Программирование вычислительных систем гибридного типа на основе метода редукции производительности*

А.И. Дордопуло[1], И.И. Левин[2], И.А. Каляев[1], В.А. Гудков[1], А.А. Гуленок[1]

Научно-исследовательский институт многопроцессорных вычислительных систем Южного федерального университета [1], ООО «Научно-исследовательский центр супер-ЭВМ и нейрокомпьютеров»[2]

В статье предлагаются методы редукции производительности, составляющие основу разрабатываемой технологии ресурсонезависимого программирования вычислительных систем гибридного типа, содержащих реконфигурируемые и микропроцессорные вычислительные узлы. Для описания различных форм организации параллельных вычислений – от структурной формы, характерной для реконфигурируемых вычислительных узлов, через структурно-процедурную и мультипроцедурную к процедурной форме организации вычислений для микропроцессорных узлов - используется язык программирования высокого уровня COLAMO. Преобразования между различными формами организации вычислений для быстрой автоматизированной адаптации прикладной программы под изменившуюся конфигурацию аппаратного ресурса вычислительной системы выполняются с помощью методов редукции производительности.

Ключевые слова: редукция производительности, язык программирования высокого уровня, программирование вычислительных систем гибридного типа, технологии ресурсонезависимого программирования.

1. Введение

Рост сложности решаемых прикладных задач требует постоянного повышения производительности вычислительных систем. Для достижения высокой реальной производительности разработчики ведут постоянный поиск новых архитектурных решений, а также создают новые технологии и средства программирования, повышающие эффективность решения прикладных задач. Одним из перспективных способов достижения высокой реальной производительности вычислительной системы (ВС) при решении задач является адаптация архитектуры ВС под структуру решаемой задачи и создание специализированного вычислительного устройства, эффективно реализующего алгоритм вычислений. Большинство практических задач требует совмещения в едином вычислительном контуре как последовательных, так и параллельных вычислительных фрагментов для эффективной реализации как структурных [1], так и процедурных [1] фрагментов вычислений. Решение этой проблемы многие разработчики видят в создании вычислительных систем с гибридной организацией вычислений, содержащих различные по архитектуре вычислительные узлы, на которых можно реализовать как структурные, так и процедурные вычисления в едином вычислительном контуре. Симбиоз узлов с разной архитектурой в рамках единой вычислительной системы теоретически позволяет повысить реальную производительность вычислительной системы за счет возможности эффективной реализации как структурных, так и процедурных фрагментов вычислений на узлах различной архитектуры вычислительной системы гибридного типа (ВСГТ).

ВСГТ может содержать реконфигурируемые вычислительные узлы и узлы универсальных микропроцессоров, в роли которых могут выступать универсальные процессоры, графические процессоры или ускорители Intel Xeon Phi. Поэтому для программирования таких вычислительных систем используются технологии программирования гетерогенных вычислительных систем: CUDA, OpenACC, OpenCL и т.д., в основе которых лежат расширения языков про-

*Работа выполнена при частичной финансовой поддержке Министерства образования и науки РФ по Соглашению о предоставлении субсидии № 14.578.21.0006 от 05.06.2014, уникальный идентификатор RFMEFI57814X0006

граммирования C, C++, FORTRAN, учитывающие архитектуру специализированного микропроцессорного узла. К существенным недостаткам этих технологий программирования относятся плохая переносимость готовых решений между ВС различной архитектуры и конфигурации и плохая масштабируемость программ. Основной причиной указанных недостатков является подход к программированию ВС, при котором осуществляется разбиение задачи на отдельные фрагменты, каждый из которых реализуется на отдельном узле (на отдельном устройстве) гибридной вычислительной системы. Таким образом, выполняется независимое программирование каждого задействованного узла ВС, в результате чего любое изменение конфигурации ВС или изменение исходного кода прикладной программы приводит к необходимости повторного разбиения задачи на фрагменты и созданию локальных программ для каждого узла ВС. В настоящее время широкое применение вычислительных систем гибридного типа для решения практических прикладных задач сдерживается высокой сложностью программирования, поскольку для эффективного использования архитектурных преимуществ всех вычислительных узлов ВСГТ необходима разработка подпрограмм для каждого узла ВСГТ с учетом различных вариантов организации вычислений.

Поэтому целью описанных в статье научных исследований является сокращение времени разработки и портации параллельных прикладных программ для ВСГТ с возможностью простой и быстрой адаптации прикладной программы под изменившуюся архитектуру или конфигурацию ВСГТ.

2. Технология ресурсонезависимого программирования ВСГТ

Для программирования ВСГТ необходимы как средства описания различных вариантов организации вычислений в едином для различных архитектур языковом пространстве, так и средства трансляции параллельных прикладных программ, объединенные в технологию ресурсонезависимого программирования ВСГТ.

Под технологией программирования понимается совокупность обобщенных и систематизированных научных знаний об оптимальных способах (приемах и процедурах) проведения процесса программирования [2], обеспечивающего в заданных условиях получение программной продукции с заданными свойствами. Под технологией ресурсонезависимого программирования будем понимать совокупность знаний, методов, технологических приемов и средств, которая обеспечивает возможность гибкого изменения и масштабирования программы под новую вычислительную архитектуру или конфигурацию вычислительной системы.

Определим объект исследований – ВСГТ – более формально, поскольку зачастую к вычислительным системам гибридного типа относят гетерогенные вычислительные системы. Это приводит к тому, что для программирования ВСГТ используют технологии программирования, предназначенные для программирования гетерогенных ВС.

Гетерогенная ВС содержит архитектурно-одинаковые вычислительные узлы разных типов с одинаковым типом организации вычислений и способом обработки информации. Типичным примером таких систем являются системы, содержащие одновременно как микропроцессоры, так и графические процессоры.

ВС гибридного типа содержит архитектурно-различные вычислительные узлы с различным типом организации вычислений и одинаковым способом обработки информации. Примером таких систем являются вычислительные устройства с микропроцессорами и кристаллами ПЛИС в едином контуре, которые и являются предметом рассмотрения в рамках настоящей статьи.

Для обеспечения функционирования унифицированных процессорных и реконфигурируемых вычислительных узлов в едином контуре языковые средства разрабатываемой технологии ресурсонезависимого программирования должны обеспечивать возможность описания фрагментов с различными типами организации вычислений, в том числе работающих с различными частотой, скважностью и разрядностью обрабатываемых данных, а также содержать гибкие и мощные средства масштабирования как фрагментов вычислений, так и числа операций в каждом фрагменте. Методы масштабирования должны обеспечивать возможность как увеличения параллелизма задачи (индукцию) при увеличении аппаратного ресурса, так и возможность сокращения (редуцирования) при сокращении вычислительного ресурса. Авторам не удалось

найти в открытых печатных источниках описаний технологий программирования ВСГТ для реконфигурируемых и микропроцессорных вычислительных узлов, обеспечивающих в автоматическом режиме как индукцию, так и редукцию параллельных программ. Поэтому представленная технология программирования базируется на опыте разработки систем программирования для РВС на основе кристаллов ПЛИС, полученном в ходе выполнения ряда научных и опытно-конструкторских работ в НИИ МВС ЮФУ.

Можно сформулировать ряд требований к технологии ресурсонезависимого программирования ВСГТ. Технология ресурсонезависимого программирования ВСГТ должна:

- обеспечивать поддержку различных форм организации вычислений (структурные, структурно-процедурные, мультипроцедурные вычисления и их различные сочетания) в едином вычислительном контуре;
- обеспечивать возможность программирования унифицированных процессорных и реконфигурируемых вычислительных узлов в едином вычислительном контуре;
- обеспечивать программирование в едином языковом пространстве на языке программирования высокого уровня;
- обеспечивать возможность простого масштабирования прикладной программы как для случая увеличения, так и для случая сокращения доступного аппаратного ресурса;
- должна обеспечивать возможность простой и быстрой адаптации прикладной программы, разработанной для одной конфигурации ВСГТ, под другую конфигурацию ВСГТ, в том числе при добавлении узлов новой архитектуры;
- обеспечивать портацию ресурсонезависимой параллельной программы с помощью перетрансляции без существенной корректировки исходного текста.

Для реализации технологии ресурсонезависимого программирования ВСГТ необходимо выбрать язык программирования, который позволит описывать различные формы организации вычислений и программировать унифицированные процессорные и реконфигурируемые вычислительные узлы в едином вычислительном контуре.

Специализированные языки высокого уровня для программирования реконфигурируемых вычислительных систем (РВС) обладают привычным для большинства программистов персональных ЭВМ синтаксисом языка С и отличаются между собой семантическими особенностями вызова и использования операторов. Для описания параллельных процессов в РВС в этих языках используется изначально последовательная парадигма языка С, семантика которого ориентирована на взаимодействие последовательных процессов, что не позволяет в полной мере использовать все возможности РВС при разработке параллельных программ на этих языках. Это приводит к семантическому разрыву между исходным информационным графом задачи, его описанием на языке высокого уровня и созданной транслятором схемотехнической реализацией. Результатом этого разрыва является существенное снижение эффективности параллельной программы - как правило, в 3-5 раз более низкая производительность по сравнению с приложениями, разработанными на языках HDL-группы.

Перспективным направлением в области программирования РВС является язык высокого уровня COLAMO[3-6], разрабатываемый в НИИ МВС ЮФУ. Язык COLAMO предназначен для описания реализации параллельного алгоритма и создания на основе принципов структурно-процедурной организации вычислений специализированной вычислительной структуры в архитектуре РВС, которая выполняет последовательную смену структурно (аппаратно) реализованных фрагментов информационного графа задачи, каждый из которых является вычислительным конвейером потока операндов. Таким образом, приложение (прикладная задача) для РВС состоит из структурной составляющей, представленной в виде аппаратно реализованных фрагментов вычислений, и процедурной составляющей, представляющей собой единую для всех структурных фрагментов управляющую программу последовательной смены вычислительных структур и организации потоков данных.

Для описания на языке COLAMO процедурной организации вычислений на универсальных процессорах и возможности быстрого перехода от процедурной реализации вычислений на универсальных процессорах к структурной организации вычислений на реконфигурируемых вычислительных узлах в языке COLAMO существует конструкция Implicit. Конструкция Implicit является конструкцией с неявным указанием реализуемой в фрагменте формы организации вычислений (принадлежности к структуре или процедуре). Такой фрагмент может быть

реализован либо процедурно, либо структурно. Переопределение способа реализации конструкции Implicit позволяет прикладному программисту без существенного изменения текста параллельной программы перейти от структурной организации вычислений к процедурной и обратно. Наличие конструкций описания как структурных, так и процедурных фрагментов вычислений прикладной задачи дает возможность программисту создавать единую прикладную программу для всех узлов ВСГТ, что позволяет рассматривать язык программирования высокого уровня COLAMO [3-6] как основу для реализации технологии ресурсонезависимого программирования как реконфигурируемых вычислительных узлов, так и универсальных узлов ВСГТ.

Однако для эффективного программирования ВСГТ языковые средства должны иметь возможность описания фрагментов вычислений, работающих с различной частотой, скважностью и разрядностью обрабатываемых данных для обеспечения масштабирования как фрагментов, так и отдельных устройств не только при увеличении аппаратного ресурса, но и при его сокращении, а также возможность работы с данными переменной разрядности для эффективного использования аппаратного ресурса ВСГТ.

3. Редукция производительности как способ масштабирования вычислений ВСГТ

Основой для простого масштабирования и адаптации прикладной программы как для случая увеличения, так и для случая сокращения доступного аппаратного ресурса, является редукция производительности прикладной программы, под которой понимается пропорциональное сокращение производительности во всех без исключения фрагментах информационного графа задачи с возможным сокращением аппаратных затрат на реализацию вычислительной структуры. Редуцирование производительности параллельных программ позволяет изменять ключевые параметры параллельных программ, поскольку структурная реализация задачи может привести к нехватке доступного аппаратного ресурса, что особенно актуально при переносе задачи на гибридные вычислительные системы различных архитектур и конфигураций.

Возможны следующие виды редукции:

- редукция производительности по вычислительным устройствам;
- редукция производительности по каналам памяти;
- редукция производительности по разрядности;
- редукция производительности по частоте.

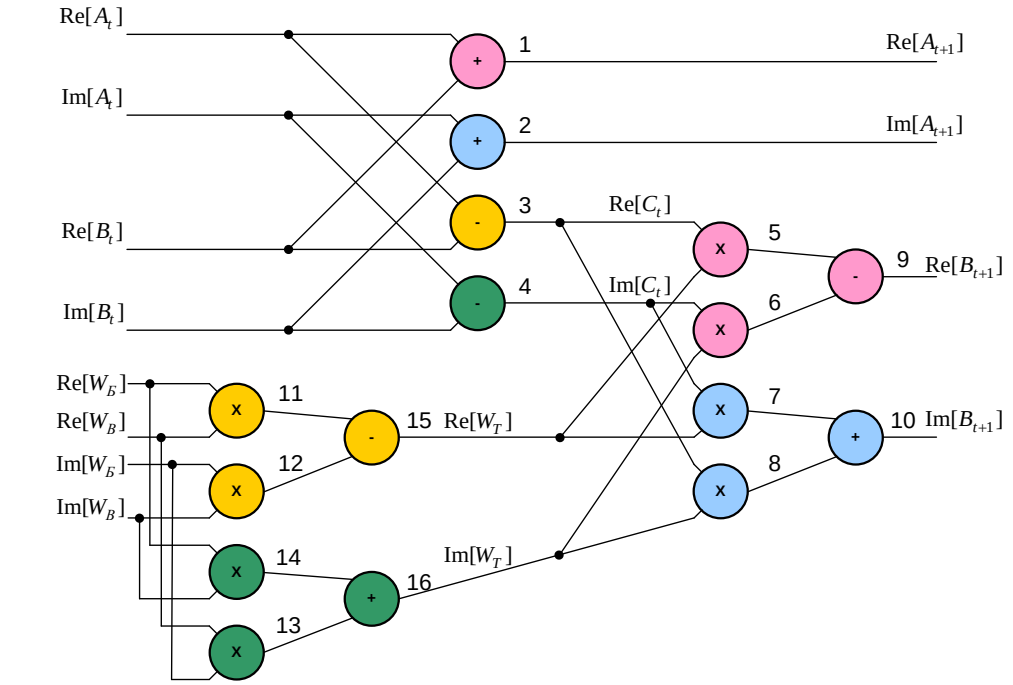
Редукция производительности по вычислительным устройствам основана на сокращении одновременно выполняемых устройств, реализующих вычислительные операции. Иллюстрация принципов функционирования редукции производительности на примере операции быстрого преобразования Фурье представлена на рис. 1 и 2. На рис. 1-а представлен исходный информационный граф операции быстрого преобразования Фурье, содержащий 16 вычислительных устройств, а на рис. 1-б представлена структурная реализация этой операции на РВС с условным заполнением ПЛИС (в допущении, что два вычислительных устройства занимают ресурс одного кристалла ПЛИС). При выполнении редукции производительности по функциональным устройствам осуществляется пропорциональное сокращение числа одновременно работающих устройств, что приводит как к сокращению производительности, так и к сокращению занимаемого аппаратного ресурса. Степень редукции производительности задает коэффициент сокращения параметра редукции. Результат редукции производительности по функциональным устройствам со степенью 2 для базовой операции быстрого преобразования Фурье представлен на рис. 2.

В представленном на рис. 2 примере видно, что число функциональных устройств сократилось вдвое (до восьми8) и результат операции получается не за один такт работы (для структурной реализации, см. рис. 1), а не менее, чем за два такта работы.

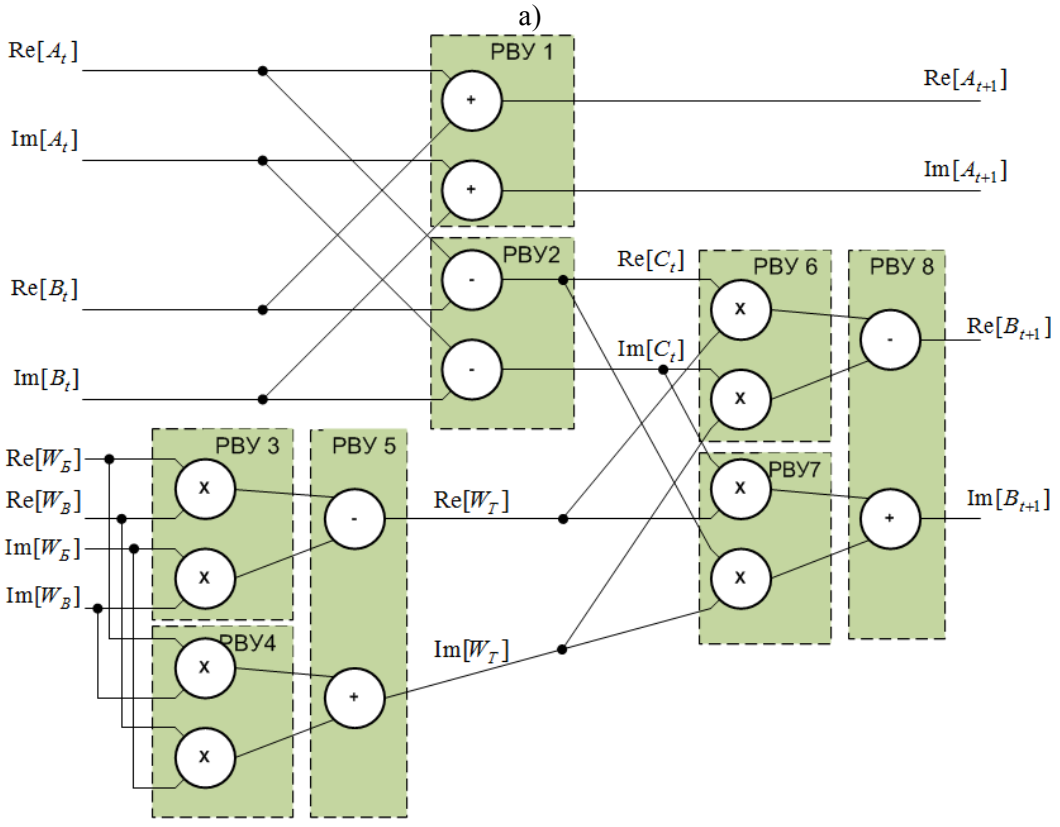
Редукция производительности по вычислительным устройствам может быть реализована в виде:

1) однокадровой параллельной программы с использованием мультиплексоров и скважности, равной степени выполняемой редукции;

2) многокадровой программы с использованием конструкции Let;
3) мультикадровой программы с использованием конструкции MultiCadr.



а)



б)

Рис. 1. Принципы функционирования редукции производительности на примере операции быстрого преобразования Фурье (а - исходный информационный граф операции быстрого преобразования Фурье, б - структурная реализация операции быстрого преобразования Фурье на РВС с условным заполнением ПЛИС)

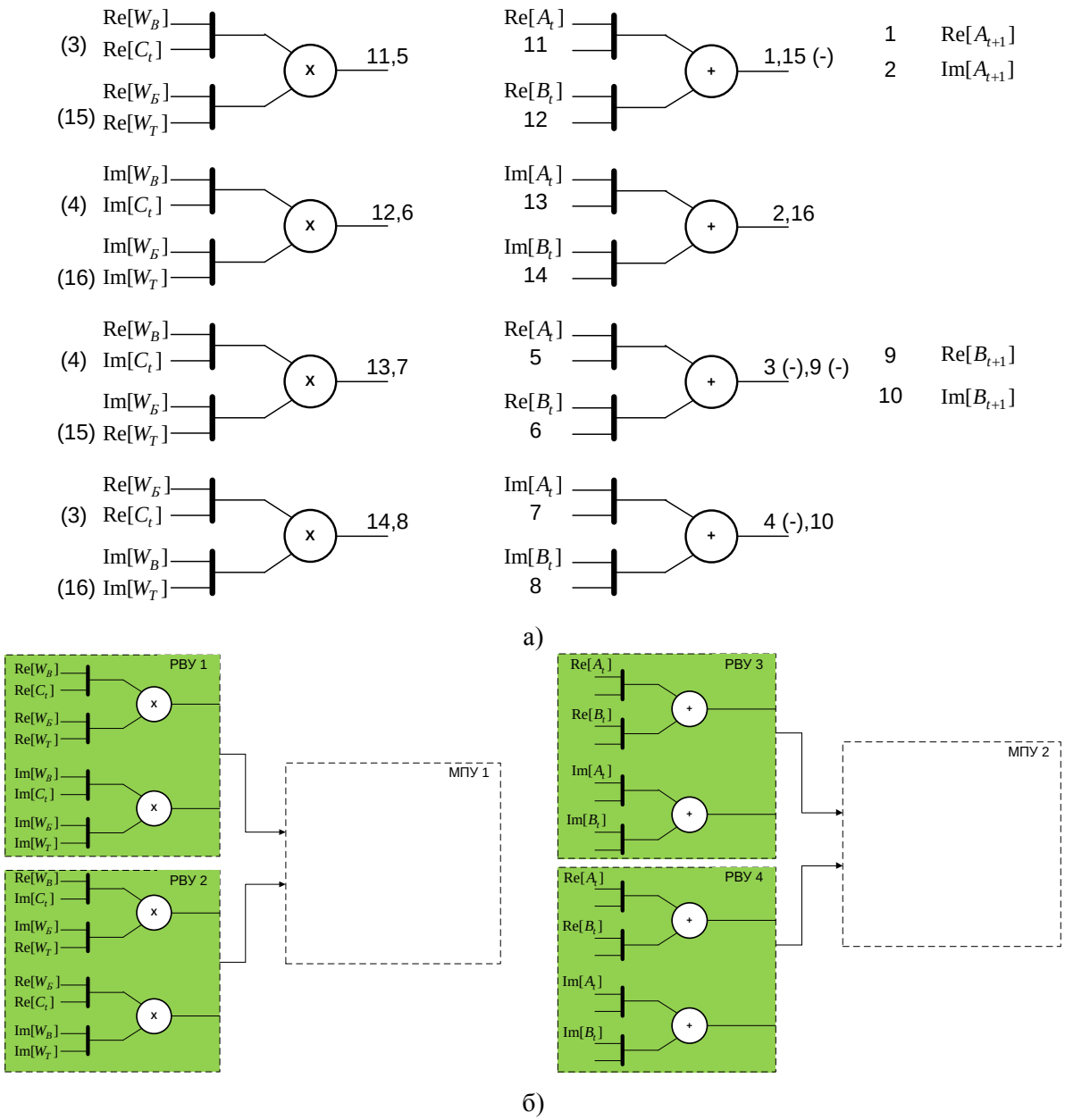


Рис. 2. Результат редукции производительности по функциональным устройствам со степенью 2 для базовой операции быстрого преобразования Фурье

Редукция производительности по функциональным устройствам в виде однокадровой программы заключается в реализации вычислений с помощью одного кадра и набора мультиплексоров, обеспечивающих управление информационными потоками данных, которые подаются в вычислительную структуру со скважностью, равной степени выполняемой редукции. Для описания редукции по вычислительным устройствам в виде однокадровой параллельной программы в язык высокого уровня была введена директива Reduction of Device следующего вида:

#Reduction of Device X;

Блок операторов

EndReduction;,

Здесь и далее X - степень выполняемой редукции, а блок операторов содержит список операторов, для которых будет выполняться редукции производительности.

Редукция по функциональным устройствам в виде мультикадровой программы позволяет перейти от структурной реализации задачи к структурно-процедурной при недостаточном аппаратном ресурсе для структурной реализации задачи. Для реализации перехода от структурной организации вычислений к структурно-процедурной организации, обеспечивающего эф-

фективное масштабирование параллельной программы, в язык высокого уровня COLAMO вводится новая директива для описания редукции по функциональным устройствам в виде мультикадровой программы. Мультикадр – это кортеж вычислительных структур, представленных в виде подкадров, с описанием функций чтения и записи для каждого из них, переключение между которыми осуществляется каждый условный такт работы. Важным отличием мультикадра от кадра является отсутствие необходимости в глобальной синхронизации информационных потоков данных при смене вычислительных структур.

Для описания редукции производительности по устройствам в виде мультикадровой программы введена конструкция следующего вида:

#Reduction of MultiCadr X;

Блок операторов

EndReduction;,

Редукция производительности по функциональным устройствам в виде многокадровой программы применяется в случае нехватки вычислительного аппаратного ресурса для реализации редукция производительности по функциональным устройствам в виде однокадровой программы и недостаточного количества каналов памяти для реализации редукции производительности по функциональным устройствам в виде мультикадровой программы.

Редукция производительности по функциональным устройствам в виде многокадровой программы заключается в реализации вычислений с помощью неизменяемой структуры LET и нескольких кадров, в каждом из которых через структуру LET проходит поток данных, а обмен промежуточными данными осуществляется через дополнительную память. Для описания редукции по вычислительным устройствам в виде многокадровой программы с использованием конструкции Let в язык высокого уровня предлагается директива вида:

#Reduction of Let X;

Блок операторов

EndReduction;.

Одним из важнейших типов редукции, обеспечивающих возможность ресурсонезависимого программирования ВСГТ, является редукция производительности по каналам памяти. Для описания редукции по каналам памяти предлагается использование в языке COLAMO следующей директивы:

#Reduction of Channel (Type) X;,

Блок операторов

EndReduction;,

где ключевое слово «Channel» указывает на выполнение редукции по каналам памяти, а параметр Type указывает на выбранный способ редукции. Все каналы памяти можно разделить на входные и выходные каналы, редукция для которых может быть выполнена независимо друг от друга. Поэтому сокращение каналов памяти можно выполнить тремя способами:

- редукцией входных каналов (задается ключевым словом Input);
- редукцией выходных каналов (задается ключевым словом Output);
- редукцией входных и выходных каналов (задается ключевым словом All).

Редукция каналов памяти заключается в объединении каналов в один общий канал путем смешивания данных или путем последовательного объединения данных друг за другом.

Редукция производительности по разрядности направленна не на сокращение устройств в вычислительной структуре, а на сокращение разрядности обрабатываемых данных за счет использования устройств меньшей разрядности, что приводит к сокращению аппаратных затрат на реализацию вычислительной структуры. При выполнении редукции производительности по разрядности управление информационными потоками данных осуществляется мультиплексором, а сами данные подаются в вычислительную структуру со скважностью, значение которой равно степени выполняемой редукции. Для описания редукции по разрядности на язык высокого уровня COLAMO предлагается использование конструкции вида: #Reduction of Bit X;

Блок операторов

EndReduction;

Редукция производительности по частоте, в отличие от рассмотренных редукций, предназначена для замедления работы вычислительных устройств на величину, равную степени выполняемой редукции, при этом скважность подачи данных в вычислительную структуру оста-

ется неизменной. Как правило, редукция производительности по частоте применяется для согласования скоростей обработки между редуцированными и нередуцированными фрагментами информационного графа в структуре прикладной задачи. Для описания редукции производительности по частоте в язык высокого уровня предлагается конструкция вида:

```
#Reduction of Frequency X;
    Блок операторов
EndReduction;,
```

где X - степень выполняемой редукции, а блок операторов содержит список операторов, для которых будут выполняться редукции производительности по разрядности.

4. Заключение

Для эффективного программирования вычислительных систем гибридного типа предложено использование языка программирования высокого уровня COLAMO как основы разрабатываемой технологии ресурсонезависимого программирования ВСГТ. Возможность описания в едином вычислительном контуре различных форм организации параллельных вычислений в совокупности с разработанными методами редукции производительности прикладной программы позволяет обеспечить ресурсонезависимость программирования ВСГТ, т.е. возможность простой и быстрой адаптации прикладной программы под изменившуюся архитектуру или конфигурацию ВСГТ. Предложенное расширение языка COLAMO обеспечивает пользователя набором необходимых средств для быстрой разработки эффективных ресурсонезависимых масштабируемых параллельных программ в едином языковом пространстве, что снижает сложность программирования ВСГТ и повышает скорость разработки параллельных прикладных программ при рациональном использовании узлов ВСГТ с разной архитектурой.

Литература

1. И.А. Каляев, И.И. Левин, Е.А. Семерников, В.И. Шмойлов Реконфигурируемые мультиконвейерные вычислительные структуры. Изд. 2-е, перераб. и доп. Под общ. ред. И.А. Каляева. Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. 344 с.
2. Иванова Г.С. Технология программирования: Учебник для вузов. М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. 320 с.
3. Каляев И.А., Левин И.И., Дордопуло А.И., Семерников Е.А. Высокопроизводительные реконфигурируемые вычислительные системы нового поколения. Научный сервис в сети Интернет: экзафлопсное будущее. Труды Международной суперкомпьютерной конференции с элементами научной школы для молодежи (Новороссийск, 19 сентября-24 сентября 2011 г.). М.: Изд-во МГУ, 2011. С. 42-49.
4. Каляев И.А., Левин И.И., Дордопуло А.И., Семерников Е.А. Реконфигурируемые вычислительные системы на основе ПЛИС семейства Virtex-6. Параллельные вычислительные технологии 2011. Сборник трудов Международной научной конференции. Челябинск-М.: Издательский центр ЮУрГУ, 2011. - С. 203–210.
5. Kalyaev I.A., Levin I.I., Dordopulo A.I., Slasten L.M. Reconfigurable Computer Systems Based on Virtex-6 and Virtex-7 FPGAs. IFAC Proceedings Volumes, Programmable Devices and Embedded Systems, Volume №12, part №1, 2013. Pp. 210-214.
6. Igor A. Kalyaev, Ilya I. Levin, Alexey I. Dordopulo, Liuba M. Slasten. FPGA-based Reconfigurable Computer Systems. Science and Information Conference (SAI), Oct 7-Oct 9, 2013, London, UK. Pp. 148-155.

Programming of hybrid computer systems based on the performance reduction method*

A.I. Dordopulo[1], I.I. Levin[2], I.A. Kalyaev[1], V.A. Gudkov[1], A.A. Gulenok[1]

Scientific Research Institute of Multiprocessor computer systems at Southern Federal University[1], "Scientific Research Centre of Supercomputers and Neurocompuiters" Co Ltd[2]

The paper considers methods of performance reduction, which, along with the high-level programming language COLAMO, are the basis of the developed technology of resource-independent programming of hybrid computer systems consisted of reconfigurable and microprocessor computational nodes. Owing to the language COLAMO it is possible to describe various forms of organization of parallel calculations – from the structural form typical for reconfigurable computational nodes, then to structural-procedural and multiprocedural forms, and at last to procedural forms of organization of calculations, used for microprocessor nodes. Transformations between these various forms of organization of calculations for fast computer-aided adaptation of the application to the modified configuration of hardware resource of the computer system is performed with the help of the methods of performance reduction.

Keywords: performance reduction, high–level programming language, programming of hybrid computer systems, technologies of resource-independent programming.

References

1. Kalyaev I.A., Levin I.I., Semernikov E.A., Shmoilov V.I. Rekonfiguriruyemiye multikonveyerniye vichislitelniye struktury. [Reconfigurable multipipeline computing structures] 2nd edition, revised and supplemented. Edited by I.A. Kalyaev. Rostov-on-Don: SSC RAS Publishing, 2009. 344 pp.
2. Ivanova G.S. Tekhnologiya programmirovaniya [Technology of programming]: Manual. M.: Bauman MSTU Publishing, 2002. 320 pp.
3. Kalyaev I.A., Levin I.I., Dordopulo A.I, Semernikov E.A. Vysokoproizvoditelniye rekonfiguriruyemiye vichislitelniye sistemy novogo pokoleniya. [High-performance reconfigurable computer systems of the next-generation] Nauchnyi servis v seti Internet: ekzaflopsnoye budusheye. Trudy Mezhdunarodnoi superkompjuternoi konferentsii s elementami nauchnoi shkoly dlya molodezhi (Novorossisk, 19 sentyabrya-24 sentyabrya 2011). [Scientific service in the Internet: exaflops future. Proceedings of the International supercomputer conference and youth scientific school (Novorossiisk, 19 September-24 September, 2011)] M.: MSU Publishing, 2011. 42-49 pp.
4. Kalyaev I.A., Levin I.I., Dordopulo A.I., Semernikov E.A. Rekonfiguriruyemiye vichislitelniye sistemy na osnove PLIS semeistva Virtex-6. [Reconfigurable computer systems based on Virtex-6 FPGAs] Parallelniye vichislitelniye tekhnologii 2011. Sbornik trudov Mezhdunarodnoi nauchnoi konferentsii. [Parallel computer technologies 2011. Proceedings of International scientific conference] Tchelyabinsk-M.: South-UIral State University Publishing Centre, 2011. – 203–210 pp.
5. Kalyaev I.A., Levin I.I., Dordopulo A.I., Slasten L.M. Reconfigurable Computer Systems Based on Virtex-6 and Virtex-7 FPGAs. IFAC Proceedings Volumes, Programmable Devices and Embedded Systems, Volume №12, part №1, 2013. Pp. 210-214.

* This paper was financially supported in part by the Ministry of Education and Science of the Russian Federation under Grant № 14.578.21.0006 from 05.06.2014, ID RFMEFI57814X0006

6. Igor A. Kalyaev, Ilya I. Levin, Alexey I. Dordopulo, Liuba M. Slasten. FPGA-based Reconfigurable Computer Systems. Science and Information Conference (SAI), Oct 7-Oct 9, 2013, London, UK. Pp. 148-155.