

## **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РЕШЕНИЯ ВЫЧИСЛИТЕЛЬНО ТРУДОЕМКИХ ЗАДАЧ НА РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ**

А.И. Дордопуло<sup>1</sup>, И.И. Левин<sup>2</sup>

<sup>1</sup>Южный научный центр РАН, Ростов-на-Дону;

<sup>2</sup>Научно-исследовательский институт многопроцессорных вычислительных систем имени академика А.В. Каляева Южного федерального университета, Таганрог;

*Рассматривается применение комплекса системного программного обеспечения для реконфигурируемых вычислительных систем при решении вычислительно трудоемких задач математической физики*

### **Введение**

Низкая эффективность и существенный разрыв между декларируемой пиковой и реальной производительностью кластерных многопроцессорных вычислительных систем (МВС) при решении сильносвязанных задач математической физики обусловлен принципиальными недостатками кластерных МВС, такими как относительно низкая скорость процедур межпроцессорного обмена, ограниченная пропускная способность коммутационной системы, сложность синхронизации множества взаимосвязанных последовательных процессов и т.д.

В отличие от кластерных систем реконфигурируемые вычислительные системы (РВС), разрабатываемые и развиваемые более 20-ти лет в Научно-исследовательском институте многопроцессорных вычислительных систем имени академика А.В. Каляева Южного федерального университета, в том числе в рамках программ Союзного государства «СКИФ» и «ТРИАДА», обеспечивают высокую реальную производительность (не ниже 60% от пиковой) при решении реальных, а не тестовых, прикладных задач. В настоящее время по заданию Федерального агентства по науке и инновациям в рамках государственного контракта №02.524.12.4002 по теме «Создание семейства высокопроизводительных многопроцессорных вычислительных систем с динамически перестраиваемой архитектурой на основе реконфигурируемой элементной базы и их математического обеспечения для решения вычислительно трудоемких задач» для семейства РВС создается комплекс средств разработки прикладных задач.

### **1. Программирование РВС**

РВС представляет собой высокопроизводительную вычислительную систему, работающую по принципу конвейерной обработки потока данных [1] под управлением хост-компьютера (персонального компьютера). Отличительной особенностью таких систем является возможность изменения архитектуры системы на логическом уровне, что позволяет создавать вычислительный конвейер, соответствующий алгоритму обработки данных на уровне аппаратных устройств для решаемой задачи. Соответствие архитектуры системы информационному графу задачи позволяет достигать высокой реальной производительности, которая, в отличие от большинства МВС кластерной архитектуры, составляет 60-90% от пиковой на широком классе практических

прикладных задач таких областей, как математическая физика, цифровая обработка сигналов, линейная алгебра, символьная обработка.

Для создания вычислительного конвейера в РВС используется структурно-процедурный принцип организации вычислений [1,2,3]: если вычислительного ресурса системы достаточно для реализации вычислительной структуры задачи целиком, то система настраивается на вычислительную структуру задачи (структурная реализация); если же ресурса недостаточно, то задача разбивается на информационно-независимые фрагменты, каждый из которых реализуется структурно. Фрагменты последовательно выполняются на РВС под управлением единой процедуры (процедурная реализация), обмен данными между фрагментами при этом происходит через память. Разбиение информационного графа задачи на информационно-независимые фрагменты, называемые кадрами, осуществляется на основе разработанных в [2,3] методов преобразования задач в эффективную кадровую форму.

Программирование РВС значительно сложнее, чем программирование МВС, построенных на основе множества микропроцессоров, поскольку для реконфигурируемых вычислительных систем необходимо задавать как вычислительную структуру системы, так и организовывать потоки данных для вычислительного конвейера. Организация вычислительной структуры называется *структурным* программированием - создание необходимых вычислительных структур в поле логических ячеек ПЛИС и связей между ними, которые определяют устройства обработки данных для информационного графа задачи. Смена вычислительных структур в процессе решения задачи и организация потоков данных получила название *процедурного* программирования [1,3] – управление и синхронизация потоков данных, которые подаются на вход вычислительных конвейеров.

Структурная составляющая описывает аппаратную реализацию информационного графа задачи или его фрагментов и, тем самым, формирует вычислительную структуру РВС как специализированной многопроцессорной системы, ориентированной на решение данной задачи. Процедурная составляющая описывает процедуры чтения и записи данных в каналах распределенной памяти, формирует потоки данных для структурной составляющей и задает последовательность смены кадров. Программирование процедурной составляющей больше схоже с традиционной программой для многопроцессорных вычислительных систем с тем отличием, что программист оперирует не переменными, а потоками данных и организует вычислительный процесс не процедурами с последовательным выполнением операторов, а кадрами с асинхронными операциями в пределах кадра. Для структурного программирования необходимы знания в области схмотехники, для процедурного – навыки в управлении потоками данных, поскольку в отличие от программ, выполняемых на традиционных микропроцессорах и оперирующих обработкой одного данного, синхронизация потоков данных для асинхронных операций в теле кадра является нетривиальной задачей.

Эффективность вычислительного процесса при реконфигурации архитектуры РВС на структурном, схмотехническом уровне может быть повышена от 10 до 100 раз в зависимости от квалификации специалиста-схмотехника по сравнению с вычислительными системами, архитектура которых не может быть изменена. Это делает, с одной стороны, РВС чрезвычайно привлекательными с точки зрения получения максимальной производительности, а с другой стороны, существенно усложняет процесс программирования, делая его сопоставимым по сложности с созданием новой вычислительной системы и требуя усилий как минимум двух высококвалифицированных специалистов – схмотехника и программиста.

Поэтому для широкого использования РВС необходимо решить проблемы сокращения сроков создания прикладных задач для РВС и упрощения процесса программирования. Решение указанных задач можно достичь созданием комплекса системного программного обеспечения и средств разработки, предоставляющих пользователю РВС возможность организации параллельных вычислительных процессов на основе новых методов и средств и удовлетворяющих следующим требованиям:

- возможности программирования как структурной, так и процедурной составляющих на языке высокого уровня;
- наличие средств, позволяющих осуществлять реконфигурацию прикладных задач без участия высококвалифицированного схемотехника;
- обеспечению совместимости и переносимости проектов между РВС разных архитектур;
- возможности масштабирования прикладной задачи при увеличении ресурса;
- возможности удаленного использования вычислительных ресурсов РВС.

## 2. Комплекс системного программного обеспечения

На основе этих требований к системному программному обеспечению для РВС, ведется создание комплекса СПО и средств разработки[4], структура которого представлена на рис.1. По функциональному назначению комплекс системного программного обеспечения разделяется на комплекс средств разработки прикладных программ, средства администрирования вычислительных ресурсов РВС и комплекс служебных программ и драйверов.

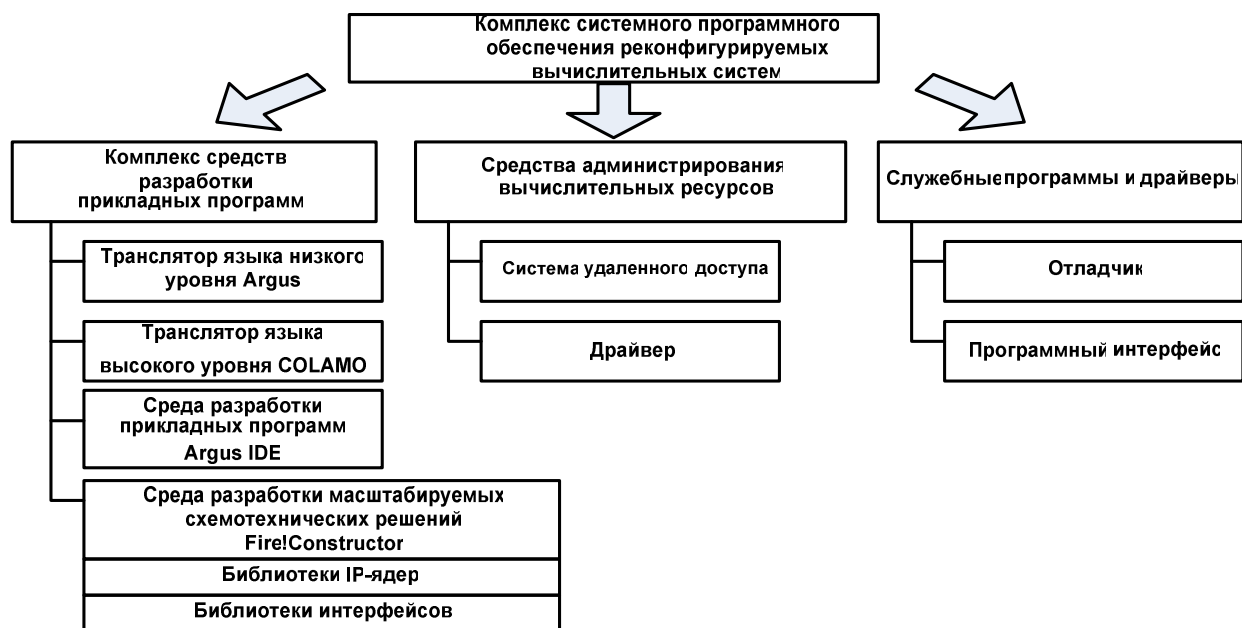


Рис.1. Структура комплекса системного программного обеспечения и средств разработки для РВС

Программный комплекс средств разработки прикладных программ содержит: язык ассемблера Argus v.3.0; язык структурно-процедурного программирования высокого уровня COLAMO v.2.0; интегрированную среду разработки аппаратно-

программных решений прикладных задач Argus IDE v.3.0, единую для всех представителей PBC и поддерживающую языки программирования Argus v.3.0 и COLAMO v.2.0; среду разработки вычислительных структур для синтеза масштабируемых параллельно-конвейерных процедур, оперирующую библиотекой схемных решений (IP-ядер).

Язык структурно-процедурного программирования высокого уровня COLAMO [2, 3,5] обеспечивает синтаксическую поддержку реконфигурации аппаратной платформы PBC и возможность использования элементов библиотеки масштабируемых IP-ядер. Транслятор COLAMO v.2.0 для всех представителей семейства PBC обеспечивает трансляцию исходного кода программы в язык ассемблера Argus v.3.0 и в VHDL посредством среды разработки масштабируемых параллельно-конвейерных процедур Fire!Constructor, создавая тем самым конфигурационные файлы для ПЛИС.

Язык структурно-процедурного программирования Argus представляет собой низкоуровневый язык (ассемблер), предназначенный для описания процедурной составляющей прикладной параллельной программы PBC [2,3] и структурной составляющей в виде коммутации универсальных макропроцессоров или специализированных схемотехнических решений. Язык Argus необходим для организации потоков данных на уровне команд контроллеров распределенной памяти и создания программ, использующих топологические и архитектурные особенности базовых модулей PBC. Программирование на языке Argus, как и на любом языке ассемблера, требует от программиста обширных знаний в области аппаратного обеспечения PBC, команд контроллеров распределенной памяти и взаимосвязей между структурными элементами системы.

Интегрированная среда разработки Argus IDE предназначена для интерактивной разработки параллельных программ на языках высокого уровня COLAMO и языке ассемблера Argus в едином языковом пространстве. Среда Argus IDE, объединяя в своем составе трансляторы языков COLAMO и Argus, обеспечивает эффективную разработку масштабируемых параллельных программ для PBC.

В среде Fire!Constructor на основе библиотеки IP-ядер и интерфейсов структурная составляющая прикладной программы распределяется по микросхемам ПЛИС с учетом связей между кристаллами и транслируется в конфигурационные файлы ПЛИС с помощью САПР схемотехнических решений Xilinx.

Средства администрирования вычислительных ресурсов и служебные программы предоставляют пользователю дополнительные возможности по управлению и использованию вычислительных ресурсов PBC.

### **3. Прикладные вычислительно трудоемкие задачи**

Создаваемый комплекс системного программного обеспечения предназначен для создания эффективных прикладных программ для PBC при решении задач различных предметных областей.

Так, при решении задачи математического моделирования гидрофизических и биогеохимических процессов в Азовском море[6], при параллельной организации вычислений на кластерной системе из 100 процессоров максимальное ускорение составит всего 1,7 раза, а с учетом издержек на пересылки данных при межпроцессорных коммуникациях ускорение может быть еще меньше. Базовый модуль PBC с частотой 200 МГц обеспечивает сокращение времени решения в 32 раза по сравнению с персональным компьютером, выигрыш по сравнению с решением рассматриваемой задачи на кластерной системе составит примерно 16 раз. Учитывая

характерный для представителей семейства РВС близкий к линейному рост производительности, решение этой же задачи на рабочей станции РВС-0.2-РС, состоящей из четырех базовых модулей, позволит сократить временные затраты для задачи гидродинамики в 128 раз по сравнению с персональным компьютером и в 45 раз по сравнению с кластерной системой.

При решении задачи моделирования движения воздушных масс и процессов их влияния на природные объекты в районе функционирования Ростовской АЭС[7], максимальное ускорение для кластерной системы на ста процессорах составит всего 5,6 раза, а с учетом издержек на пересылки данных при межпроцессорных коммуникациях ускорение может быть еще меньше (примерно в 1,5 раза), что составит 3,7 раза. Базовый модуль РВС с частотой 200 МГц обеспечивает сокращение времени решения в 48 раз по сравнению с ПЭВМ, т.е. выигрыш по сравнению с решением рассматриваемой задачи на кластерной системе составит примерно 12 раз.

Создаваемое в рамках проекта системное и прикладное программное обеспечение должно обеспечить потенциальным пользователям представителей семейства РВС удобство программирования сложных практических задач, в том числе и большого размера, решение которых на МВС традиционной архитектуры либо требует недопустимо больших временных затрат, либо вообще невозможно. Реализация прикладных задач пользователей на РВС позволяет обеспечить высокую реальную производительность системы, близкую к пиковой производительности на данной задаче, и достичь практически линейного роста производительности при наращивании аппаратного ресурса.

## Список литературы

1. Воеводин, В.В. Параллельные вычисления / В.В. Воеводин, Вл.В. Воеводин - С.-Петербург: «БХВ-Петербург», 2002. - 599 с.
2. Каляев, А.В. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений / А.В. Каляев, И.И. Левин. - М.: «Янус-К», 2003. – 380 с.
3. Каляев, И.А. Реконфигурируемые мультиконвейерные вычислительные структуры / И.А. Каляев, И.И. Левин, Е.А. Семерников, В.И. Шмойлов. – Ростов/Д: Изд-во ЮНЦ РАН, 2008. - 320 с.
4. Дордопуло, А.И. Семейство многопроцессорных вычислительных систем с динамически перестраиваемой архитектурой / А.И. Дордопуло, И.А. Каляев, И.И. Левин, Е.А. Семерников// Материалы Четвертой Международной научной молодежной школы «Высокопроизводительные вычислительные системы». - Таганрог: Изд-во ТТИ ЮФУ, 2007. – С. 68-74.
5. Левин, И.И. Язык параллельного программирования высокого уровня для структурно-процедурной организации вычислений / И.И. Левин// Труды Всероссийской научной конференции. - М.: Изд-во МГУ, 2000. – С.108-112.
6. Sukhinov, A.I. Three-dimensional models for shallow water basins and its finitedifference realization / A.I. Sukhinov, V.S. Vasiliev // Proc. of the 3rd International Conference FDS2000, September 1-4, 2000, Palanga, Lithuania, Finite difference schemes: theory and applications.– Vilnius: IMI, 2000. p.227-235.
7. Крукиер, Л.А. Математическое моделирование гидродинамики Азовского моря при реализации проектов реконструкции его экосистемы / Л.А. Крукиер// Матем. Моделирование. -1991 .Т.3. -№9. -С.3-20.