

ПРОГРАММНАЯ ИНЖЕНЕРИЯ SOFTWARE ENGINEERING

УДК 004.272.44, 004.94

Д. Н. Змеев, науч. сотр., e-mail: zmeievdn@ippm.ru,

А. В. Климов, ст. науч. сотр., e-mail: klimov@ippm.ru,

Н. Н. Левченко, канд. техн. наук, зав. отделом, e-mail: nick@ippm.ru,

А. С. Окунев, канд. техн. наук, вед. науч. сотр., e-mail: oku@ippm.ru,

А. Л. Стемповский, д-р техн. наук, акад. РАН, директор, e-mail: ippm@ippm.ru

Институт проблем проектирования в микроэлектронике РАН (ИППМ РАН), Зеленоград, Москва

Эмуляция аппаратно-программных средств параллельной потоковой вычислительной системы "Буран"

Описывается эмулятор, реализующий архитектуру параллельной потоковой вычислительной системы (ППВС) "Буран" на кластерных компьютерах. Эмулятор является составной частью программного комплекса для оценки возможности выполнения задач пользователя с использованием потоковой модели вычислений и определения оптимальной архитектуры. Приведены основные предпосылки к созданию программного эмулятора ППВС "Буран", особенности его построения, а также базовые алгоритмы организации основного цикла работы процесса эмуляции.

Ключевые слова: потоковая модель вычислений, эмулятор ППВС "Буран", спецвычислители, новая парадигма вычислений

Введение

Высокопроизводительные вычислительные системы кластерного типа демонстрируют низкую реальную производительность на большом круге актуальных задач и поэтому требуется изменение парадигмы вычислений [1].

Примером таких актуальных задач являются задачи из области молекулярной динамики, разнообразные графовые задачи или задачи, представляющие новое поколение прямых методов решения СЛАУ, соединяющих технику разреженных матриц с распараллеливанием вычислений, использующихся в новых САПР, работа над которыми ведется в ИППМ РАН [2, 3].

Еще более актуальна задача изменения парадигмы вычислений для разрабатываемых суперкомпьютерных систем, которые всегда были и остаются передовым фронтом в области вычислительной техники.

В связи с этим в ИППМ РАН продолжается работа над проектом параллельной потоковой вычислительной системы (ППВС) "Буран" [4], которая реализует оригинальную потоковую модель вычислений с динамически формируемым контекстом. Потоковая модель вычислений с динамически формируемым контекстом обладает рядом свойств, которые позволяют преодолевать проблемы, особенно остро проявляющиеся при повышении производительности компьютеров до эксафлопсного уровня [5, 6].

При создании вычислительной системы возникает необходимость в моделировании ее архитектуры. Поскольку архитектура разрабатываемой вычислительной системы принципиально отличается от традиционных систем, то отсутствует возможность использовать готовые компоненты для аппаратного моделирования, а из-за большого объема оборудования возникают трудности макетирования на ПЛИС.

Ранее была разработана поведенческая блочно-регистрационная модель (ПБРМ) ППВС "Буран" [7]. Данная модель — потактовая, с широким диапазоном настраиваемых параметров. На ПБРМ были исследованы задачи различных классов и получены данные [8–11], свидетельствующие о правильности выбора архитектурной реализации параллельной потоковой вычислительной системы и о высокой степени масштабирования программ, значительно превосходящей в этом показателе кластерные вычислительные системы.

Поскольку данная модель функционирует на рабочей станции, то у нее имеются ограничения на размерность пропускаемых задач и на производительность самой модели. Кроме того, эффект, связанный с увеличением степени масштабируемости, может быть получен только при моделировании тысяч вычислительных ядер системы.

В целях исследования и подтверждения полученных ранее результатов на ПБРМ в задачах большей размерности было решено создать эмулятор ППВС для работы на кластерных вычислительных системах.

Эмулятор ППВС является составной частью испытательного программного комплекса, предназначенного для оценки возможности выполнения задач пользователя с использованием потоковой модели вычислений и определения оптимальной архитектуры (с прогнозируемыми параметрами) для конкретного класса задач. Программный комплекс будет доступен пользователям через Internet и предоставит возможность создания параллельных программ и архитектур систем, адаптированных под определенный класс задач, как на базе ПЛИС, так и в виде заказного кристалла.

Потоковая модель вычислений

В основе потоковой модели вычислений с динамически формируемым контекстом лежит активация вычислительных квантов по готовности данных (рис. 1, см. третью сторону обложки). Вычислительный квант представляет собой программу, которая, будучи активированной, выполняется до конца без привлечения дополнительной информации, т. е. без приостановки процесса вычисления на подкачку дополнительных внешних данных. Различные вычислительные кванты между собой взаимодействуют и сохраняют состояние только через отправку сообщений (токенов), активирующих новые кванты. В узле-отправителе определяются и передаваемое значение, и адрес получателя, тем самым мы работаем в парадигме "раздачи" [12]. В этом состоит принципиальное отличие нашего подхода от традиционного (парадигмы "сбора"), когда вычислительный процесс сам запрашивает нужные ему данные из памяти или у других процессов.

Динамически формируемый контекст (ДФК) в нашей модели вычислений позволяет осуществлять одновременное и параллельное выполнение программы узла над различными вычислительными квантами, которые имеют разный контекст. Контекст представляет собой часть токена, по которому осуществляется поиск "совпадающих" токенов. Весь контекст полностью доступен для программиста. Программа узла получает всю необходимую информацию из контекста о местоположении токена и, соответственно, знает куда и какое число результирующих токенов послать в зависимости от выполняемых условий.

Для программирования в потоковой модели вычислений с ДФК создан параллельный язык DFL. Согласно принципам программирования параллельной потоковой вычислительной системы для программиста отсутствует представление о памяти в традиционном понимании, тем не менее, в нашей модели вычислений введено понятие виртуального адресного пространства токенов, размерность которого связана с разрядностью полей ключа токена.

Для ограниченного класса программ — так называемого линейного (или аффинного) — создается транслятор с последовательного языка программирования в параллельный язык DFL.

Архитектура параллельной потоковой вычислительной системы "Буран"

ППВС "Буран" представляет собой многоядерную масштабируемую вычислительную систему (рис. 2) [12, 13]. В состав каждого вычислительного ядра системы входят исполнительное устройство, процессор сопоставления, коммутатор токенов, блок хэширования. Между ядрами в системе передаются единицы информации в виде токенов. Токеном называется структура, в состав которой входят данное, контекст, который однозначно определяет положение операнда в виртуальном адресном пространстве задачи, и набор служебных полей. Токены являются основными информационными объектами, с которыми работает аппаратура вычислительной системы. Коммутация между ядрами осуществляется по номеру вычислительного ядра. Этот номер вырабатывается в блоке хэширования на основе содержимого полей токена и настраиваемой программистом функции распределения.

Процессор сопоставления в ППВС обеспечивает взаимодействие токенов различных типов в зависимости от содержимого поля токена "код операции", сравнивая поля ключей токенов с учетом кратности и оценивая состояние других полей токенов [14—16]. Ключи токенов сопоставляются в ассоциативной памяти ключей, а сами токены хранятся в памяти токенов.

Параллельная потоковая вычислительная система, реализующая потоковую модель вычислений с динамически формируемым контекстом, обладает рядом достоинств [17]:

- хорошая масштабируемость системы, что позволяет реализовать возможность создания как многоядерного кристалла, так и многопроцессорной вычислительной системы на базе таких кристаллов;
- аппаратное выявление неявного параллелизма задачи в ходе ее выполнения;
- наличие аппаратно-программных средств управления вычислительным процессом задачи;
- асинхронная работа отдельных блоков системы;

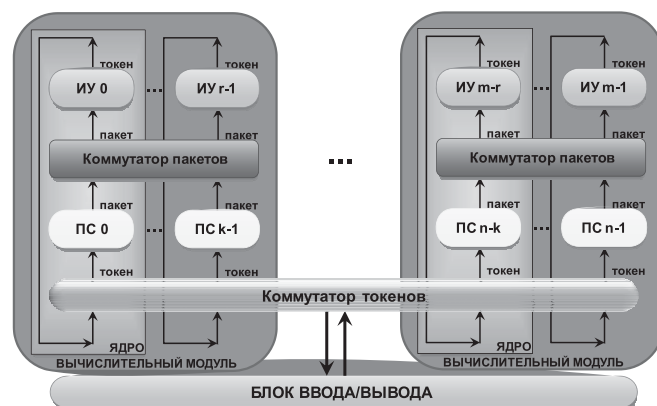


Рис. 2. Архитектура параллельной потоковой вычислительной системы "Буран"

- нивелирование задержки в коммуникационной сети за счет потоковой организации вычислительного процесса и др.

Необходимость создания эмулятора

Создание программного эмулятора параллельной потоковой вычислительной системы "Буря" преследует несколько целей.

Во-первых, эмулятор представляет собой часть аппаратно-программного комплекса (АПК). В состав АПК входят: средства параллельного языка DFL, потактовая модель ПБРМ, эмулятор ППВС, RTL-модель вычислительного ядра и вычислительного модуля, макет.

Во-вторых, эмулятор позволяет рядовым пользователям программировать в парадигме "раздачи", исследовать прохождение параллельных программ в привычной для них среде, используя при этом модель вычислений с управлением потоком данных. Также можно будет апробировать разработанные средства "аппаратного" масштабирования программ и оценить потенциал потоковой модели вычислений.

В-третьих, разработчики вычислительной системы "Буря" используют эмулятор для отработки различных вариантов аппаратных решений и получения статистической информации для оценки эффективности их внедрения.

В-четвертых, эмулятор используется для исследования работы ППВС "Буря" на большом числе вычислительных ядер для широкого круга задач с существенно большей их размерностью по сравнению с поведенческой блочно-регистрационной моделью.

В-пятых, поскольку создаваемые в нашей парадигме программы обычно обладают хорошей масштабируемостью, то несмотря на относительно высокие накладные расходы (на сам процесс эмуляции), при распараллеливании на большое число вычислительных ядер эмулятор может демонстрировать вполне конкурентную эффективность по сравнению с кластерными вычислительными системами, что для некоторых задач позволит снизить время их решения.

Ускорения работы на программном эмуляторе ППВС удалось добиться как за счет параллельной реализации модели системы, так и за счет упрощения модели, отказавшись от потактового моделирования. При этом программный эмулятор ППВС обеспечивает максимально возможную производительность, сохраняя при этом функциональность системы. Потактовое моделирование вносит синхронность в процессы, являющиеся имманентно асинхронными, что в результате препятствует распараллеливанию и значительному замедлению из-за простоев в синхронизации и ожидании обменов. Было решено отказаться от потактового моделирования еще и потому, что с точки зрения оценки адекватности моделирования отсутствие учета времени (тактов) не будет являться критичным. Для оценки адекватности моделирования существуют

такие оценки работы, как число выполненных узлов, число порожденных токенов, число токенов, переданных по сети (реальной или имитируемой). Подсчет необходимых показателей можно вести опционально, указывая необходимые непосредственно перед запуском процесса моделирования программы.

Описание эмулятора ППВС

Эмулятор ППВС является статической библиотекой на языке С с использованием интерфейса обмена сообщениями (MPI) и предназначен для работы на UNIX-системах.

Библиотеки и функции. В видимой части библиотеки доступны функции инициализации эмулятора и формирования токенов, а также параметры, через которые программист может влиять на процесс. В закрытой части реализован алгоритм сопоставления токенов и процесс их приема-передачи между вычислительными ядрами.

При разработке программы с использованием библиотеки эмулятора ППВС программист перестраивает алгоритм последовательной программы в парадигму "раздачи" [12], в которой функции не называются явно, а формируются токены с указателями на них. Каждая такая функция должна быть определенного типа. При формировании токена, например, следует пользоваться функцией DFE_SendEx или одним из ее переопределений: DFE_Send — для простого токена; DFE_MSend — для многоходового токена; DFE_GSend — для глобального токена; DFE_HSend — для выходного токена.

Запуск эмулятора ППВС. Инициализация и запуск эмулятора проводятся передачей эмулятору указателей на все узлы с помощью соответствующей функции. Это делается для того, чтобы эмулятор знал, какой узел вызывать при совпадении ключей токенов в памяти сопоставления. Далее в любом порядке устанавливаются параметры эмулятора с помощью функции формирования начальных данных и функции, учитывающей метод формирования начальных данных. Существует два варианта метода формирования токенов (начальных данных задачи):

- все токены формируются в нулевом ядре и затем рассылаются на другие ядра в соответствии с функцией распределения вычислений;
- токены формируются только в ядре, которому они "принадлежат" в соответствии с функцией распределения вычислений.

В эмуляторе реализованы три базовых функции распределения вычислений: "стандартная", "по полю" и "смещение полей". Также можно указать и свою функцию распределения вычислений, при этом единственным требованием является то, чтобы результат вычисления номера ядра был в пределах количества выделенных ядер.

Определение окончания эмуляции. Для определения окончания процесса эмуляции реализовано два способа. Первый способ — "по числу выходных токенов". В этом случае все выходные токены акку-

мулируются в нулевом ядре, и когда их число достигает установленного значения, на все ядра посылается сигнал прекращения эмуляции. Вторым способом — "определение тишины". Он реализован следующим образом: когда у нулевого ядра на входе нет токенов для обработки, оно рассылает остальным ядрам запрос об их состоянии. В качестве ответа нулевое ядро получает разность значений между полученными и отправленными токенами каждого вычислительного ядра. Если сумма этих "разностей" для всех ядер равна нулю, то это означает, что в эмуляторе не осталось необработанных токенов. Для "гарантированного" определения окончания проводится контрольный "запрос состояний ядер" и в случае повторения результата рассылается сигнал о прекращении эмуляции.

Работа эмулятора ППВС

Рабочий цикл эмулятора можно условно разбить на три этапа: инициализация, эмуляция и завершение работы.

Инициализация. На этапе инициализации эмулятора происходит:

- выделение памяти под буфера и области хранения токенов;
- заполнение таблицы сопоставления;
- исполнение пользовательской функции преобразования начальных данных в токены.

Эмуляция. Этап собственно эмуляции разделен на две части: ожидание токенов и обработка токенов. Токены, сформированные из начальных данных, поступают на вход эмулятора. Все пришедшие в ядро токены хранятся во входном буфере, реализованном по принципу FIFO. Далее из этого буфера извлекается первый токен и анализируется тип токена. Если данный токен является сигналом прекращения эмуляции, то функция незамедлительно переходит к этапу завершения работы. Если токен является выходным, то он записывается в список выходных токенов. Во всех остальных случаях токен обрабатывается функцией сопоставления.

Пришедший токен в процессе сопоставления последовательно сравнивается со всеми записанными ранее в память токенами. Сначала сравниваются их контексты с учетом комбинированной маски. При положительном результате сравниваются коды команд (в соответствии с таблицей сопоставления) и типы взаимодействия. Если оба сравнения прошли удачно, то в зависимости от кода команд выполняются те или иные действия над обоими токенами. Результатом этих действий может быть:

- удаление токена из памяти;
- формирование пакета;
- изменение токена;
- новая посылка пришедшего токена на обработку.

При формировании пакета определяется номер узла для его обработки, после чего он вызывается с указателем на пакет в качестве параметра. При выполнении узла формируются новые токены. Для

каждого такого токена с помощью функции распределения рассчитывается номер ядра. Если этот номер совпадает с текущим ядром, токен помещается во входной буфер. В противном случае он помещается в выходной буфер (свой для каждого ядра). При переполнении выходного буфера его содержимое с помощью функции MPI_Send передается в соответствующее ядро, не дожидаясь обработки оставшихся токенов, после чего буфер обнуляется. По завершении выполнения узла пришедший токен записывается в память только в том случае, если указатель на него не обнулен, а обнуление происходит при исчерпании токеном кратности.

По завершении обработки всех входных токенов, а она продолжается, пока входной буфер не опустеет, проводится контрольная рассылка содержимого выходных буферов на соответствующие ядра, после чего функция возвращается на этап ожидания токенов.

Ожидание токенов происходит при использовании блокирующей функции MPI_Probe, выход из которой означает получение списка токенов или "запроса состояний ядер". Для нулевого ядра отличие заключается в том, что "прослушивание состояния ядер" проводится неблокирующей функцией MPI_Iprobe, результатом выполнения которой является информация о наличии или отсутствии на входе каких-либо сигналов.

При получении новых токенов они последовательно переносятся из списка во входной буфер, после чего эмулятор переходит к этапу их обработки.

Завершение работы. Этап завершения работы включает в себя обработку выходных токенов, вывод на терминал статистических показателей и освобождение выделенной в процессе эмуляции памяти.

Собираемая статистика. В качестве статистических показателей на терминал выводятся следующие данные:

- общее время эмуляции;
- время формирования входных токенов;
- время выдачи токенов на HOST;
- время сопоставления;
- время выполнения узлов;
- число сопоставлений токенов;
- объем переданных данных в байтах;
- число переданных токенов.

Все временные показатели вычисляются как среднее значение для всех ядер, а количественные показатели суммируются.

Работа эмулятора на программах

Для проверки работы эмулятора ППВС выбирались задачи, которые требуют больших вычислительных ресурсов.

Традиционно для решения этих задач нужны высокомасштабируемые программы, однако большинство программ, выполняемых на высокопроизводительных системах, создаются с использованием "простого" параллелизма (разбиваются на части) и плохо масштабируются.

Это касается, например, задач молекулярной динамики. Для них существует много алгоритмов, которые пытаются распараллеливать на кластерных системах. Причем для этих алгоритмов (от "простых" в реализации, до более "трудных" классов — алгоритмов "пространственной декомпозиции") распараллеливание по данным оказалось проблематичным из-за нерегулярности этого класса задач. Алгоритм NAMD2 [18], например, написан на Charm++ и доказал свою способность распараллеливаться на тысячу ядер, однако этот алгоритм не может быть распараллелен на десятки и сотни тысяч ядер. И, вообще, известные алгоритмы для задач этой области трудно распараллеливались, так как построены на синхронизации после каждого шага.

Эксперименты с созданным эмулятором ППВС проводились на высокопроизводительных кластерах МСЦ РАН, на кластерах "Чебышев" и "Ломоносов" в МГУ. Необходимо отметить, что для прогона задач на эмуляторе ППВС DFL-код программы транслируется на язык C и сопрягается с библиотекой, реализуемой с использованием MPI.

Вместе с программой пользователем задается некая функция распределения вычислений (формула, зависящая от полей контекста). При выборе функции следует обеспечить относительную равномерность загрузки ядер и минимизировать число посылок токенов между вычислительными ядрами.

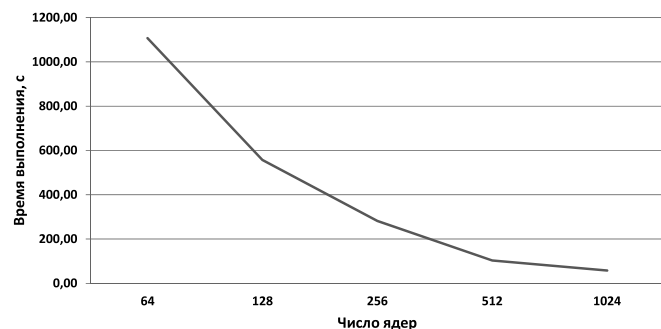


Рис. 3. Зависимость времени выполнения программы "Теплопроводность" от числа ядер

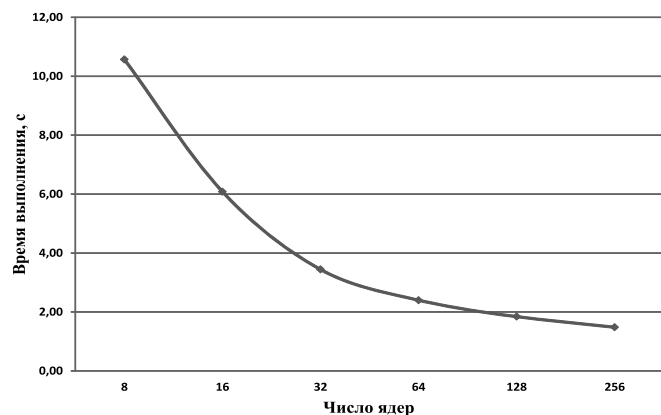


Рис. 4. Зависимость времени выполнения задачи "Молекулярная динамика" от числа ядер на суперкомпьютере "Ломоносов", 64к частиц, 100 итераций, 128 кубоидов

На ряде задач получены хорошие результаты с точки зрения повышения степени масштабируемости программ. В частности, на кластерах МГУ удалось промоделировать прохождение задач на тысячах ядер, чего не удавалось ранее. Полученные результаты анализируются. Программируются и готовятся к запуску новые актуальные задачи.

В качестве примеров приведены результаты прогона задачи "Теплопроводность" на эмуляторе (кластер "Чебышев") (рис. 3) и задачи из области молекулярной динамики [19] (суперкомпьютер "Ломоносов") (рис. 4), демонстрирующие хороший рост производительности в зависимости от увеличения числа ядер.

Для задачи из области молекулярной динамики сначала были проведены экспериментальные расчеты на потактовой регистровой модели ППВС. В силу ограничений, накладываемых этой моделью, использовали только задачи объемом 64 частицы, 256 частиц и 1024 частицы. Затем созданная программа была адаптирована (со значительным увеличением расчетных данных) для ее прохождения на эмуляторе вычислительной системы на суперкомпьютерах кластерного типа, таких как "Ломоносов" и "Чебышев", установленных в МГУ.

Эксперименты проводили на различных конфигурациях системы — от 8 до 256 вычислительных ядер. На рис. 4 видно, что с ростом числа ядер происходит практически линейное уменьшение времени выполнения задачи на эмуляторе. Это означает, что размерность самой задачи и параллелизм, заложенный в реализованном алгоритме, обеспечивают эффективное использование имеющихся вычислительных ресурсов.

На эмуляторе нашей системы на задаче с числом атомов 16К (т. е. меньшим в два раза) и даже на задаче размерностью 4К масштабирование продолжается вплоть до 256 ядер. Надо отметить, что часть вычислительной нагрузки при работе эмулятора составляют накладные расходы на организацию работы самого эмулятора.

Вообще ускорение работы средств моделирования на эмуляторе составило более трех порядков по сравнению с моделированием работы ППВС на программной блочно-регистровой модели системы, так как регистровая модель работает на обычном персональном компьютере.

Заключение

В статье описаны принципы и особенности организации эмуляции архитектуры ППВС "Буря" на кластерных компьютерах, базовые функции эмулятора и основные режимы его работы.

На основе проведенных работ уже сейчас можно сделать вывод о том, что наличие такого эмулятора ППВС и модели вычислений с управлением потоком данных может рассматриваться как альтернативная модель параллельного программирования для традиционных кластерных суперкомпьютеров.

Создаваемые в ней программы обычно обладают высокой степенью масштабируемости и потому, несмотря на относительно высокие накладные расходы, при большом числе MPI-процессов могут продемонстрировать вполне конкурентную эффективность.

Помимо этого благодаря разработке эмулятора создается также альтернативное направление для создания спецвычислителей в новой парадигме вычислений. Эмулятор позволит разработчикам оценить перспективность создания той или иной архитектуры, адаптированной под конкретную задачу в новой парадигме вычислений.

Эмулятор позволяет также перейти к макетированию, а так как в ИППМ РАН сейчас ведутся работы, связанные с оптимальным выбором состава библиотечных элементов (RTL-синтез) для создания отечественных ПЛИС [20], возможен вариант реализации макета нашей системы на таких ПЛИС.

Программистам эмулятор предоставит возможность программировать в парадигме "раздачи", пропускать эти программы, используя модель вычислений с управлением потоком данных, апробировать разработанные средства "аппаратного" масштабирования программ, а в будущем эмулятор станет частью программного комплекса определения параметров оптимальной архитектуры спецвычислителей для задач пользователей.

Разработчикам вычислительной системы эмулятор позволит обрабатывать различные варианты аппаратных решений, получать статистическую информацию и оценку эффективности от их внедрения. В перспективе планируется промоделировать с помощью эмулятора задачи, использующие модулярные алгоритмы для создания отказоустойчивых спецвычислителей [21].

Работа выполнена с использованием ресурсов суперкомпьютерного комплекса МГУ имени М. В. Ломоносова [22].

Список литературы

1. **Стемпковский А. Л., Климов А. В., Левченко Н. Н., Окунев А. С.** Методы адаптации параллельной потоковой вычислительной системы под задачи отдельных классов // Информационные технологии и вычислительные системы. 2009. № 3. С. 12–21.
2. **Гуарий М. М., Жаров М. М., Русаков С. Г., Ульянов С. Л.** Методы моделирования в частотной области периодических режимов автогенераторных схем // Информационные технологии. 2014. № 10. С. 36–43.
3. **Гуарий М. М., Жаров М. М., Русаков С. Г., Ульянов С. Л.** Методы моделирования в частотной области нелинейных радиотехнических схем // Вестник РГРТУ. 2015. № 1 (выпуск 51). С. 52–60.
4. **Донгарра Дж.** Свежий взгляд из-за океана // Суперкомпьютеры. 2013. № 2 (14). С. 6–8.
5. **Горбунов В., Эйсымонт Л.** Экзафлопсный барьер: проблемы и решения // Открытые системы. 2010. № 5. С. 12–15.
6. **Горбунов В., Елизаров Г., Эйсымонт Л.** Экзафлопсные суперкомпьютеры: достижения и перспективы // Открытые системы. 2013. № 97. С. 10–14.
7. **Левченко Н. Н., Окунев А. С., Змеев Д. Н.** Расширение возможностей поведенческой блочно-регистровой модели параллельной потоковой вычислительной системы // Матер. Пятой Международной молодежной научно-технической конференции "Высокопроизводительные вычислительные системы (ВПВС 2008)". Таганрог, Россия, 2008. С. 371–374.

8. **Левченко Н. Н., Окунев А. С.** Исследование применения многоходовых узлов в параллельной потоковой вычислительной системе для задач фильтрации // Матер. Пятой Международной молодежной научно-технической конференции "Высокопроизводительные вычислительные системы (ВПВС 2008)". Таганрог, Россия, 2008. С. 367–370.

9. **Левченко Н. Н., Окунев А. С., Климов А. В.** Реализация задачи "Повышение контрастности изображения" на параллельной потоковой вычислительной системе // Матер. Международной научно-технической конференции "Многопроцессорные вычислительные и управляющие системы (МВУС—2009)". Дивноморское, Россия, ТТИ ЮФУ, 2009. Т. 1. С. 197–200.

10. **Левченко Н. Н., Окунев А. С.** Анализ прохождения задачи "обнаружение дефектов" на параллельной потоковой вычислительной системе // Труды Третьей Всероссийской научной конференции "Методы и средства обработки информации". М.: МАКС Пресс, 2009. С. 346–350.

11. **Андреев П. П., Левченко Н. Н., Окунев А. С., Цветков В. В.** Реализация алгоритма сжатия изображения jpeg на параллельной потоковой вычислительной системе // Матер. Международной научно-технической конференции "Многопроцессорные вычислительные и управляющие системы (МВУС—2009)". Дивноморское, Россия, ТТИ ЮФУ, 2009. Т. 1. С. 126–128.

12. **Климов Арк. В., Окунев А. С.** Потоковая модель вычислений как путь к экзафлопсу // Электрон. изд.: Тр. междунар. суперкомпьютерной конференции "Научный сервис в сети ИНТЕРНЕТ: экзафлопсное будущее", Новороссийск, 19–24 сентября 2011 г., [Электронный ресурс]. М.: Изд-во МГУ, 2011. С. 261–266. URL: <http://agora.guru.ru/abrau2011/pdf/261.pdf> (дата обращения: 05.05.2015).

13. **Климов Арк. В., Левченко Н. Н., Окунев А. С.** Модель вычислений с управлением потоком данных как средство решения проблем больших распределенных систем // Матер. Второй Всероссийской научно-технической конференции "Суперкомпьютерные технологии" (СКТ-2012). Дивноморское, Геленджикский район, 24–29 сентября 2012 года. 2012.

14. **Левченко Н. Н., Окунев А. С., Яхонтов Д. Е.** Исследование работы процессора сопоставления параллельной потоковой вычислительной системы "Буран" // Всероссийская научно-техническая конф. "Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС — 2012): сб. тр. М.: ИППМ РАН, 2012. С. 467–470.

15. **Левченко Н. Н., Окунев А. С., Шурчков И. О., Яхонтов Д. Е.** Варианты реализации контроллера памяти параллельной потоковой вычислительной системы для работы с векторными и многоходовыми узлами // Всероссийская научно-техническая конф. "Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС — 2012): сб. тр. М.: ИППМ РАН, 2012. С. 463–466.

16. **Змеев Д. Н., Левченко Н. Н., Окунев А. С., Климов А. В.** Архитектура планировщика процессора сопоставления ППВС "Буран" // Всероссийская научно-техн. конф. "Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС): сб. трудов. М.: ИППМ РАН, 2014. № 4. С. 75–78.

17. **Климов Арк. В., Левченко Н. Н., Окунев А. С.** Преимущества потоковой модели вычислений в условиях неоднородных сетей // Информационные технологии и вычислительные системы. 2012. № 2. С. 36–45.

18. **James C. Phillips, Rosemary Braun et al.** Scalable molecular dynamics with NAMD // Journal of Computational Chemistry. 2005. Vol. 26. P. 1781–1802.

19. **Климов А. В., Левченко Н. Н., Окунев А. С., Змеев Д. Н., Стемпковский А. Л.** Исследование возможности асинхронной реализации задачи молекулярной динамики на ППВС "Буран" // Качество. Инновации. Образование. 2014. № 10. С. 46–51.

20. **Гаврилов С. В., Глебов А. Л., Стемпковский А. Л.** Структурная оптимизация цифровых КМОП схем // Информационные технологии и вычислительные системы. 2002. № 4. С. 40–47.

21. **Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А., Соловьев А. Н., Мячиков М. В.** Моделирование возникновения неисправностей для оценки надежностных характеристик логических схем // Информационные технологии. 2014. № 11. С. 30–36.

22. **Воеводин Вл. В., Жуматий С. А., Соболев С. И., Антонов А. С., Брызгалов П. А., Никитенко Д. А., Стефанов К. С., Воеводин Вад. В.** Практика суперкомпьютера "Ломоносов" // Открытые системы. 2012. № 7. С. 36–39.

D. N. Zmejev, Researcher, e-mail: zmejevdn@ippm.ru,
A. V. Klimov, Senior Researcher, e-mail: klimov@ippm.ru,
N. N. Levchenko, Ph. D., Head of Department, e-mail: nick@ippm.ru,
A. S. Okunev, Ph. D., Leading Researcher, e-mail: oku@ippm.ru,
A. L. Stempkovsky, Academician, Director of IDPM RAS, e-mail: ippm@ippm.ru
Institute for Design Problems in Microelectronics of Russian Academy of Sciences

Emulation on Hardware and Software of the Parallel Computing Streaming System "Buran"

The article describes the emulator, that implements the architecture of parallel dataflow computing system (PDCS) "Buran" on cluster computers. The emulator is a part of the software complex to assess the possibility of tasks execution by using a parallel dataflow computing system and to determine the optimal architecture. The article describes the main preconditions for the creation of a software emulator of PDCS "Buran", construction features, as well as basic algorithms of the organization of the main cycle of the emulation process.

Keywords: dataflow computation model, emulator of PDCS "Buran", special calculator, new paradigm of computing

References

1. Stempkovskij A. L., Klimov A. V., Levchenko N. N., Okunev A. S. Metody adaptacii parallel'noj potokovoj vychislitel'noj sistemy pod zadachi otdel'nyh klassov. *Informacionnye tehnologii i vychislitel'nye sistemy*, 2009, no. 3, pp. 12–21 (in Russian).
2. Gurarij M. M., Zharov M. M., Rusakov S. G., Ul'janov S. L. Metody modelirovaniya v chastotnoj oblasti periodicheskikh rezhimov avtogeneratornyh shem. *Informacionnye tehnologii*, 2014, no. 10, pp. 36–43 (in Russian).
3. Gurarij M. M., Zharov M. M., Rusakov S. G., Ul'janov S. L. Metody modelirovaniya v chastotnoj oblasti nelinejnyh radiotekhnicheskikh shem. *Vestnik RGRU*, 2015, no. 1, pp. 52–60 (in Russian).
4. Dongarra Dzh. Svezhij vzgljad iz-za okeana. *Superkomp'yutery*. 2013, no. 2 (14), pp. 6–8 (in Russian).
5. Gorbunov V., Jejsymont L. Ekzaflopsnyj bar'er: problemy i reshenija. *Otkrytie sistemy*. 2010, no. 5, pp. 12–15 (in Russian).
6. Gorbunov V., Elizarov G., Jejsymont L. Jekzaflopsnye superkomp'yutery: dostizhenija i perspektivy. *Otkrytie sistemy*, 2013, no. 07, pp. 10–14 (in Russian).
7. Levchenko N. N., Okunev A. S., Zmeev D. N. Rasshirenie vozmozhnostej povedencheskoj blochno-registrovoj modeli parallel'noj potokovoj vychislitel'noj sistemy. *Materialy Pjatoj Mezhdunarodnoj molodezhnoj nauchno-tehnicheskaj konferencii "Vysokoproizvoditel'nye vychislitel'nye sistemy (VPVS 2008)"*, Taganrog, Russia, 2008, pp. 371–374 (in Russian).
8. Levchenko N. N., Okunev A. S. Issledovanie primeneniya mnogovhodovyh uzlov v parallel'noj potokovoj vychislitel'noj sisteme dlja zadach fil'tracii. *Materialy Pjatoj Mezhdunarodnoj molodezhnoj nauchno-tehnicheskaj konferencii "Vysokoproizvoditel'nye vychislitel'nye sistemy (VPVS 2008)"*, Taganrog, Russia, 2008, pp. 367–370 (in Russian).
9. Levchenko N. N., Okunev A. S., Klimov A. V. Realizacija zadachi "Povyshenie kontrastnosti izobrazhenija" na parallel'noj potokovoj vychislitel'noj sisteme. *Materialy Mezhdunarodnoj nauchno-tehnicheskaj konferencii "Mnogoprocessornye vychislitel'nye i upravljajushhie sistemy (MVUS-2009)"*, Divnomorsk, Russia, 2009, vol. 1, pp. 197–200 (in Russian).
10. Levchenko N. N., Okunev A. S. Analiz prohozhdenija zadachi "obnaruzhenie defektov" na parallel'noj potokovoj vychislitel'noj sisteme. *Trudy Tret'ej Vserossijskoj nauchnoj konferencii "Metody i sredstva obrabotki informacii"*, Moscow, 2009, pp. 346–350 (in Russian).
11. Andreev P. P., Levchenko N. N., Okunev A. S., Cvetkov V. V. Realizacija algoritma szhatija izobrazhenija jpeg na parallel'noj potokovoj vychislitel'noj sisteme. *Materialy Mezhdunarodnoj nauchno-tehnicheskaj konferencii "Mnogoprocessornye vychislitel'nye i upravljajushhie sistemy (MVUS-2009)"*, Divnomorsk, Russia, 2009, vol. 1, pp. 126–128 (in Russian).
12. Klimov Ark. V., Okunev A. S. Jelektron. izd.: Potokovaja model' vychislenij kak put' k jekzaflopsu. *Trudy mezhdunarodnoj superkomp'yuternoj konferencii "Nauchnyj servis v seti INTERNET: jekzaflopsnoe budushhee"*, 2011, URL: <http://agora.guru.ru/abrau2011/pdf/261.pdf> (accessed: 05.05.2015) (in Russian).
13. Klimov Ark. V., Levchenko N. N., Okunev A. S. Model' vychislenij s upravleniem potokom dannyh kak sredstvo reshenija problem bol'shih raspredelennyh system. *Materialy Vtoroj Vserossijskoj nauchno-tehnicheskaj konferencii "Superkomp'yuternye tehnologii" (SKT-2012)*, Divnomorsk, Russia, 2012 (in Russian).
14. Levchenko N. N., Okunev A. S., Jahontov D. E. Issledovanie raboty processora sopostavlenija parallel'noj potokovoj vychislitel'noj sistemy "Buran". *Vserossijskaja nauchno-tehnicheskaja konferencija "Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem (MJeS - 2012)"*: sbornik trudov, 2012, pp. 467–470 (in Russian).
15. Levchenko N. N., Okunev A. S., Shurchkov I. O., Jahontov D. E. Varianty realizacii kontrollera pamjati parallel'noj potokovoj vychislitel'noj sistemy dlja raboty s vektornymi i mnogovhodovymi uzlamy. *Vserossijskaja nauchno-tehnicheskaja konferencija "Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem (MJeS - 2012)"*: sbornik trudov, 2012, pp. 463–466 (in Russian).
16. Zmeev D. N., Levchenko N. N., Okunev A. S., Klimov A. V. Arhitektura planirovshhika processora sopostavlenija PPVS "Buran". *Vserossijskaja nauchno-tehnicheskaja konferencija "Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem (MJeS)"*: sbornik trudov, 2014, no. 4, pp. 75–78 (in Russian).
17. Klimov Ark. V., Levchenko N. N., Okunev A. S. Preimushhestva potokovoj modeli vychislenij v uslovijah neodnorodnyh setej. *Informacionnye tehnologii i vychislitel'nye sistemy*, 2012, no. 2, pp. 36–45 (in Russian).
18. James C. Phillips, Rosemary Braun et al. Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry*, 2005, vol. 26, pp. 1781–1802.
19. Klimov A. V., Levchenko N. N., Okunev A. S., Zmeev D. N., Stempkovskij A. L. Issledovanie vozmozhnosti asinhronnoj realizacii zadachi molekulyarnoj dinamiki na PPVS "Buran". *Kachestvo. Innovacii. Obrazovanie*, 2014, no. 10, pp. 46–51 (in Russian).
20. Gavrilov S. V., Glebov A. L., Stempkovskij A. L. Strukturnaja optimizacija cifrovych KMOP shem. *Informacionnye tehnologii i vychislitel'nye sistemy*, 2002, no. 4, pp. 40–47 (in Russian).
21. Stempkovskij A. L., Tel'puhov D. V., Solov'ev R. A., Solov'ev A. N., Mjachikov M. V. Modelirovanie vzniknovenija neispravnostej dlja ocenki nadezhnostnyh harakteristik logicheskikh shem. *Informacionnye tehnologii*, 2014, no. 11, pp. 30–36 (in Russian).
22. Voevodin V. V., Zhumatij S. A., Sobolev S. I., Antonov A. S., Bryzgalov P. A., Nikitenko D. A., Stefanov K. S., Voevodin Vad. V. Praktika superkomp'yutera "Lomonosov". *Otkrytie sistemy*, 2012, no. 7, pp. 36–39 (in Russian).