

РЕАЛИЗАЦИЯ ИТЕРАЦИОННЫХ МЕТОДОВ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ В ЗАДАЧАХ МАТЕМАТИЧЕСКОЙ ФИЗИКИ НА РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

© 2016 г. И.И. Левин, А.И. Дордопуло, А.В. Пелипец

Научно-исследовательский институт многопроцессорных вычислительных систем

Южного федерального университета (Таганрог, Российская Федерация)

(347928 Таганрог, ул. Чехова, 2, ГСП-284, г. Таганрог, Ростовская область),

E-mail: levin@mvs.tsure.ru, scorpio@mvs.tsure.ru, pelipets@mail.ru

Поступила в редакцию: 12.07.2016

В статье рассматриваются характерные особенности реализации итерационных методов решения систем линейных уравнений в задачах математической физики на параллельных вычислительных системах, которыми являются геометрическая декомпозиция расчетной области и распараллеливание данных внутри последовательно выполняемых процессором итераций с интенсивным информационным обменом между процессорами и памятью. Стандартные методы реализации итерационных методов решения систем линейных уравнений при множественных транзакциях с памятью и межпроцессорных обменах, существенно снижающих реальную производительность, дополнительно требуют от вычислительной системы наличия большого числа физических линий связи для реализации сложных топологий и иерархических схем хранения данных. Выходом является использование многопроцессорных систем с реконфигурируемой архитектурой, позволяющее адаптировать свою архитектуру под структуру итерационных алгоритмов математической физики путем распараллеливания по итерациям. В статье рассмотрена реализация метода Якоби для решения краевой задачи Дирихле для уравнения Лапласа на реконфигурируемой вычислительной системе, на примере которой показано сокращение количества внешних каналов обмена как одного из наиболее критических ресурсов реконфигурируемой вычислительной системы.

Ключевые слова: реконфигурируемые вычислительные системы, программируемые логические интегральные схемы, численные методы математической физики, распараллеливание по итерациям, вычислительный конвейер.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Левин И.И., Дордопуло А.И., Пелипец А.В. Реализация итерационных методов решения систем линейных уравнений в задачах математической физики на реконфигурируемых вычислительных системах // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2016. Т. 5, № 4. С. 5–18. DOI: 10.14529/cmse160401.

Введение

В процессе численного решения модельных уравнений математической физики часто возникает необходимость нахождения неизвестных в системах линейных алгебраических уравнений (СЛАУ) высокого порядка (для числа уравнений $n > 1000$). В данных условиях прямые методы, такие, как метод исключений Гаусса, могут быть неэффективны из-за высоких требований к оперативной памяти, большой вычислительной трудоемкости и накопления ошибок округления [1]. Поэтому предпочтительными становятся итерационные методы, которые позволяют найти приближенное решение СЛАУ за конечное число итераций, количество которых зависит от требуемой точности.

Параллельная форма реализации итерационных методов решения СЛАУ основана на концепции геометрического параллелизма, при котором исходная расчетная область, описывающая состояние физической среды в определенных точках пространства, разбивается по одному или двум пространственным направлениям с последующим распределением подобластей между процессорами. Ключевой особенностью реализаций итерационных методов на параллельных многопроцессорных системах является то, что итерации выполняются последовательно, а распараллеливание задачи происходит по данным на уровне вычислений внутри итерации [2]. При этом каждый процессор обрабатывает свою часть области в течение всех итераций, обмениваясь результатами с другими процессорами и с внешней оперативной памятью.

Однако необходимость интенсивных взаимодействий типа процессор-процессор и процессор-память сопряжена с большими накладными расходами, порой сводящими на нет весь эффект от распараллеливания. Особенно актуальной эта проблема становится по мере увеличения масштабов многопроцессорных систем и количества компонентов, входящих в них [3].

На параллельных вычислительных системах с «жесткой» архитектурой способы решения данной проблемы, в основном, сводятся к выбору подходящей топологии и организации системы хранения данных в виде сложной иерархической структуры. Это, в свою очередь, порождает проблему дефицита каналов памяти для обеспечения доступа к различным уровням иерархии.

Для решения указанных проблем активно проводятся исследования, связанные с поиском способов ускорения решения СЛАУ итерационными методами на высокопроизводительных системах. Работа [4] описывает применение гибридной версии технологий параллельного программирования OpenMP–MPI для сокращения накладных расходов вычислений в задачах механики жидкости методом конечных элементов. В статье [5] делается обзор опыта применения графических ускорителей для решения итерационными методами разреженных систем линейных уравнений, приводится сравнение эффективности реализаций алгоритмов на GPU NVIDIA TESLA с CPU Intel Xeon, использующим специализированные библиотеки линейной алгебры Intel MKL. Публикация [6] посвящена асинхронным итерационным алгоритмам, позволяющим уменьшить коммуникационные расходы по сравнению с другими алгоритмами, за счет устранения простаивающих узлов из-за ожидания синхросообщений. Приведенные экспериментальные результаты показывают, что модифицированный релаксационный метод Саусвелла уменьшает затраты на коммуникации по сравнению с другими асинхронными итерационными методами и классическим синхронным методом Якоби. Множество работ касаются использования гетерогенных вычислительных архитектур в решении разреженных СЛАУ итерационными методами. В частности, вопросам построения адаптированных итерационных алгоритмов, эффективно распределяющих вычислительную нагрузку в системах CPU-GPU, посвящены статьи [7–9].

В настоящее время альтернативой традиционным многопроцессорным архитектурам при решении задач математической физики все чаще выступают реконфигурируемые вычислительные системы (PBC), построенные на основе программируемых логических интегральных схем (ПЛИС) [10]. Здесь также проблема дефицита внешних каналов обмена является весьма актуальной, поскольку удельное количество пользовательских выводов относительно логической емкости кристалла снижается в каждом поколении

ПЛИС [11–13]. Применение различных методов распараллеливания по данным лишь усугубляет эту проблему.

«Гибкая» архитектура РВС позволяет, не меняя исходный алгоритм задачи, адаптировать вычислитель под ее информационную структуру, что дает возможность организовать для матричных задач не только геометрическое распараллеливание (по данным), но и временное (по итерациям). Это ведет к существенному сокращению количества внешних каналов обмена, являющихся одним из наиболее критических ресурсов реконфигурируемой вычислительной системы.

Целью данной работы является разработка и исследование алгоритма решения на РВС систем линейных алгебраических уравнений методом Якоби для матриц специального вида в задачах математической физики. В разделе 1 настоящей статьи дается описание преобразования краевой задачи Дирихле для уравнения Лапласа для решения ее в структурно-процедурной форме. В разделе 2 анализируются практические результаты решения на РВС краевой задачи Дирихле для уравнения Лапласа, приводятся сравнение полученных результатов с реализацией данной задачи на параллельной кластерной системе. В заключении обсуждаются перспективы применения реконфигурируемых систем для решения систем линейных уравнений итерационными методами в задачах математической физики и указываются направления дальнейших исследований.

1. Преобразование в структурно-процедурную форму краевой задачи Дирихле для уравнения Лапласа

Рассмотрим в качестве примера решение задачи Дирихле для уравнения Лапласа, которое используется для описания различных стационарных (установившихся) физических процессов [14]. Пусть требуется решить задачу Дирихле для уравнения Лапласа, имеющего общий вид $\Delta u(x, y, z) = 0$, которое задано в двумерном пространстве дифференциальной формой

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (1)$$

где $u(x, y)$ – функция с граничными условиями первого рода, значение которой необходимо вычислить.

Тогда в результате дискретизации уравнения (1) методом конечных разностей на квадратной координатной сетке ϖ согласно пятиточечному шаблону получим разностную (алгебраическую) форму

$$\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{h^2} = 0, \quad (2)$$

где $i, j = 1, 2, \dots, N-1$;

h – постоянный шаг сетки, $hN = 1$.

Если построить сетку с постоянным шагом $h = \frac{1}{10}$, то поскольку порядок системы равен $n = (N-1)^2$, получим систему из 81 уравнения, которую можно последовательно решить методом Якоби с помощью формулы общего вида

$$u_{ij}^{(k+1)} = \frac{1}{4}(u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k), \quad (3)$$

где k – номер итерации.

Общая формула (3) эквивалентна системе линейных уравнений

$$\begin{cases} u_{11}^{k+1} = \frac{1}{4}(u_{21}^k + u_{01}^k + u_{12}^k + u_{10}^k) \\ u_{12}^{k+1} = \frac{1}{4}(u_{22}^k + u_{02}^k + u_{13}^k + u_{11}^k) \\ \dots \\ u_{99}^{k+1} = \frac{1}{4}(u_{10,9}^k + u_{89}^k + u_{9,10}^k + u_{98}^k) \end{cases} \quad (4)$$

Исходя из того, что значения функции $u(x,y)$ на границе рассматриваемой области известны заранее, система (4) может быть преобразована к виду

$$\begin{cases} u_{11}^{k+1} = \gamma(u_{21}^k + c_{01} + u_{12}^k + c_{10}) \\ u_{12}^{k+1} = \gamma(u_{22}^k + c_{02} + u_{13}^k + u_{11}^k) \\ \dots \\ u_{99}^{k+1} = \gamma(c_{10,9} + u_{89}^k + c_{9,10} + u_{98}^k) \end{cases}, \quad (5)$$

где c_{ij} – известные граничные значения функции;
 γ – константное значение множителя, $\gamma = \frac{1}{4}$.

Итерационный процесс вычислений завершается при выполнении условия

$$|u_{ij}^{k+1} - u_{ij}^k| \leq \varepsilon, \quad (6)$$

где ε – это требуемая точность.

Система (5) позволяет отобразить процесс вычислений неизвестных u_{ij} на $k+1$ -ой итерации в виде полного информационного графа β , представленного в параллельном виде на рис. 1. Здесь информационные вершины соответствуют множеству исходных данных и результатов, операционные вершины изображают операции с данными, а дуги показывают направления информационного обмена между вершинами.

Особенности граничных условий первого рода определяют количество и порядок информационных дуг, соединяющих информационные вершины-источники с операционными вершинами-приемниками, а именно: адресные вершины $\langle u_{11}^k, u_{1,N-1}^k, u_{N-1,1}^k, u_{N-1,N-1}^k \rangle$, соответствующие точкам, примыкающим к границе в углах области решения, соединяются с операционными вершинами-приемниками двумя дугами.

Адресные вершины u_{ij}^k , соответствующие точкам, примыкающим к границе области (исключая угловые точки $\langle u_{11}^k, u_{1,N-1}^k, u_{N-1,1}^k, u_{N-1,N-1}^k \rangle$), соединяются с операционными вершинами-приемниками тремя дугами. Адресные вершины u_{ij}^k , соответствующие остальным точкам внутри области, имеют по четыре исходящих дуги. Адресные вершины c_{ij} , соответствующие граничным точкам области решения (заданные значения функции), имеют по одной исходящей дуге.

Схема информационных зависимостей вершин, соответствующих узлам сетки внутри области решения, проиллюстрирована на рис. 2.

В пределах одной итерации информационный граф параллельной формы данной задачи можно рассматривать как кортеж информационно-независимых подграфов $\langle \alpha_1, \alpha_2, \dots, \alpha_n \rangle$. Информационная независимость подграфов α_i определяется отсутствием дуг, соединяющих операционные вершины двух или более подграфов в кортеже.

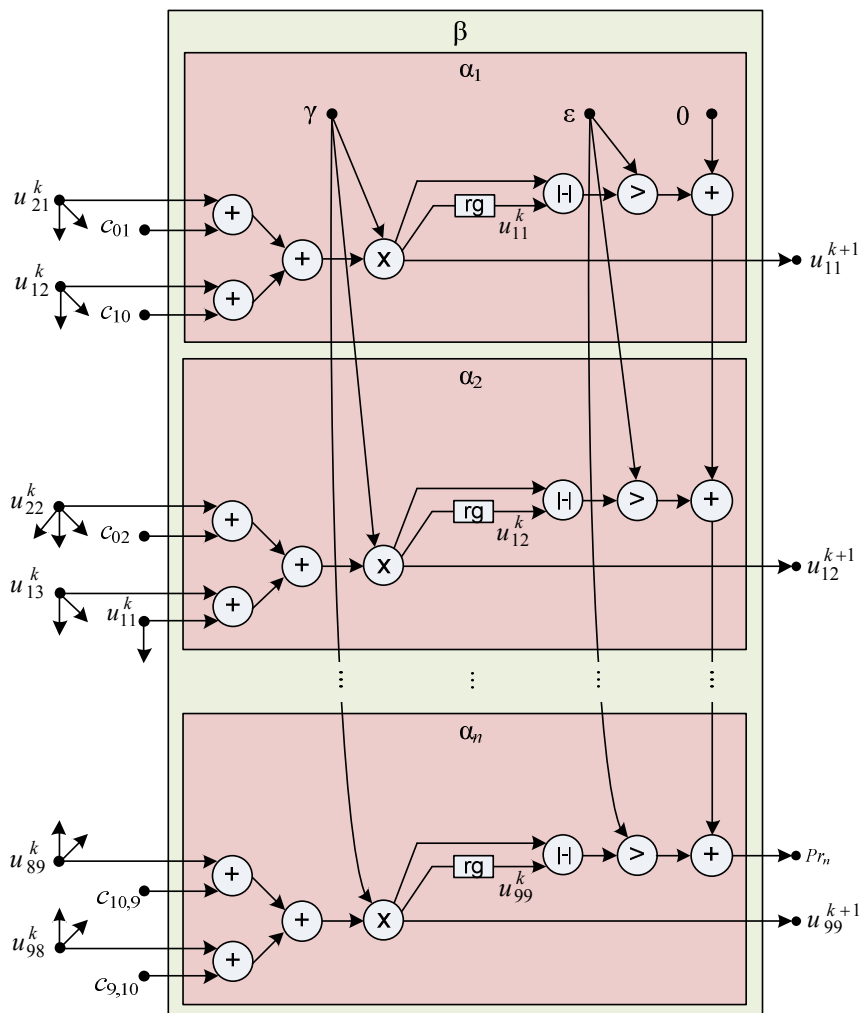


Рис. 1. Параллельная форма метода Якоби решения уравнения Лапласа

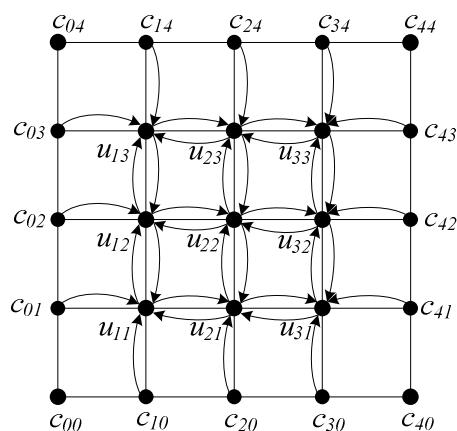


Рис. 2. Схема информационных зависимостей вершин, соответствующих узлам сетки с шагом $h = \frac{1}{4}$

В соответствии с пятиточечным шаблоном каждый из n изоморфных информационных подграфов α_i на $(k+1)$ -й итерации реализует одно умножение и три сложения операндов для получения на выходе очередного узлового значения функции u_{ij}^{k+1} . Кроме этого, операции модуля разности, сравнения и суммирования реализуют процедуру проверки условия сходимости (6), результатом которой является предикат Pr_n . Управляющая программа останавливает вычислительный процесс при обнаружении предиката, значение которого говорит о выполненном условии сходимости.

Общее число операционных вершин полного графа β параллельной формы для решения n систем уравнений будет составлять $7n$. Если бы количество итераций, которые необходимо пройти до завершения процесса вычислений, было известно заранее (например, m), то алгоритм мог быть представлен в абсолютно-параллельной форме с числом базовых подграфов, равным $n \cdot m$. Но поскольку необходимое количество итераций изначально не определено и зависит от выполнения условий сходимости, то число информационных подграфов в параллельной форме равно размерности матрицы n , а количество реализованных ими итераций может быть произвольным. Промежуточные результаты итерации k , являющиеся исходными данными для итерации $(k+1)$, должны записываться и считываться из внешней оперативной памяти.

Количество аппаратного ресурса для реализации параллельной формы задачи рассчитывается по формуле

$$A = n \cdot N(\alpha_i), \quad (7)$$

где $N(\alpha_i)$ – количество ресурса, необходимого для реализации базового информационного подграфа α_i .

Реализация параллельной формы метода Якоби (рис. 1) возможна при наличии в вычислительной системе количества процессоров, достаточного для параллельной обработки всех узлов координатной сетки ϖ . Из (7) следует, что структурная реализация параллельной формы задачи требует наличия $7n$ процессорных элементов и $5n$ внешних каналов обмена. Для систем уравнений больших размерностей необходимый ресурс может превосходить параллельные возможности аппаратуры. В этом случае необходимо перейти от параллельной к структурно-процедурной (кадровой) форме вычислений и провести редукцию производительности путем сокращения числа подграфов, операционных и адресных вершин. При переходе к кадровой форме в результате векторизации подмножества информационных вершин представляются в виде адресных вершин-массивов G , функция которых состоит в генерации потока адресов обращения к памяти, где хранятся массивы обрабатываемых данных (результатов), а также вывод операндов в вычислительные устройства.

Пусть L – общий объем аппаратного ресурса вычислительной системы, такой, что $L < A$. Тогда реализация структурно-процедурной кадровой формы задачи может быть осуществлена путем редукции производительности по числу базовых подграфов с коэффициентом

$$r = \left\lfloor \frac{A}{L} \right\rfloor. \quad (8)$$

Если аппаратного ресурса вычислительной системы достаточно для структурной реализации не более, чем r базовых подграфов α_i , то кадровая форма задачи может быть получена путем сокращения полного графа β в $\frac{n}{r}$ раз. Степень редукции производи-

тельности по подграфам [15] определяется возможностью структурной реализации определенного числа базовых подграфов в теле кадра. Предельным случаем редукции по подграфам является кадр Q , состоящий из одного базового подграфа α (рис. 3). При этом вычислительный процесс расчета узловых значений функции на текущей итерации ($k+1$) будет представлять собой циклическое (от 1 до n) выполнение кадра с получением на каждом шаге цикла очередного приближения u_{ij}^{k+1} .

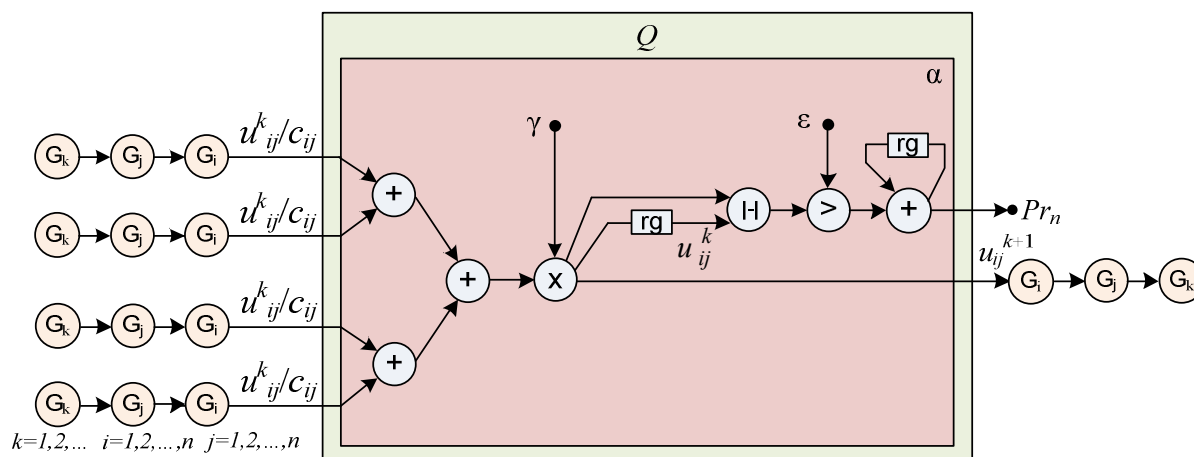


Рис. 3. Кадровая форма метода Якоби решения уравнения Лапласа, состоящая из одного базового подграфа

Особенностью данного кадра является то, что выходом адресных вершин-массивов G являются как вычисленные на предыдущей итерации узловые значения функции u_{ij}^k , так и граничные значения c_{ij} , представляющие собой константы. И константы, и найденные приближения сеточной функции хранятся в одном массиве внешней оперативной памяти, в котором обновляются только значения u_{ij}^{k+1} , вычисленные на текущей итерации.

Дальнейшая редукция производительности путем сокращения функциональных устройств базового подграфа α_i может быть осуществлена путем замены трех вершин суммирования значений функции на одну, где сложения будут производиться последовательно. Это позволит не только сократить аппаратные затраты на функциональные устройства, но и уменьшить количество внешних коммуникационных каналов. В данном случае число операционных вершин базового подграфа было сокращено с семи до пяти, а количество внешних каналов уменьшилось вдвое – с шести до трех (рис. 4).

Особенностью разностной аппроксимации оператора Лапласа на пятиточечном шаблоне является то, что каждое уравнение системы (кроме точек, примыкающих к граничным областям) содержит пять неизвестных, используемых для нахождения приближенных значений u_{ij} функции $u(x, y)$ в узлах двумерной сетки (x_i, y_j) . Поэтому обратная связь накапливающего сумматора в базовом подграфе содержит четыре регистра rg для хранения промежуточных результатов суммирования. При аппроксимации на шаблонах, содержащих большее число точек, количество регистров в обратной связи накапливающего сумматора должно быть соответственно увеличено.

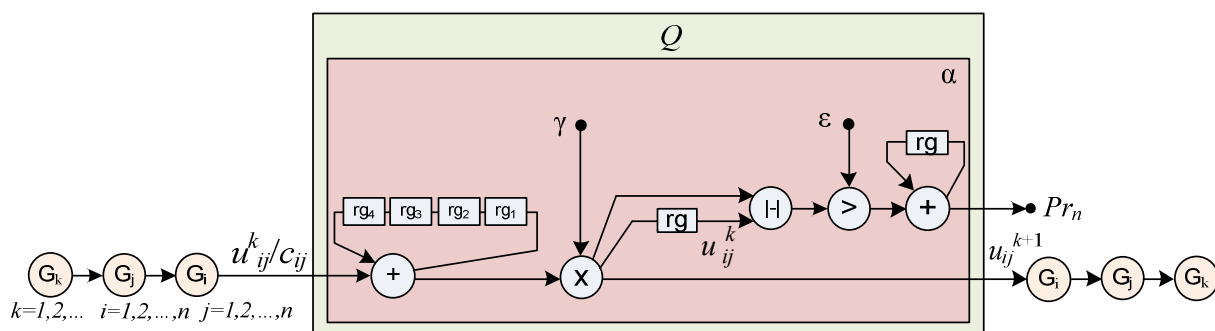


Рис. 4. Кадровая форма метода Якоби решения уравнения Лапласа с предельно сокращенным числом базовых подграфов и операционных вершин

Кадровая форма задачи может состоять как из нескольких, так и из одного предельно редуцированного базового подграфа. Выбор реализации, состоящей из одного базового подграфа, главным образом обусловлен коммуникационными ресурсами вычислительной системы и возможностью построения конвейерной структуры с минимально возможным числом внешних и внутренних информационных каналов.

При параллельно-конвейерной организации вычислительного процесса базовый подграф, соответствующий одной ступени конвейера, будет иметь вид, представленный на рис. 5.

Каждая ступень конвейера реализует функционально законченный фрагмент P , общее количество которых упорядочено в кортеж $\langle P^1, P^2, \dots, P^k \rangle$ по числу итераций k .

На каждой итерации $(k+1)$ на вход ступени поступает поток операндов, представляющих собой константные значения функции c_{ij} на границе области и ее приближения u_{ij}^k , полученные на предыдущей итерации. Выходом ступени является поток транзитных данных c_{ij} и рассчитанных значений u_{ij}^{k+1} , которые поступают на вход следующей ступени для вычисления узловых значений функции на следующей итерации.

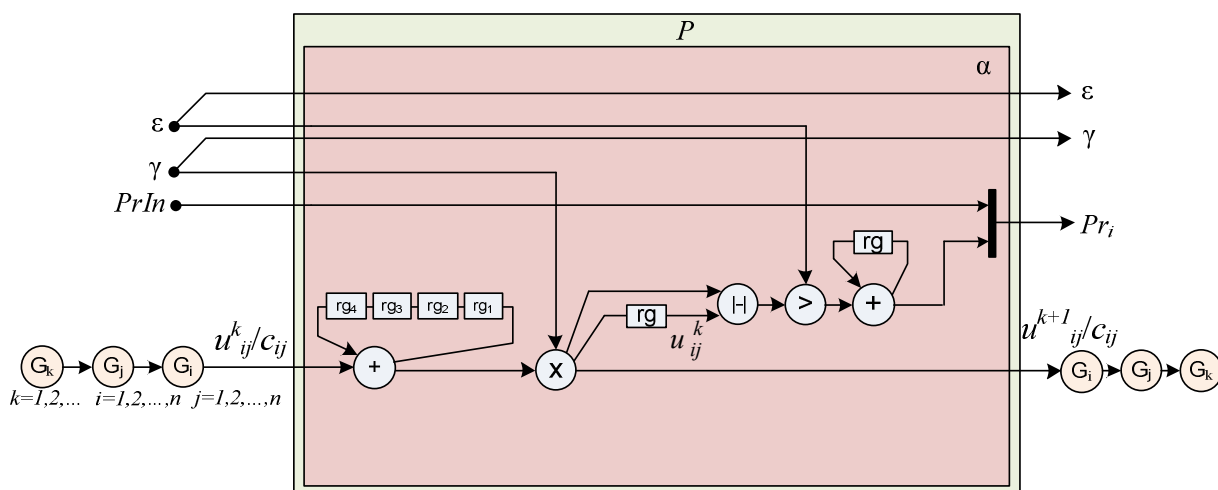


Рис. 5. Базовый подграф метода Якоби решения уравнения Лапласа, соответствующий одной ступени вычислительного конвейера

Помимо информационных дуг, связанных с поступлением и выдачей вычисляемых приближений функции, структура базового подграфа должна быть дополнена связями, осуществляющими транзит констант ϵ, γ в следующую ступень конвейера. Также в структуру подграфа вводится дополнительный вход $PrIn_i$, на который подается значе-

ние предиката предыдущей ступени. Выходной предикат Pr_i , формируемый внутри подграфа, получается в результате мультиплексирования потока предикатов предыдущих ступеней и собственного вычисленного предиката.

Таким образом, в вычислительном конвейере, представленном на рис. 6, достигается параллелизм решения задачи на уровне итераций.

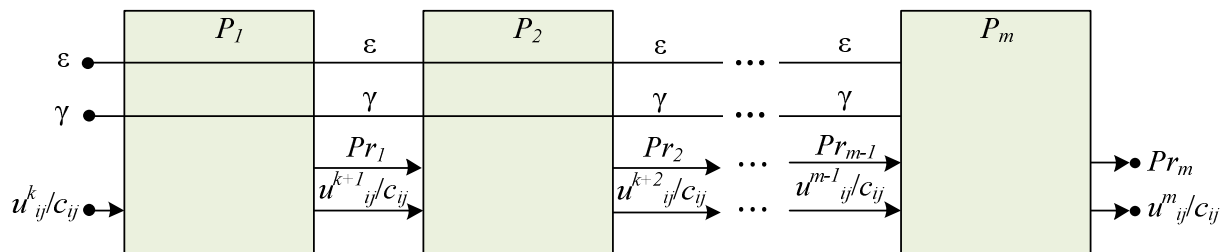


Рис. 6. Параллельно–конвейерная форма метода Якоби решения уравнения Лапласа с применением распараллеливания по итерациям

2. Практические результаты решения на РВС краевой задачи Дирихле для уравнения Лапласа

Численное решение задачи Дирихле для уравнения Лапласа методом Якоби было реализовано авторами на базовом вычислительном модуле «Тайгета» [16]. В результате аппроксимации 5–точечной разностной схемы на квадратной сетке с шагом $h = 0,01$ была получена СЛАУ размерностью $n = 10^4$. При размещении в одной ПЛИС XC7VX485T–1FFG1761 ($48 \cdot 10^6$ эквивалентных вентилях, 400 МГц, 192 элементарных процессора стандарта IEEE-754) базового модуля 59 ступеней конвейера время решения СЛАУ составило 0,22 с. Для решения этой задачи на персональном компьютере (Intel Core i5–3570K, 3.4 ГГц, 8 Гб ОЗУ) при использовании одного ядра процессора потребовалось 4,61 с. Таким образом, на одной ПЛИС по сравнению с ПК было получено ускорение вычислений примерно в 21 раз.

Всего базовый модуль «Тайгета» содержит 32 ПЛИС, что позволяет разместить в едином решающем поле вычислительный конвейер, состоящий из 1888 ступеней.

Масштабирование алгоритма продемонстрировало увеличение скорости решения задачи при увеличении числа используемых кристаллов ПЛИС. На графиках на рис. 7 и 8 представлены полученные экспериментальные результаты для времени решения задачи и ускорения по сравнению с одним ядром процессора Intel Core i5–3570K, 3.4 ГГц для количества ПЛИС от 1 до 32.

Результаты вычислительных экспериментов для параллельной реализации на кластере МВС–1000/ИВМ алгоритма численного решения методом Якоби краевой задачи на кластерной многопроцессорной вычислительной системе для числа процессов от 1 до 22 приведены в работе [17]. Согласно полученным в [17] экспериментальным данным, стабильное увеличение ускорения при решении задачи для всех рассматриваемых в [17] вариантов реализации коммуникаций и способа декомпозиции вычислительной области наблюдалось вплоть до использования 22–х процессоров, после чего ускорение решения задачи приобрело неустойчивый характер по причине преобладания времени межпроцессорных обменов над вычислениями.

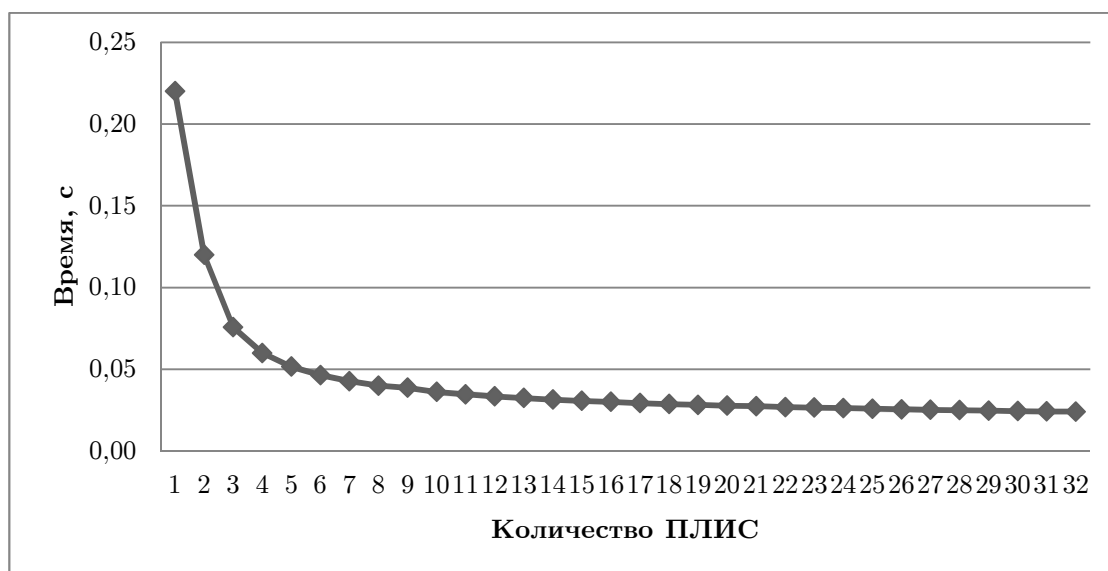


Рис. 7. Метод Якоби решения уравнения Лапласа. Зависимость времени вычислений от числа используемых для решения задачи ПЛИС

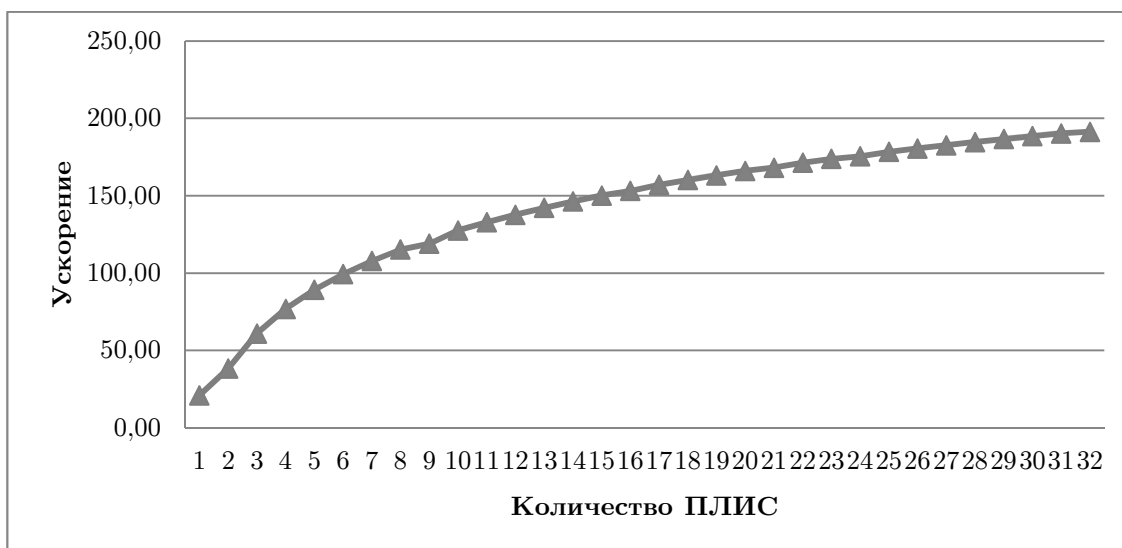


Рис. 8. Метод Якоби решения уравнения Лапласа. Изменение ускорения в зависимости от числа используемых для решения задачи ПЛИС

Согласно данным, приведенным на рис.7 и 8 при использовании ПЛИС для численного решения задачи Дирихле для уравнения Лапласа методом Якоби наблюдается устойчивый рост ускорения решения задачи при увеличении используемого вычислительного ресурса до 32 кристаллов ПЛИС включительно. Параллельно-конвейерная форма реализации матричных задач минимизирует использование внешних выводов, но остается требовательной к остальным вычислительным ресурсам системы – тактовой частоте, объему логики и количеству внутрикристальной памяти ПЛИС. Использование ПЛИС для решения матричных задач позволяет существенно увеличить производительность, в том числе за счет использования кристаллов новых семейств. Так, для каждого следующего семейства ПЛИС Xilinx, начиная с семейства Virtex-4, наблюдается не менее чем 25% рост тактовых частот и, как правило, двукратное увеличение логической

емкости ПЛИС. Ресурс встроенной блочной памяти в каждой серии растет еще большими темпами: реализованное на 16 нм техпроцессе семейство Virtex UltraScale+ имеет в четыре раза больше памяти, чем в предыдущей серии Virtex-7 с нормами 28 нм. Поскольку объем внутрикристалльной памяти имеет критическое значение для успешной реализации задач решения больших СЛАУ, прогресс этого показателя делает использование ПЛИС весьма перспективным.

Заключение

Предложенный конвейерно-параллельный способ решения систем линейных уравнений в задачах математической физики может быть применен и к другим итерационным методам, представимым в потоковом виде, среди них: метод Гаусса—Зейделя, попеременно-треугольный метод и так далее. Реализация этого способа сегодня осуществима на ПЛИС и построенных на их основе реконфигурируемых системах, что говорит в пользу широкого использования программируемых архитектур в высокопроизводительных вычислениях.

Применение метода распараллеливания по итерациям позволяет существенно сократить количество внешних каналов обмена по сравнению с методами распараллеливания по данным. Поскольку стандартные методы и средства в настоящее время не могут в полной мере решить проблему падения реальной производительности в ряде задач с интенсивным доступом к данным, имеет смысл использовать альтернативные многопроцессорные архитектуры с «гибкой» реализацией различных форм распараллеливания.

Среди направлений дальнейших исследований можно выделить поиск эффективных алгоритмов решения на РВС разреженных СЛАУ итерационными методами с различного вида предобуславливанием (на основе неполного LU-разложения, на основе аппроксимации обратной матрицы, на основе алгебраического многосеточного метода и так далее). В настоящее время остается недостаточно изученным вопрос о соотношении времени, затрачиваемого на построение матрицы предобуславливателя, и последующего процесса нахождения решения с сокращенным количеством итераций.

Работа выполнена при частичной финансовой поддержке Южного научного центра Российской академии наук по теме №0256-2015-0080 (00-16-15) государственного задания в рамках программы 1.5П «Проблемы создания высокопроизводительных, распределенных и облачных систем и технологий. Интеллектуальные информационные технологии и системы».

Литература

1. Самарский А.А., Гулин А.В. Численные методы математической физики: учебное пособие. 2-е изд. М.: Научный мир, 2003. 316 с.
2. Гергель В.П. Теория и практика параллельных вычислений. М.: БИНОМ, 2007. 424 с.
3. Asanovic K. et al. A view of the parallel computing landscape // Communications of the ACM. 2009. Vol. 52, No. 10. P. 56–67.
4. Guo X. et al. Developing a scalable hybrid MPI/OpenMP unstructured finite element model // Computers & Fluids. 2015. Vol. 110. P. 227–234.

5. Li R., Saad Y. GPU-accelerated preconditioned iterative linear solvers // The Journal of Supercomputing. 2013. Vol. 63, No. 2. P. 443–466.
6. Wolfson-Pou J., Chow E. Reducing Communication in Distributed Asynchronous Iterative Methods // Procedia Computer Science. 2016. Vol. 80. P. 1906–1916.
7. Liu X., Zhong Z., Xu K. A hybrid solution method for CFD applications on GPU-accelerated hybrid HPC platforms // Future Generation Computer Systems. 2016. Vol. 56. P. 759–765.
8. Deng L. et al. CPU/GPU Computing for AN Implicit Multi-Block Compressible Navier-Stokes Solver on Heterogeneous Platform // International Journal of Modern Physics: Conference Series. World Scientific Publishing Company. 2016. Vol. 42. P. 163–166.
9. Stroia I. et al. GPU accelerated geometric multigrid method: Comparison with preconditioned conjugate gradient // High Performance Extreme Computing Conference (HPEC) IEEE. 2015. P. 1–6.
10. Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И. Реконфигурируемые мультиконвейерные вычислительные структуры. Ростов-на-Дону: Изд-во ЮНЦ РАН, 2008. 320 с.
11. Virtex-6 Family Overview. Xilinx, 2015. URL: http://www.xilinx.com/support/documentation/data_sheets/ds150.pdf (дата обращения: 16.06.2016).
12. Series FPGAs Overview. Xilinx, 2015. URL: http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf (дата обращения: 16.06.2016).
13. UltraScale Architecture and Product Overview. Xilinx, 2016. URL: http://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf (дата обращения: 16.06.2016).
14. Тихонов А.Н. Самарский А.А. Уравнения математической физики. М.: Наука, 1972. 736 с.
15. Левин И.И., Сорокин Д.А., Мельников А.К., Дордопуло А.И. Решение задач с существенно-переменной интенсивностью потоков данных на реконфигурируемых вычислительных системах // Вестник компьютерных и информационных технологий. М.: Машиностроение, 2012. № 2. С. 49–56.
16. Левин И.И., Дордопуло А.И., Каляев И.А., Доронченко Ю.И., Раскладкин М.К.. Современные и перспективные высокопроизводительные вычислительные системы с реконфигурируемой архитектурой // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». 2015. Т. 4, № 3. С. 24–39.
17. Каропова Е. Д., Шайдуров В. В., Вдовенко М. С. Параллельные реализации метода конечных элементов для краевой задачи для уравнений мелкой воды // Вестник ЮУрГУ. Серия «Математическое моделирование и программирование». 2009. Вып. 3, № 17(150). С. 73–85.

IMPLEMENTATION OF ITERATION METHODS FOR SOLUTION OF LINEAR EQUATION SYSTEMS IN PROBLEMS OF MATHEMATICAL PHYSICS ON RECONFIGURABLE COMPUTER SYSTEMS

© 2016 I.I. Levin, A.I. Dordopulo, A.V. Pelipets

*Academician A.V. Kalyaev SRI Multiprocessor Computer System at Southern Federal
University (2 Chekhov st., GSP-284, Taganrog, 347928, Russia)*

E-mail: levin@mvs.tsure.ru, scorpio@mvs.tsure.ru, pelipets@mail.ru

Received: 12.07.2016

In the paper we consider unique features of implementation of iteration methods for solution of linear equation systems in problems of mathematical physics on parallel computer systems, such as geometric decomposition of the computational domain and data parallelization in sequentially performed iterations with intensive data exchange between processors and memory. Standard methods of implementation of iteration methods of solution of linear equation systems with multiple exchanges with memory and between processors, which considerably reduce the performance, require a big number of communication channels in the computer system for implementation of complex topologies and hierarchic schemes of data storage. The solution of this problem is use of multiprocessor systems with reconfigurable architecture which allow adaptation of their architecture to the structure of iteration algorithms of mathematical physics owing to iteration parallelization. In this paper we analyze implementation of the Jacobi method for the Dirichlet problem for the Laplace equation on a reconfigurable computer system. This implementation is an example which illustrates reduction of the number of external data exchange channels, which are the most critical resource of the reconfigurable computer system.

Keywords: reconfigurable computer systems, FPGAs, numerical methods of mathematical physics, iteration parallelization, computational pipeline.

FOR CITATION

Levin I.I., Dordopulo A.I., Pelipets A.V. Implementation of Iteration Methods for Solution of Linear Equation Systems in Problems of Mathematical Physics on Reconfigurable Computer Systems. Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering. 2016. vol. 5, no. 4. pp. 5–18. (in Russian) DOI: 10.14529/cmse160401.

References

1. Samarskij A.A., Gulin A.V. *Chislennye metody matematicheskoy fiziki* [Numerical Methods of Mathematical Physics]. 2 ed. M: Nauchnyj mir, 2003. 316 p.
2. Gergel' V.P. *Teorija i praktika parallel'nyh vychislenij* [Theory and Practice of Parallel Calculations]. M.: BINOM, 2007. 424 p.
3. Asanovic K. et al. A View of the Parallel Computing Landscape. *Communications of the ACM*, 2009. vol. 52, no. 10. pp. 56–67. DOI: 10.1145/1562764.1562783.
4. Guo X. et al. Developing a Scalable Hybrid MPI/OpenMP Unstructured Finite Element Model. *Computers & Fluids*. 2015. vol. 110. pp. 227–234. DOI: 10.1016/j.compfluid.2014.09.007.
5. Li R., Saad Y. GPU-accelerated Preconditioned Iterative Linear Solvers. *The Journal of Supercomputing*. 2013. vol. 63, no. 2. pp. 443–466. DOI: 10.1007/s11227-012-0825-3.

6. Wolfson-Pou J., Chow E. Reducing Communication in Distributed Asynchronous Iterative Methods. *Procedia Computer Science*. 2016. vol. 80. pp. 1906–1916. DOI: 10.1016/j.procs.2016.05.501.
7. Liu X., Zhong Z., Xu K. A Hybrid Solution Method for CFD Applications on GPU-accelerated Hybrid HPC Platforms. *Future Generation Computer Systems*. 2016. vol. 56. pp. 759–765. DOI: 10.1016/j.future.2015.08.002.
8. Deng L. et al. CPU/GPU Computing for AN Implicit Multi-Block Compressible Navier-Stokes Solver on Heterogeneous Platform. *International Journal of Modern Physics: Conference Series*. World Scientific Publishing Company. 2016. vol. 42. pp. 163–166.
9. Stroia I. et al. GPU Accelerated Geometric Multigrid Method: Comparison with Preconditioned Conjugate Gradient. *High Performance Extreme Computing Conference (HPEC)*. IEEE, 2015. pp. 1–6. DOI: 10.1109/HPEC.2015.7322480.
10. Kaljaev I.A., Levin I.I., Semernikov E.A., Shmojlov V.I. *Rekonfiguriruemye mul'tikonvejernye vychislitel'nye struktury* [Reconfigurable Multipipeline Computing Structures]. Rostov-on-Don: ed. SSC RAS, 2008. 320 p.
11. *Virtex-6 Family Overview*. Xilinx, 2015. Available at: http://www.xilinx.com/support/documentation/data_sheets/ds150.pdf (accessed: 16.06.2016).
12. *7 Series FPGAs Overview*. Xilinx, 2015. Available at: http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf (accessed: 16.06.2016).
13. *UltraScale Architecture and Product Overview*. Xilinx, 2016. Available at: http://www.xilinx.com/support/documentation/data_sheets/ds890-ultrascale-overview.pdf (accessed: 16.06.2016).
14. Tihonov A.N. Samarskij A.A. *Uravenenija matematicheskoy fiziki* [Equations of Mathematical Physics]. Moscow: Nauka, 1972. 736 p.
15. Levin I.I., Sorokin D.A., Mel'nikov A.K., Dordopulo A.I. The Decision of Tasks with Essential and Variable Intensity of Data Streams on Reconfigurable Computing Systems. *Vestnik komp'juternyh i informacionnyh tehnologij* [Bulletin of Computer and Information Technologies]. Moscow: Mashinostroenie, 2012. no. 2. pp. 49–56. (in Russian)
16. Levin I.I., Dordopulo A.I., Kaljaev I.A., Doronchenko Y.I., Rasklakin M.K. Modern and Next-generation High-performance Computer Systems with Reconfigurable Architecture. *Vestnik YuUrGU. Seriya «Vychislitel'naja matematika i informatika»* [Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering]. 2015. vol. 4, no. 3. pp. 24–39. DOI: 10.14529/cmse150303. (in Russian)
17. Karepova E.D., SHajdurov V.V., Vdovenko M.S. Parallel Realization of a Method of Final Elements for a Regional Task for the Equations of Small Water. *Vestnik YuUrGU. Seriya «Matematicheskoe modelirovanie i programmirovanie»* [Bulletin of the South Ural State University. Series: Mathematical Modeling, Programming & Computer Software]. 2009. vol. 3, no. 17(150). pp. 73–85. (in Russian)