Dordopulo Alexey Igorevich – Southern Scientific Centre of the Russian Academy of Sciences; e-mail: scorpio@mvs.tsure.ru; 41, Chekhov aven.; Rostov-on-Don, 344006, Russia; phone: +78634315491; cand. of eng. sc.; senior scientist.

Sorokin Dmitry Anatolievich – Scientific-Research Institute Multiprocessing Computing Systems after Kalyaev of South Federal University; e-mail: jotun@inbox.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634315491; scientific associate.

УДК 004.272.22

Ю.И. Доронченко

ПРОЦЕДУРА МЕЖИТЕРАЦИОННОЙ РЕДУКЦИИ ЗАДЕРЖЕК ДЛЯ КОНВЕЙЕРНОЙ РЕАЛИЗАЦИИ РЕКУРСИВНЫХ ВЫЧИСЛЕНИЙ*

Предложен способ сокращения аппаратных затрат при конвейерной реализации рекурсивных вычислений в вычислительно трудоемких алгоритмах защиты информации. Впервые показана эффективность применения предложенного метода для реконфигурируемых вычислительных систем. Применение предложенной процедуры межитерационной редукции обеспечивает минимальные рекурсивные связи при распределении вычислений во времени, что позволяет сократить затраты на синхронизацию от 1,3 до 6 раз и повысить удельную производительность конвейерной реализации рекурсивных вычислений на реконфигурируемой вычислительной системе.

Конвейерные вычисления; защита информации; рекурсивные вычисления; редукция, синхронизация.

Y.I. Doronchenko

PROCEDURE OF INTER-ITERATION REDUCTION OF DELAYS FOR PIPELINE RECURSIVE CALCULATIONS

The paper covers the method of reduction of hardware resource needed for pipeline recursive calculations in computationally laborious algorithms of information security. For the first time the author has shown effectiveness of application of the suggested method in reconfigurable computer systems. The developed procedure of inter-iteration reduction provides minimal recursive constraints for time distributed calculations and reduction of synchronization costs from 1.3 up to 6 times. Besides, it helps to increase specific performance of pipeline recursive calculations for reconfigurable computer system.

Pipeline calculations; information security; recursive calculations; reduction; synchronization.

Вопросы безопасности больших вычислительных и управляющих систем требуют оперативной обработки больших массивов данных с помощью вычислительно-трудоемких алгоритмов, что обусловливает применение высокопроизводительных вычислительных комплексов, реализующих методы блокирования несанкционированного доступа к информации. Реализация подобных методов защиты зачастую требует существенного аппаратного ресурса, поэтому актуальны методы, позволяющие снизить требования к ресурсу а, следовательно, и к массогабаритным характеристикам и стоимости.

При построении конвейерных вычислительных структур [1] существенные затраты аппаратного ресурса требуют вопросы синхронизации информационных потоков. В этой связи предлагается метод, позволяющий для рекурсивных вычис-

* Исследования выполнены при финансовой поддержке Министерства образования и науки РФ, госконтракт № 14.527.12.0004 от 03 октября 2011 г.

лений выполнить редукцию задержек в конвейере. Данный подход применим для различных классов научно-технических задач, в том числе криптографии, что обусловливает его применение при обеспечении безопасности различных систем.

Одним из методов эквивалентного преобразования цифровых схем является известный метод редукции регистров. Этот метод основан на замещении задержек входных параметров некоторой функции задержкой результата этой функции и наоборот. Пример простейших редукций представлен на рис. 1.

В задачах, широко использующих рекурсивные вычисления, глубокие задержки, как правило, неизбежны, поэтому весьма затруднительно при построении конвейера оптимально выполнить синхронизацию потоков данных.

Обычно на практике метод редукции регистров применяют в простейших случаях в пределах одной итерации алгоритма.

Однако, рассматривая задачу в целом, можно выполнить редукцию и в более широких пределах, охватив несколько итераций вычислений.

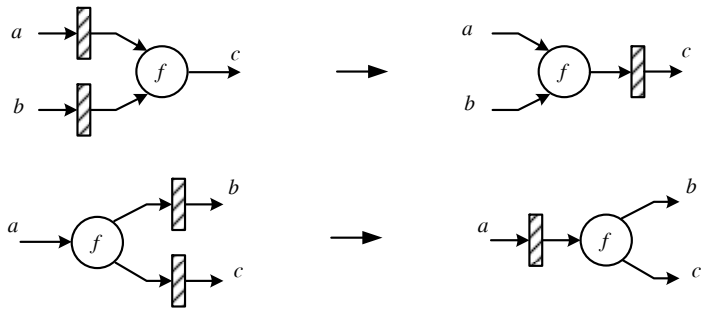

Рис. 1. Пример использования метода редукции регистров

В общем случае предлагаемый метод редукции соответствует двукратному (по времени и пространству) вложенному циклу и основан на том, что вычисления на каждом шаге алгоритма распределяются во времени и в пространстве таким образом, чтобы рекурсивные связи были минимальны.

Будем рассматривать задачи, информационный граф которых описывается следующим выражением:

$$G = \bigcup_{i=1}^{n} V_i \bigcup_{j=1}^{m} \left[\bigcup_{i=1}^{n} D\left(z_{i,j}, w_{i-j}\right) \right] \bigcup D\left(Y, w_n\right), \qquad w_{1-j} = X_j, \tag{1}$$

где V_i – множество изоморфных информационных подграфов, $i = 1..n$;

$z_{i,j}$ – j-я входная информационная вершина подграфа V_i, $j = 1..m$;

w_{i-j} – выходная информационная вершина подграфа V_{i-j};

X_j – входные информационные вершины графа задачи G;

Y – выходная информационная вершина графа G.

Приведенное описание характерно для однонаправленных хэш-функций, для которых подграфы V_i являются итерациями алгоритма. Будем полагать, что данный граф и все его вершины являются конвейеризованными и обрабатывают большое количество данных. Так как для дальнейшего рассуждения количество данных в конвейере не является принципиальным, для простоты изложения опустим индекс, соответствующий номеру обрабатываемого данного. Фрагмент графа задачи представлен на рис.2. Чтобы не загромождать рисунок, указаны не все связи между подграфами. Вообще каждый подграф V_i имеет $2m$ связей.

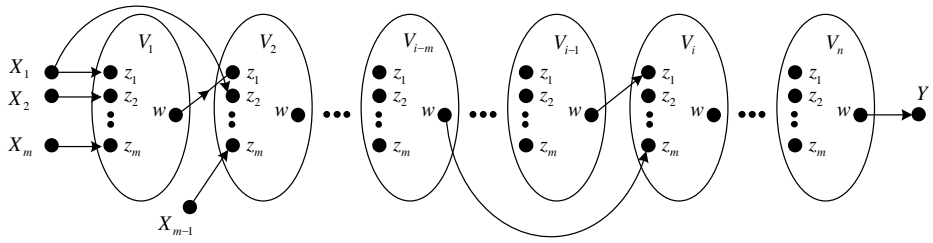

Рис. 2. Фрагмент графа задачи

Рассмотрим подсистему синхронизации, необходимую для реализации данной задачи. Пусть величина задержки, вносимой в результате вычислений подграфа V_i, равна l тактов. Необходимо для каждого подграфа V_i синхронизировать потоки данных, следующие по дугам $\bigcup_{j=1}^{m} D\left(z_{i+j,j}, w_i\right)$. В каждой из этих дуг должны быть реализованы линии задержки глубиной $(j-1)l$ тактов. Так как по всем дугам следуют одинаковые потоки данных, то очевидно, что построение m линий задержек нерационально. Необходимо создание для каждого подграфа V_i одной линии задержки глубиной $(m-1)l$ тактов. Синхронизация данных в конвейере для двух линий задержки показана на рис. 3.

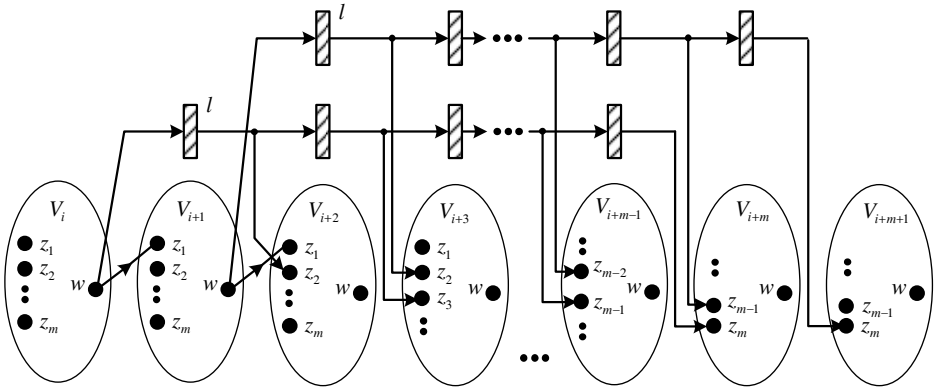

Рис. 3. Синхронизация данных в конвейере

Таким образом, общее количество задержек для всех n подграфов V_i

$$\alpha = nl(m-1). \tag{2}$$

Для простоты рассуждения здесь не показано, что для последних $(n-m)$ подграфов количество элементов задержки l будет, очевидно, меньше, чем $(m-1)$. Этим фактом можно пренебречь в силу того, что $n >> m$.

Из последней формулы видно, что при больших n аппаратные затраты на синхронизацию могут быть очень велики, что отрицательно сказывается на удельной производительности вычислительной системы. Также из формулы следует, что так как параметры n и m являются параметрами задачи и не могут быть изменены, то для снижения α необходимо уменьшать латентность вычислений l в

конвейере V_i. Отчасти это верно и сокращение l в некоторой степени снизит расход аппаратуры, однако, с другой стороны, может привести к уменьшению частоты работы вычислительного устройства. Покажем, как, применяя межитерационную регистровую редукцию, можно кардинально сократить затраты на построение подсистемы синхронизации при решении подобных задач.

Для этого нужно подробнее рассмотреть структуру подграфа V_i. Пусть подграф V_i представляет собой объединение нескольких взаимосвязанных подграфов U_j, которые определенным образом связаны с входными информационными вершинами $z_{i,j}$. Характер связей может быть различным. Для наглядности будем полагать, что каждый подграф U_j связан с соответствующей вершиной $z_{i,j}$ так, как показано на рис.4. Тогда подграф V_i можно описать следующим образом:

$$V_i = \bigcup_{j=1}^{m} U_j \bigcup_{j=1}^{m} D\left(U_j, U_{j-1}\right) \bigcup_{j=1}^{m} D\left(U_j, z_{i,j}\right), \quad U_0 = w_i . \tag{3}$$

Следует отметить, что порядок выполнения подграфов U_j (а, следовательно, и связи между ними $\bigcup_{j=1}^{m} D\left(U_j, U_{j-1}\right)$) достаточно условный, так как если говорить о хэш-функциях, в качестве U_j обычно выступает операция сложения, обладающая свойством ассоциативности.

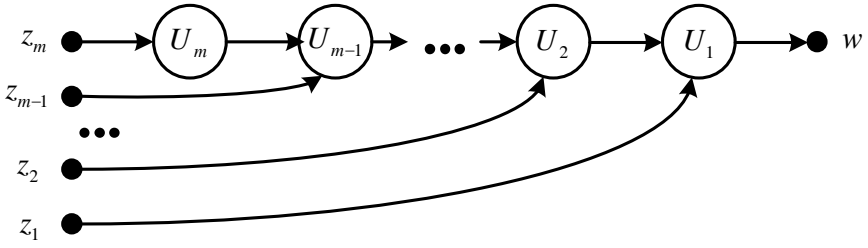

Рис. 4. Подграф V_i

Положим для простоты количество подграфов U_j $m = 4$, пусть каждый из них в процессе вычислений вносит единичную задержку. Заметим, что результат каждой $(i-j)$-й итерации алгоритма (результат подграфов V_{i-j}) является входом операции U_j подграфа V_i. Распределяя во времени вычисления в итерациях так, чтобы рекурсивные связи были минимальны, можно получить пространственно-временную картину вычислений, показанную на рис. 5.

Подобное распределение вычислений будем называть межитерационной регистровой редукцией.

Как видим, синхронизация потоков данных между итерациями алгоритма происходит автоматически и не требует внесения дополнительных задержек. Аппаратные затраты на синхронизацию в данном случае сводятся к минимуму, что приводит к увеличению удельной производительности вычислительной системы.

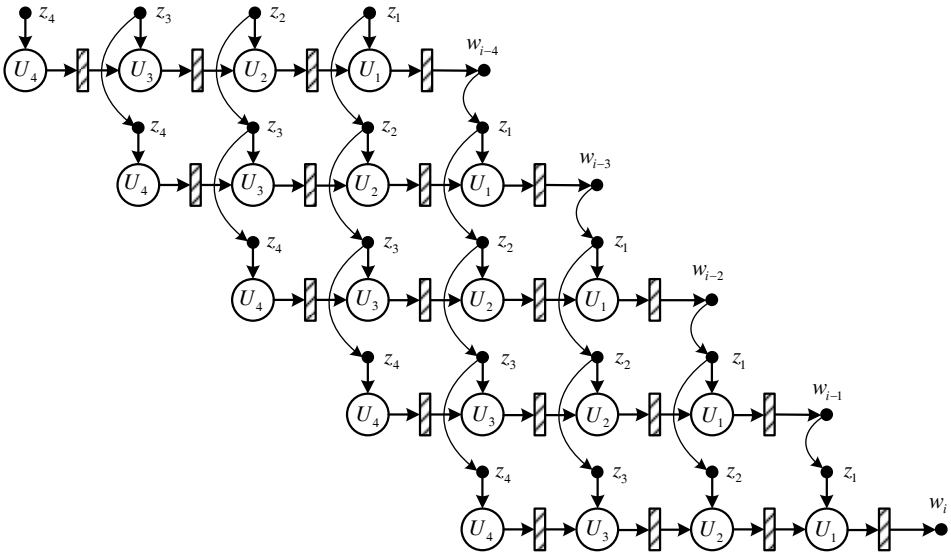

Рис. 5. Межитерационная редукция

В общем случае структура подграфа V_i является более сложной. В зависимости от алгоритма решаемой задачи в подграфе V_i могут меняться количество операционных вершин U_j, количество информационных вершин $z_{i,j}$, число и характер информационной зависимости между подграфами V_i. Реализация различных алгоритмов подобной структуры на реконфигурируемой вычислительной системе показала, что распределение вычислений во времени и пространстве с минимальными рекурсивными связями в соответствии с данным подходом к организации вычислительного процесса позволяет сократить затраты на синхронизацию от 1,3 до 6 раз.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Каляев А.В., Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: ООО «Изд-во Янус-К», 2003. – 380 с.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

Доронченко Юрий Иванович – Научно-исследовательский институт многопроцессорных вычислительных систем им. А.В. Каляева Южного федерального университета; e-mail: doronchenko@mvs.tsure.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 88634315491; к.т.н.; зав. лабораторией.

Doronchenko Yrij Ivanovich – Scientific-Research Institute Multiprocessing Computing Systems after Kalyaev of South Federal University; e-mail: doronchenko@mvs.tsure.ru; 2, Checkhov street, Taganrog, 347928, Russia; phone: +78634315491; cand. of eng. sc.; head of laboratory.