УДК 519.6

ВЫЧИСЛИТЕЛЬНЫЙ ПОЛИГОН КАК СРЕДСТВО ИССЛЕДОВАНИЯ ПРОГРАММНО-АППАРАТНЫХ ПЛАТФОРМ

А. С. Антонов[1], **Б. Ю. Крысанов**[1]

Вычислительный полигон предоставляет средства для оперативного доступа по сети Интернет к вычислительным ресурсам с целью проведения небольших предварительных экспериментов и выбора программно-аппаратной платформы. Вычислительный полигон применяется для проектирования оптимальной программно-аппаратной платформы под некоторый класс задач, оперативного тестирования новых кластерных установок, а также используется в учебном процессе. В статье приводятся результаты сравнения различных кластеров НИВЦ МГУ, полученные на основе тестовых программ Вычислительного полигона.

1. Введение. Во многих современных областях науки, где традиционного последовательного способа обработки данных недостаточно и необходимо существенно ускорить процесс вычислений, все большее применение находят параллельные вычисления. Однако практически все ученые-прикладники, работающие в той или иной среде параллельных вычислений, сталкиваются с общей проблемой: как эффективно использовать параллельные компьютеры? Ответ на данный вопрос оказывается весьма и весьма не простым. Любая параллельная вычислительная среда обладает своими характерными особенностями, для работы в ней, во многих случаях, нужно знать технологии параллельного программирования, параллельные методы решения задач, архитектуру параллельных компьютеров, методы анализа структуры программ и алгоритмов и другие смежные дисциплины, не характерные для предметной области ученого-прикладника: химика, биолога, физика, медика. Зачастую оказывается невозможно преодолеть возникающие трудности без помощи специалистов в области высокопроизводительных вычислений.

Учебно-научный центр МГУ по высокопроизводительным вычислениям [1] обладает набором высокопроизводительных вычислительных ресурсов (`http://parallel.ru/cluster`), для эффективного использования которых требуется значительный объем специальных знаний в области параллельных вычислений (рис. 1). Непростой задачей является также выбор наиболее подходящей платформы для конкретной задачи пользователя.

Специалисты Центра оказывают посильную помощь пользователям в освоении как вычислительных ресурсов, так и параллельных технологий. Для этого предназначена система онлайн и оффлайн консультаций [2], предоставление документации и учебных материалов, чтение учебных курсов и т.д.

Однако очевидно, что невозможно оказать эффективную помощь каждому пользователю с каждой возникающей у него задачей. Необходимо создать механизм, который помог бы максимально облегчить специалисту-прикладнику понимание путей разрешения совершенно новых проблем, встающих перед ним при знакомстве с высокопроизводительной техникой. Одним из способов такой помощи является предоставление пользователю возможности исследовать эффективность выполнения на различных программно-аппаратных платформах критических фрагментов программ. Такая возможность реализуется средствами Вычислительного полигона [3], использующего вычислительные ресурсы НИВЦ МГУ.

2. Описание функционирования и использования полигона. Вычислительный полигон (Интернет-адрес `http://parallel.ru/polygon/`) предназначен для оперативного доступа к вычислительным ресурсам с целью проведения небольших предварительных экспериментов и выбора программно-аппаратной платформы. Главное, что нужно понять пользователю в начале работы, это на какой тип вычислителя ориентироваться. Основных параметров несколько: выбор между компьютерами с общей или распределенной памятью, соотношение между скоростью процессоров и скоростью обмена данными в коммуникационной среде, выбор технологии параллельного программирования, выбор алгоритмического подхода и некоторые другие.

Запуская при помощи Вычислительного полигона тестовые задачи на предоставляемых программно-аппаратных платформах, пользователь может оценить эффективность реализации тех или иных коммуникационных схем при использовании различных сетевых технологий (на настоящий момент Scalable

[1] Научно-исследовательский вычислительный центр, Московский государственный университет им. М.В. Ломоносова, 119992, Москва; e-mail: asa@parallel.ru

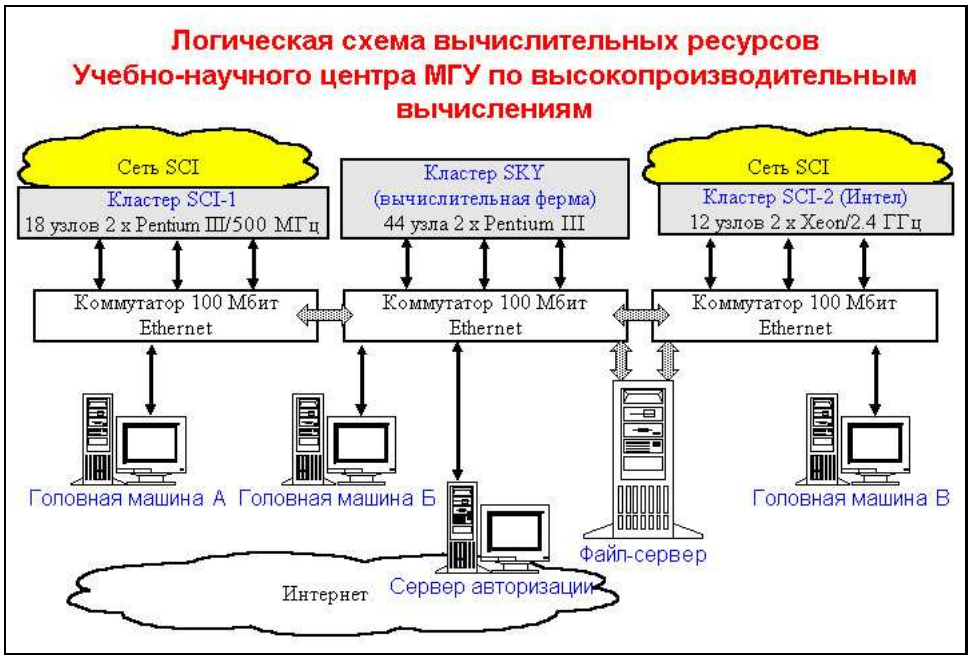

Рис. 1. Высокопроизводительные вычислительные ресурсы НИВЦ МГУ

Coherent Interface и Fast Ethernet). Кроме того, можно сравнить эффективность различных конструкций, предоставляемых технологиями параллельного программирования (например, различных вариантов пересылок или глобальных операций в MPI), а в конечном итоге, постараться оценить предполагаемую эффективность возможных параллельных реализаций собственной задачи пользователя на предоставляемых вычислительных ресурсах.

На данный момент в Вычислительном полигоне реализованы следующие типовые алгоритмические структуры межпроцессорного взаимодействия: моделирование вычислений на сеточной области, пересылка данных от каждого процессора каждому, пересылка данных по кольцу, двунаправленный обмен данными, быстрое преобразование Фурье, барьерная синхронизация, время на пересылку небольших сообщений.

Перед началом работы с Вычислительным полигоном пользователь должен ввести свой адрес электронной почты. В дальнейшем идентификация пользователя производится как по e-mail, так и IP-адресу. Используя web-интерфейс, пользователь выбирает интересующее его тестовое приложение, формирует запрос на выбор целевой программно-аппаратной среды и задает параметры для выполнения приложения на выбранном вычислителе. Опираясь как на доступный парк вычислительных систем Центра, так и на возможность оперативного управления распределением заданий между системами, производится запуск тестового приложения и возврат результатов пользователю.

На данный момент в качестве аппаратных платформ Вычислительного полигона НИВЦ МГУ предоставляются два вычислительных кластера — SCI (36 процессоров Pentium III/500, соединенных высокоскоростной коммуникационной сетью SCI) и SKY (88 процессоров Pentium III/850–1000, соединенных коммуникационной сетью Fast Ethernet). Задача с нужными параметрами передается на выполнение системе очередей Cleo (`http://parallel.ru/cluster/batch.html`) на выбранном кластере. В зависимости от заданного пользователем режима получения результатов (непосредственно в окне браузера или по указанному адресу электронной почты) Web-интерфейс либо дожидается постановки задачи на выполнение и получения результатов, либо только информирует пользователя о постановке его задачи на выполнение.

Естественно, на запуск программ в таком режиме наложен целый ряд ограничений административного характера. Так, все задачи, выбираемые в качестве типовых алгоритмических структур, достаточно просты и даже при максимально допускаемых значениях параметров выполняются не более нескольких минут; количество предоставляемых под эти задачи процессоров жестко ограничено сверху, задачи ставятся в очередь со стандартным приоритетом и т.д. Для предотвращения намеренной или случайной перегрузки вычислительных ресурсов многократной постановкой задач в очередь предусмотрена блокировка приема запросов к Web-интерфейсу с одного IP-адреса на определенный промежуток времени.

Ведется полный протокол использования Вычислительного полигона всеми пользователями.

Функциональность Вычислительного полигона на базе вычислительных ресурсов НИВЦ МГУ реализована CGI-скриптами, написанными на зыке Perl, Web-сервер — Apache, платформа — RedHat Linux, для постановки задач на исполнение используется система очередей Cleo.

3. Типовые алгоритмические структуры. В Вычислительном полигоне реализованы следующие структуры.

1) *Моделирование вычислений на сеточной области.* Задана произвольная матрица и некоторая функция, вычисляющая элемент матрицы по значениям соседних элементов. Требуется равномерно распределить вычисления между заданным числом процессоров и определенное число раз пересчитать все элементы матрицы по заданным формулам, которые для ширины граничного слоя 1 имеют вид

$$A(i,j) = (A(i-1,j) + A(i+1,j) + A(i,j-1) + A(i,j+1))/4.$$

Для выполнения задачи создается прямоугольная решетка процессоров, каждый из них получает свою часть исходной матрицы и производит вычисления над соответствующими элементами. Пересылки данных требуются между соседними процессорами по обоим измерениям, причем объем необходимых пересылок зависит от ширины граничного слоя данных.

2) *Пересылка от каждого процессора каждому.* Каждый процесс приложения посылает свою порцию данных всем остальным процессам. Моделирование данной операции производится вызовами функций библиотеки MPI десятью способами, включающими использование стандартных функций `MPI_Bcast`, `MPI_Allgather`, а также различные варианты использования неблокирующих посылок и приемов данных.

3) *Пересылки по кольцу.* Данная программа моделирует различные алгоритмы передачи сообщений между процессорами по кольцевой топологии. Моделируются три способа коммуникаций процессоров:

— последовательная передача данных по кольцу при помощи блокирующих функций;

— одновременный однонаправленный сдвиг по кольцу при помощи неблокирующих функций;

— одновременный двунаправленный сдвиг по кольцу при помощи неблокирующих функций.

4) *Двунаправленный обмен данными.* Программа позволяет исследовать двунаправленную передачу сообщений между двумя процессорами, находящимися на одном или на разных узлах кластера, при помощи различных функций MPI:

— совмещенный прием/передача сообщений (`MPI_Sendrecv`);

— прием/передача сообщений с блокировкой (`MPI_Send`, `MPI_Recv`);

— прием/передача сообщений без блокировки (`MPI_Isend`, `MPI_Irecv`);

— буферная посылка данных (`MPI_Bsend`, `MPI_Irecv`);

— синхронная посылка данных (`MPI_Ssend`, `MPI_Irecv`);

— посылка данных по готовности (`MPI_Rsend`, `MPI_Irecv`).

5) *Быстрое преобразование Фурье.*

6) *Барьерная синхронизация.*

7) *Время на пересылку небольших сообщений.*

4. Исследование результатов запуска тестовых программ. Результаты прогона программ Вычислительного полигона получены на двух кластерах НИВЦ МГУ (SCI и SKY).

Первая программа хорошо демонстрирует масштабируемость задачи, моделирующей реальную схему коммуникаций при вычислениях в узлах двумерной решетки. Так, на графике (рис. 2) хорошо видно, как уменьшается время на выполнение данной программы в зависимости от используемого количества процессоров.

Этот же рисунок полезен для сравнения двух использовавшихся кластеров. Хорошо видно, что кластер SKY имеет более мощные процессоры, чем кластер SCI, так как затрачивает на вычисления в узлах меньше времени. Но сеть кластера SCI значительно более высокоскоростная, поэтому на коммуникации кластер SCI затрачивает гораздо меньше времени. Очевидно, что в данном случае второе оказывается более важным и суммарное время выполнения программы на кластере SCI значительно меньше, чем на кластере SKY.

Естественно, что мощность процессоров кластера SKY начинает сказываться при увеличении размера матрицы или усложнении вычислений в каждом узле, когда каждому процессору достается больше выполняемых операций. Напротив, при увеличении нагрузки на коммуникационную сеть (за счет увеличения объема или количества пересылаемых данных) начинает сказываться производительность коммуникационной сети кластера SCI. График на рис. 3 демонстрирует этот эффект.

Поскольку в рассматриваемом примере размер матрицы достаточно большой (3000×3000), то при небольшом объеме пересылаемых данных (что в данном случае задается шириной граничного слоя) кластер

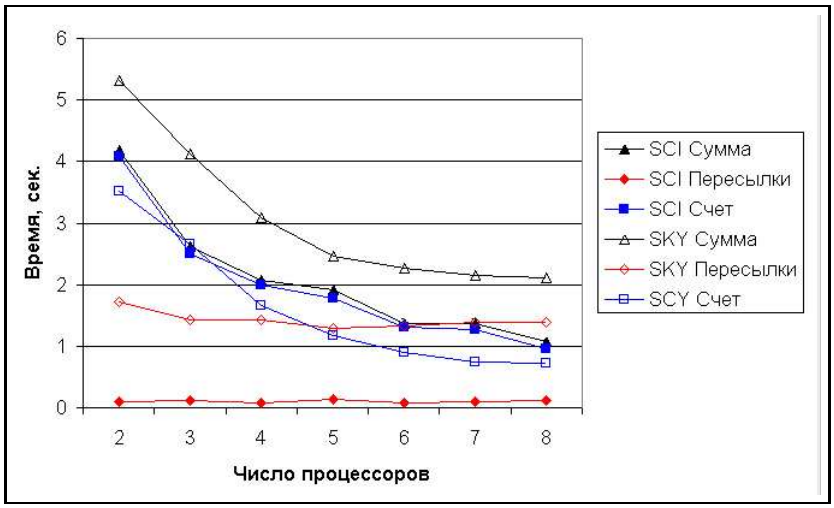

Рис. 2. Первая задача, сравнение кластеров SCI и SKY (размер матрицы 1000×1000, ширина граничного слоя 1)

Рис. 3. Первая задача, сравнение двух кластеров в зависимости от ширины граничного слоя (матрица 3000×3000, 8 процессоров)

SKY выполняет задачу быстрее, но при увеличении ширины граничного слоя уже кластер SCI значительно выходит вперед по общему времени выполнения программы.

Вторая программа вычислительного полигона позволяет не только выполнять сравнение различных программно-аппаратных сред на операции пересылки от каждого процессора каждому, но и сравнить возможные способы реализации такой операции при помощи различных функций MPI. Так, рис. 4 демонстрирует сравнение некоторых способов реализации пересылки от каждого процессора каждому на кластере SKY.

Хорошо видно, что значительно опережает все остальные способы использование стандартной функции `MPI_Allgather` (график 6). Также достаточно неплохо пользоваться другой стандартной функцией `MPI_Broadcast` (график 5).

Остальные графики демонстрируют различные варианты использования для рассылки от каждого процессора каждому операций пересылки данных типа “точка–точка”. Очевидно, что использовать эти операции можно по-разному и что получаемый эффект при этом также сильно различается. Так, использование пересылок в строго фиксированном порядке, начиная с нулевого процесса и до последнего (график 4а), является далеко не лучшим вариантом, поскольку сначала выстраивается очередь сообще-

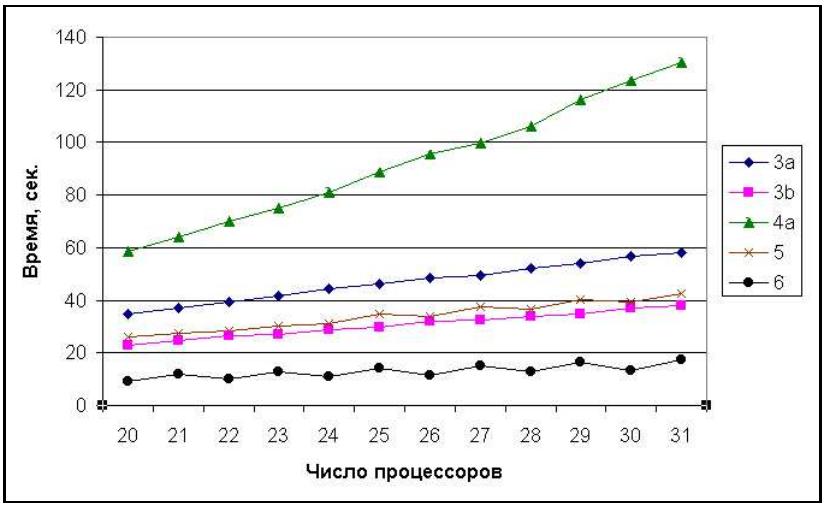

Рис. 4. Пересылка от каждого процессора каждому, кластер SKY
(длина — сообщений 400000 элементов типа double)

ний для нулевого процесса, затем для первого и т.д. Гораздо лучше распределить нагрузку на сеть при помощи посылок сообщений в другом порядке (например, как в варианте 3b, когда каждый процесс начинает коммуникации с посылки сообщения процессу с номером на единицу больше и далее в порядке увеличения номеров процессов).

Третья программа вычислительного полигона демонстрирует общение процессоров в кольцевой топологии. Результаты, полученные на кластере SCI, представлены на рис. 5.

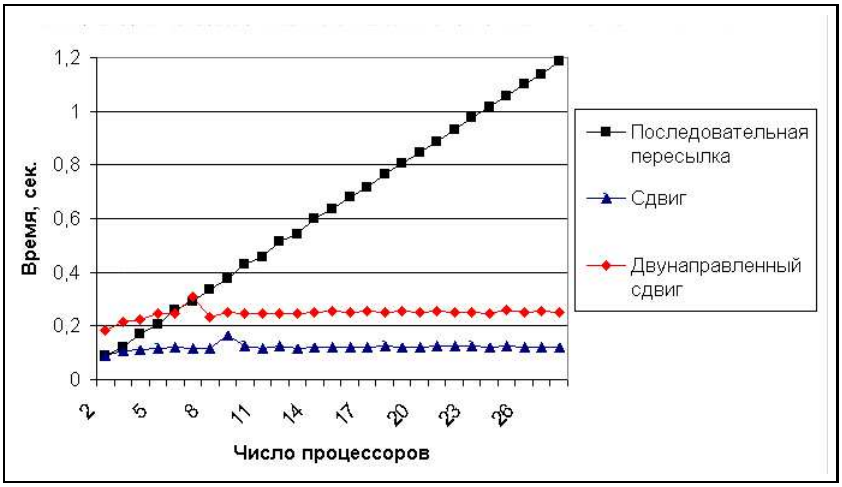

Рис. 5. Пересылки в кольцевой топологии, кластер SCI
(длина сообщений — 400000 элементов типа double)

Время на последовательную пересылку по кольцу, как и следовало ожидать, растет практически линейно с ростом числа процессоров. Времена же на циклические сдвиги в кольцевой топологии от числа процессоров практически не зависят, причем двунаправленный сдвиг требует практически точно в два раза больше времени, чем однонаправленный.

Четвертая программа вычислительного полигона позволяет сравнить эффективность разных способов общения двух выделенных процессоров. На рис. 6 приведены некоторые результаты, полученные на кластере SCI. Из этого рисунка видно, что лучшим способом обмена данными двух выделенных процессоров является стандартная функция `MPI_Sendrecv`, а наиболее долгим вариантом из различных типов операции `MPI_Send` является буферизованная посылка при помощи функции `MPI_Bsend`, предусматриваю-

щая предварительное копирование отправляемого сообщения в выделенный буфер.

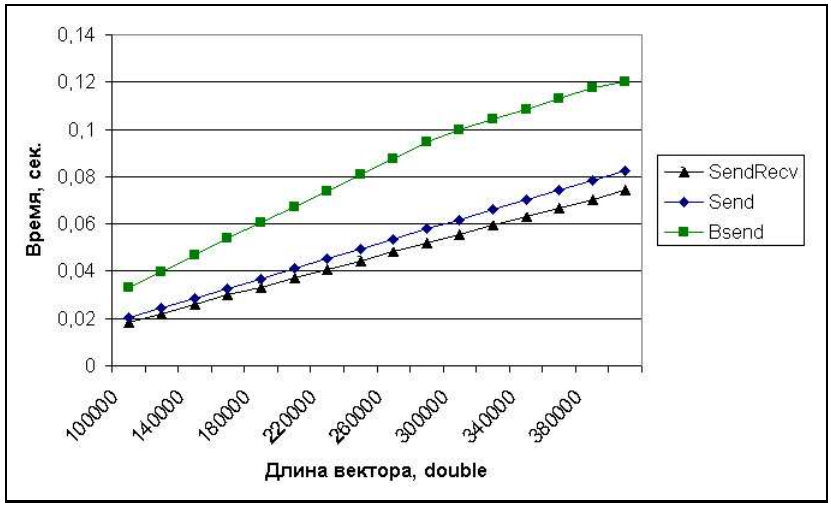

Рис. 6. Обмен между процессорами одного узла кластера SCI

Рис. 7 демонстрирует сравнение вариантов обмена данными между двумя выделенными процессорами кластеров SCI и SKY, когда оба процессора находятся на одном узле или на разных узлах. Это позволяет сравнить не только коммуникационные сети двух кластеров между собой, но и сравнить их использование с общением процессоров через общую память.

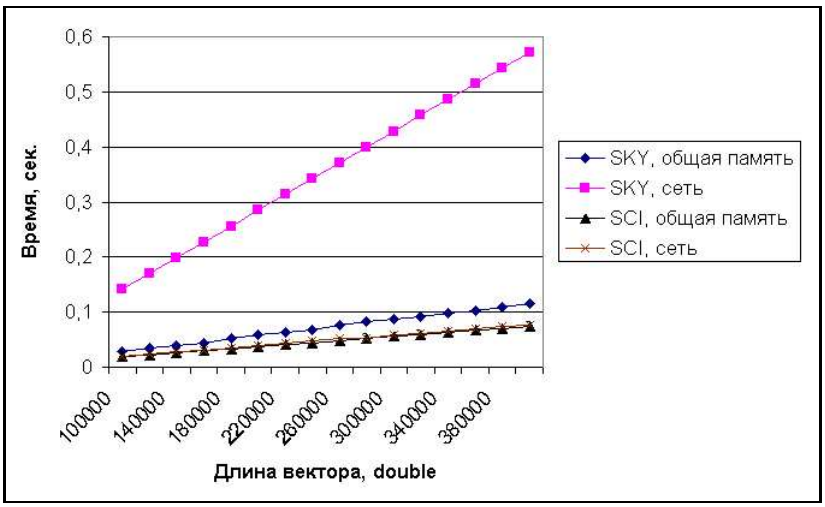

Рис. 7. Операция `MPI_Sendrecv` на кластерах SCI и SKY по сети или через общую память

Из графиков видно, что использование коммуникационной сети SCI только незначительно дольше общения по общей памяти, а вот использование сети Fast Ethernet (на кластере SKY) увеличивает время обмена данными примерно в шесть раз по сравнению с общением процессоров через общую память.

5. Заключение. Вычислительный полигон применяется для проектирования оптимальной программно-аппаратной платформы под некоторый класс задач и для оперативного тестирования новых кластерных установок, а также используется в учебном процессе. Вычислительный полигон находится в состоянии постоянного развития и совершенствования. Улучшается интерфейс пользователя, расширяется функциональность, добавляются возможности администрирования. В скором будущем намечается увеличение количества разнотипных программно-аппаратных платформ Вычислительного полигона за счет добавления новых кластеров, многопроцессорных компьютеров с общей памятью, сетевых технологий.

Очевидно, что при помощи простых тестовых фрагментов невозможно удовлетворить вычислительные потребности пользователей, однако в данном случае важна принципиальная возможность доступа к различным вычислительным ресурсам, позволяющая пользователю "пощупать" платформы и правильно сориентироваться в их выборе и использовании. Такой подход можно назвать "тестированием без программирования", он может быть одним из шагов на пути создания систем "вычислений без программирования", которые предоставляют пользователю набор универсальных вычислительных программ, формируя наборы параметров для которых, можно производить необходимые вычисления в той или иной предметной области. Подобный подход реализуется, например, в совместной разработке Тверского государственного технического университета и Санкт-Петербургского государственного политехнического университета для решения задач вычислительной гидродинамики [4].

В более отдаленной перспективе видится использование разработок, подобных Вычислительному полигону, в качестве точек входа пользователей в глобальные сети метакомпьютинга [5].

Работа выполнена при поддержке РФФИ, грант 02–07–90442.

СПИСОК ЛИТЕРАТУРЫ

1. *Воеводин Вл.В.* Проект профессионального центра в сети Интернет: www.parallel.ru // Интернет и современное общество. Тезисы второй Всероссийской научно-методической конференции. СПб.: Изд-во С.-Петербур. ун-та, 1999. 55.
2. *Андреев А.Н., Антонов А.С., Воеводин Вл.В., Жуматий С.А.* Профессиональные научные центры в сети Интернет: `http://parallel.ru` // Технологии информационного общества — Интернет и современное общество. Материалы Всероссийской объединенной конференции. СПб.: Изд-во С.-Петерб. ун-та, 2001. 11–13.
3. *Антонов А.С., Крысанов Б.Ю.* Web-сервис для определения базовых характеристик параллельных программ // Технологии информационного общества — Интернет и современное общество. Труды V Всероссийской объединенной конференции. СПб.: Изд-во С.-Петерб. ун-та, 2002. 69–71.
4. *Балашов М.Е., Горячев В.Д., Лукашенко А.В., Рыков Д.С., Смирнов Е.М.* ИВС для решения задач вычислительной гидродинамики с кластерной поддержкой // Научный сервис в сети Интернет. Труды Всероссийской научной конференции. М.: Изд-во МГУ, 2002. 216–218.
5. *Воеводин Вл.В., Филамофитский М.П.* Суперкомпьютер на выходные // Открытые системы. 2003. № 5. 43–48.

Поступила в редакцию
20.09.2003