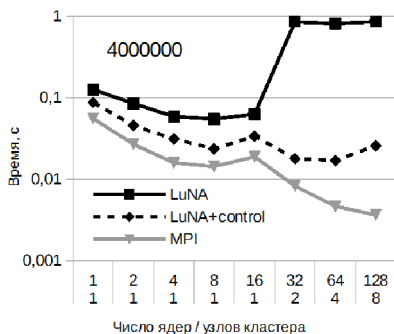
Было выполнено сравнительное тестирование трех реализаций разработанного фрагментированного алгоритма:

1) реализация в системе LuNA, основанная только на исходном фрагментированном алгоритме, в которой большинство решений по распределению ресурсов и управлению принимаются автоматически;

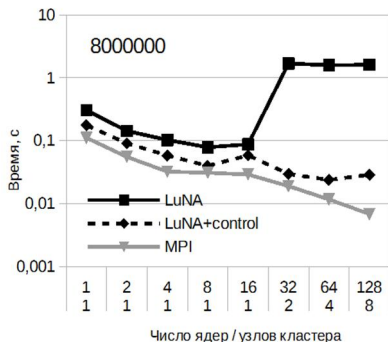
2) реализация в системе LuNA, дополненная написанной вручную управляющей программой, которая управляет распределением ресурсов и порядком срабатывания фрагментов вычислений;

3) ручная реализация с использованием библиотеки MPI.

Тестирование проводилось на кластере МВС-10П (МСЦ РАН, г. Москва). Результаты сравнения представлены на рис. 4. Приведено среднее время выполнения одной итерации цикла while (рис. 1б) за 20 шагов моделирования по времени при различных размерах задачи на различном числе ядер. Видно, что при использовании только автоматических средств система LuNA не справляется с эффективным исполнением алгоритма и имеет большие накладные расходы, что является проблемой всех систем такого типа. Но использование средств ручного управления позволяет достичь производительности, близкой к производительности ручной реализации в MPI.



(а)



(б)

Рис. 4. Сравнение производительности различных реализаций алгоритма IADE-RB-CG: размер задачи 4000000 узлов (а) и 8000000 узлов (б)

Был разработан фрагментированный алгоритм метода IADE-RB-CG. Алгоритм был реализован с использованием библиотеки MPI и в системе LuNA. Результаты тестирования показали, что использование средств ручного управления распределением ресурсов и порядком исполнения позволяет достигнуть производительности LuNA-программы, близкой к производительности ручной реализации в MPI.

### Литература

1. *Malyshkin V.E., Perepelkin V.A.* LuNA Fragmented Programming System, Main Functions and Peculiarities of Run-Time Subsystem // PaCT-2011 proceedings, Springer, LNCS 6873 (2011), P. 53–61.
2. *Вальковский В.А., Малышкин В.Э.* Синтез параллельных программ и систем на вычислительных моделях. Новосибирск: Наука, Сибирское отделение, 1988. 129 с.
3. *Sahimi M.S., Ahmad A., Bakar A.A.* The Iterative Alternating Decomposition Explicit (IADE) method to solve the heat conduction equation // International Journal of Computer Mathematics. 1993. Vol. 47. P. 219–229.
4. *Mansor N. Abu, Zulkifle A.K., Alias N., Hasan M.K., Boyce M.J.N.* The Higher Accuracy Fourth-Order IADE Algorithm // Journal of Applied Mathematics. 2013. Vol. 2013. P. 1–13.
5. *Alias Norma, Sahimi M.S., Abdullah A.R.* Parallel strategies for the iterative alternating decomposition explicit interpolation-conjugate gradient method in solving heat conductor equation on a distributed parallel computer systems // Proceedings of The 3rd International Conference On Numerical Analysis in Engineering. 2003. Vol. 3. P. 31–38.