ОПЫТ ПОСТРОЕНИЯ И ИСПОЛЬЗОВАНИЯ В ГРИД-ТЕХНОЛОГИЯХ КЛАСТЕРА ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ СПИИРАН

М.Ю.Петров,
Санкт-Петербургский институт информатики и автоматизации РАН
199178, Санкт-Петербург, 14-я линия В.О., д.39
miha@edu.nw.ru, miha@aspid.nw.ru

УДК 681.3+519.6
М. Ю. Петров. **Опыт построения и использования в Грид-технологиях кластера высокопроизводительных параллельных вычислений СПИИРАН** // Труды СПИИРАН. Вып. 1, т. 3. — СПб: СПИИРАН, 2003.
Аннотация. *Рассматривается опыт применения Грид технологий в создании кластеров высокопроизводительных параллельных вычислений. Приведено краткое описание многоуровневой модели Грид архитектуры. Анализируется опыт создания параллельных программ, который показал потребность в дальнейшем развитии технологии параллельных вычислений. Представлены структуры параллельных программ и схемы взаимодействия процессов. Показана необходимость создания эффективной системы управления распределением ресурсов. Намечены дальнейшие пути развития кластера СПИИРАН в направлении интеграции с кластерами других академических институтов* - Библ. 6 назв.

UDC 681.3+519.6
M. Y. Petrov. **Some experience of construction and use the cluster for high performance parallel computations in Grid-technologies** // SPIIRAS Proceedings. Issue 1, v. 3. — SPb: SPIIRAS, 2003.
Abstract. *The paper presents the recent practice of grid technologies applications in constructing of high performance cluster. Brief description of layered grid architecture is presented. The analysis being implemented of the experience of parallel program (software) development reveal the need in the further progress of parallel computations. Parallel programs structures and process interaction schemes are also presented in the paper. The need for creation of efficient system of resource distribution control is demonstrated. The perspectives and trends of SPIIRAS cluster construction are outlined towards the integration with clusters of the other academic institutions.* - Bibl. 5 items.

1. Грид-технологии — общие представления

Развитие систем высокопроизводительных вычислений ведется по двум основным направлениям: повышение быстродействия и распараллеливание различных этапов вычислительного процесса, где, в свою очередь, можно выделить внутренний параллелизм (конвейеризация, расслоение памяти и др.), и создание параллельных вычислительных систем. Тенденции развития вычислительных систем таковы: быстродействие компьютеров удваивается каждые 18 месяцев, пропускная способность сетей удваивается каждые 9 месяцев, и разница между ними составляет порядок величины в течение 5 лет.

Однако имеющиеся на сегодняшний день ресурсы эксплуатируются не эффективно и сервисы пользователя ограничены. Обусловлено это тем, что современные Internet-технологии направлены на коммуникации и обмен информацией между компьютерами, но не обеспечивают интегрированного подхода к согласованному использованию ресурсов для выполнения вычислений. Технологии корпоративных распределенных вычислений, такие, как CORBA и Enterprise Java, позволяют разделять ресурсы, но только в пределах одной организации. Для решения проблем интеграции вычислительных ресурсов в середине 90-х создана концепция и технология Грид [1].

Труды СПИИРАН. 2003. Вып. 1. Т. 3. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2003. Issue 1. V. 3. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Технология Грид-вычислений подразумевает взаимодействие множества ресурсов, гетерогенных по своей природе и географически удаленных. Количество объединяемых ресурсов может быть от нескольких элементов до нескольких тысяч и более. При этом возникает потенциальная возможность снижения производительности по мере наращивания ресурсов. Следовательно, приложения, которые требуют для своего решения объединения большого числа географически удаленных ресурсов, должны разрабатываться таким образом, чтобы быть минимально чувствительными к времени задержки. При объединении большого количества ресурсов отказы элементов являются не исключением, а правилом. Поэтому управление ресурсами или приложениями должно осуществляться динамически, чтобы извлечь максимум производительности из доступных в данное время ресурсов и сервисов [2]. Таким образом, основными свойствами Грид-технологии являются:

- распределенность по природе;
- возможность динамической конфигурации;
- неоднородность (гетерогенность) структуры;
- защищенность программ и данных;
- возможность объединения ресурсов различных организаций.

В течение последних лет на основе многоуровневой модели Грид-архитектуры (по аналогии с моделью ISO/OSI) разработаны протоколы, сервисы и инструменты, позволяющие создавать среду разделения ресурсов, обладающую необходимыми свойствами [3]. Структура модели и соотношение с многоуровневой архитектурой Internet-протоколов приведены на рисунке 1.

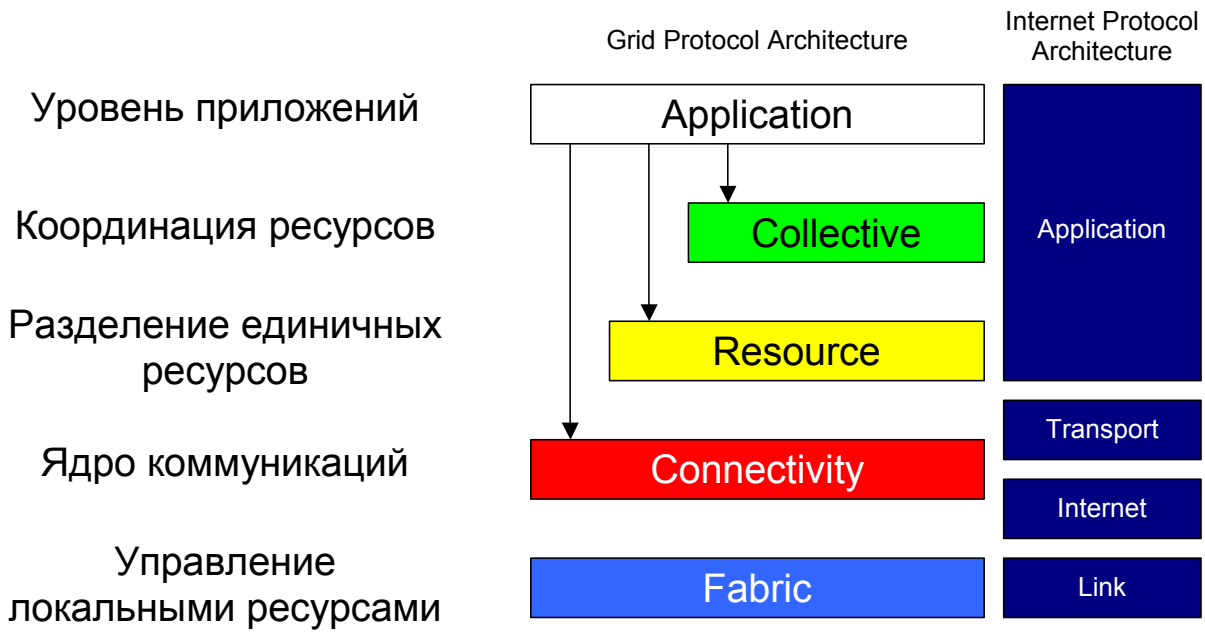

Рис. 1. Многоуровневая модель Грид-архитектуры

Ядром многоуровневой модели являются протоколы Resource и Connectivity, на которые возложены функции обеспечения разделения индивидуальных ресурсов. Уровень Collective отвечает за координацию использования

Труды СПИИРАН. 2003. Вып. 1. Т. 3. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2003. Issue 1. V. 3. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

имеющихся ресурсов и безопасность, доступ к ним осуществляется с помощью протоколов Fabric.

В качестве базовой конструкции узла Грид сети могут выступать векторно-параллельные машины или отдельные компьютеры, но чаще всего применяются кластерные системы, которые, по сравнению с векторными машинами, обладают рядом привлекательных качеств, и, в частности, относительно низкой стоимостью при сопоставимой производительности. Таким образом, высокопроизводительные вычислительные ресурсы становятся доступными и технологично развиваемыми. Удельная стоимость в расчете на единицу производительности (Gflop/S) для таких систем продолжает падать, при этом программное, математическое и методическое обеспечение для них чаще всего свободно распространяемо.

С целью координации реальных Грид-проектов в науке и промышленности предложен своего рода метапроект “Глобус” (http://www.globus.org/), объединяющий множество частных проектов. В рамках их реализации разрабатываются и внедряются стандартные Грид-протоколы для обеспечения совместного функционирования различных вычислительных комплексов и развития инфраструктуры параллельных вычислений, представляющие интерес как потенциальные стандарты.

2. Программное обеспечение для параллельных вычислений

Широкое распространение компьютеров с распределенной памятью определило и появление соответствующих систем программирования. Обычно, в таких системах нет единого адресного пространства, и для обмена данными между параллельными процессами используется явная передача сообщений через коммуникационную среду. Отдельные процессы описываются с помощью традиционных языков программирования, а для организации их взаимодействия вводятся дополнительные функции. Поэтому системы программирования, основанные на явной передаче сообщений, существуют в виде интерфейсов и библиотек.

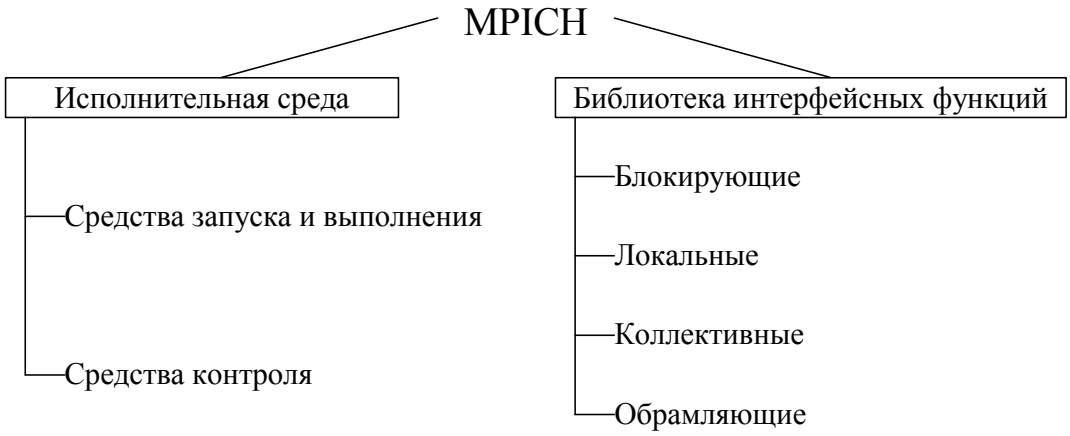

Рис. 2. Структура пакета MPICH

Наиболее распространенной системой программирования на основе передачи сообщений является MPI (Message Passing Interface) [4], представляющий

Труды СПИИРАН. 2003. Вып. 1. Т. 3. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2003. Issue 1. V. 3. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

собой следующий над SMP уровень распараллеливания. Одной из реализаций стандарта MPI является пакет MPICH, структурный состав которого представлен на рисунке 2.

Исполнительная среда взаимодействует с процессами прикладной задачи и ядром операционной системы и занимается распределением процессоров процессам на этапе запуска, а также сбором результатов выполнения процессов (рис. 3).

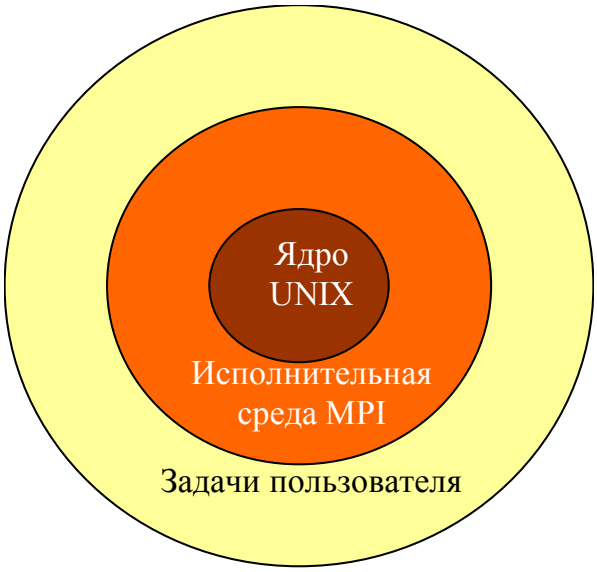

Рис. 3. Уровни взаимодействия MPI в вычислительной среде

Стандарт поддерживает создание параллельных программ путем объединения процессов с различными исходными текстами, однако чаще для всех параллельных процессов используется один и тот же исходный текст.

В создании параллельных программ существуют определенные трудности. Первая из них заключается в том, что не всем разработчикам алгоритмов и программистам ясны возможные сферы применения и преимущества технологии параллельных вычислений. Помимо традиционных “матричных” задач, такая технология, с достаточной степенью эффективности, может быть применена к задачам, распадающимся на несколько параллельных ветвей (рис. 4), “поисковым”, “сортировочным” (рис. 5). В этом случае имеющиеся “непараллельные” программы фактически нужно переписывать заново для встраивания задачи в параллельную структуру кластера.

Вторая трудность заключается в отсутствии доступных средств автоматизации распараллеливания программ. Частичному решению этой проблемы способствует объектно-ориентированный подход в сочетании с CASE-средствами и метаязыками для моделирования разрабатываемых программных систем. При этом при использовании метаязыков, таких как UML, динамические диаграммы языка (например, диаграмма последовательности и др.) позволяют выделить параллельные, независимые на определенном этапе процессы уже на стадии проектирования.

Труды СПИИРАН. 2003. Вып. 1. Т. 3. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2003. Issue 1. V. 3. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

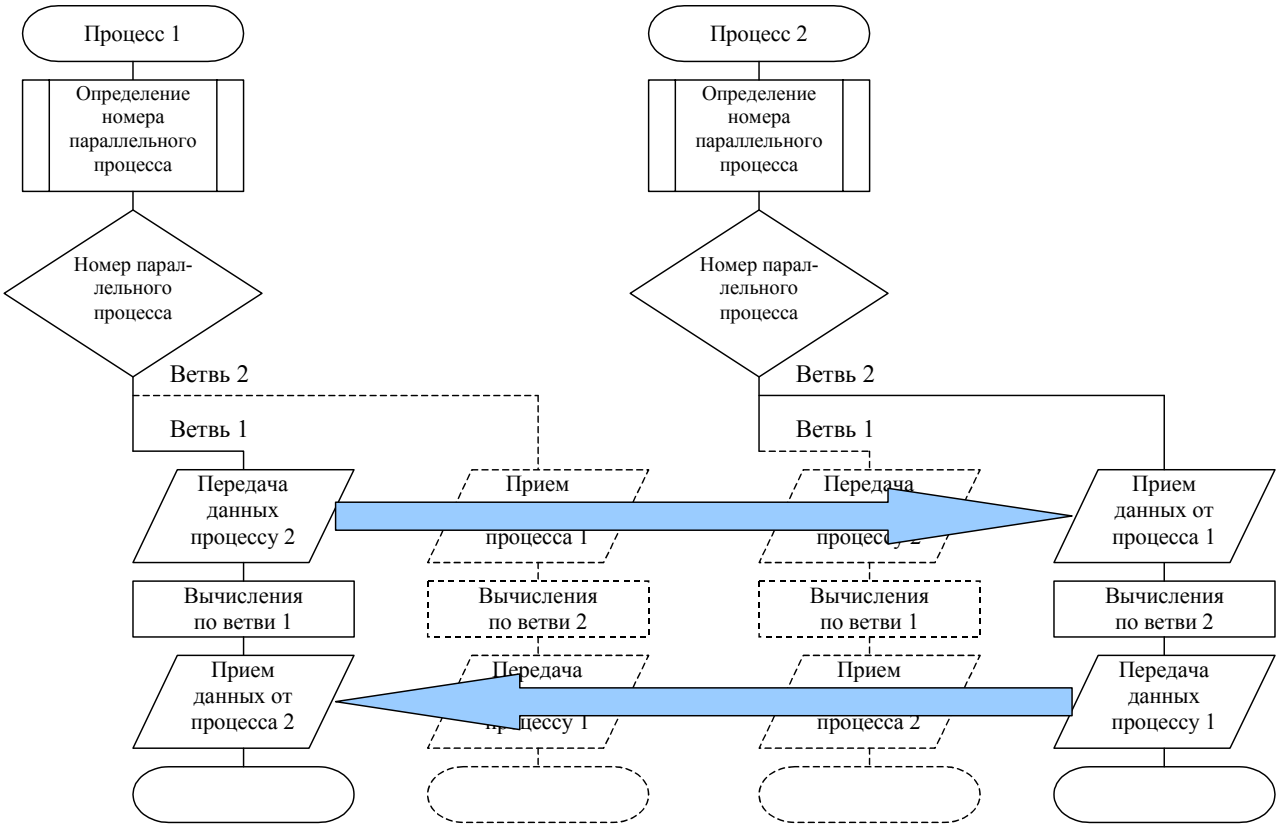

Рис. 4. Структура параллельной программы с фиксированным числом процессов и схема взаимодействия процессов

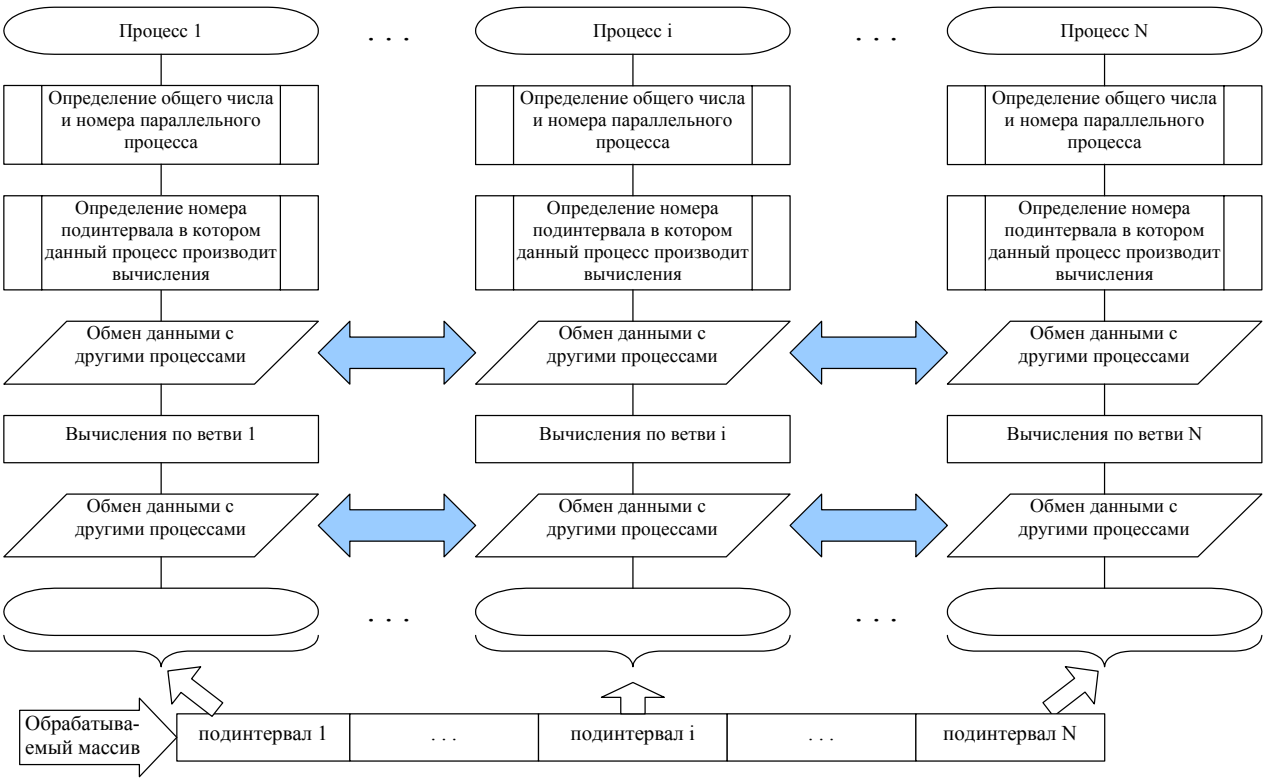

Рис. 5. Структура параллельной программы с переменным числом процессов и схема взаимодействия процессов

Труды СПИИРАН. 2003. Вып. 1. Т. 3. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2003. Issue 1. V. 3. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

Третья трудность состоит в том, что сложность программного обеспечения персональных компьютеров возрастает год от года. Соответственно, для поддержания работоспособности и эксплуатации требуются высококвалифицированные и высокооплачиваемые специалисты. Кроме того, объем и сложность библиотек, пакетов прикладных программ и инструментария для разработки прикладных программ также велики и динамично развиваются. Поэтому проведение эпизодических, высокопроизводительных вычислений было бы экономичнее выполнять на центрах коллективного пользования, с помощью кластерных установок, где пользователь получит квалифицированную консультацию, проведет расчеты в пакетном режиме, воспользуется соответствующими средствами автоматизации программирования и библиотеками стандартных программ.

3. Кластер СПИИРАН

Компьютерное моделирование объектов в различных областях знаний, в частности в геофизике, в биологии, требует построения разностных аналогов трехмерной структуры, для чего вычислительной мощности персональных компьютеров крайне недостаточно [5]. Использование суперЭВМ кластерного типа дает возможность проведения таких расчетов, но для широкого круга пользователей не хватает средств доступа с достаточно скоростными каналами связи, опыта работы на многопроцессорных системах, библиотек программ, инструментария для распараллеливания программ, необходимых пособий и методики обучения.

Имеющийся опыт, накопленный в СПИИРАН, показал нецелесообразность дистанционной отладки программ, так как это существенно замедляет процесс их разработки. Задержки вызваны не низкой скоростью передачи данных, а необходимостью ожидания в очереди. Все это привело к тому, что в 2002 году в СПИИРАН, на базе центра коллективного пользования, был создан кластер высокопроизводительных параллельных вычислений. В состав кластера входят: узлы на базе двухпроцессорной аппаратной платформы Intel PIII и PIV; узлы на базе платформы Sparc Ultra-10; система коммуникаций между узлами; управляющая ЭВМ.

При выборе программного обеспечения создаваемого кластера в основном ориентация была сделана на использование свободно распространяемого программного обеспечения (так называемого “freeware”). Исходя из этой предпосылки, для узлов кластера выбрана сетевая операционная система FreeBSD, в состав которой входят компиляторы языков программирования C и Fortran (77/90). Дополнительно на каждом из узлов кластера установлен пакет MPICH. Производительность кластера в данной конфигурации - до 2,5 Gflop/S.

Для подключения пользователей к вычислительным ресурсам создаваемого кластера, а также объединения с вычислительными ресурсами других организаций, использована существующая высокоскоростная волоконно-оптическая магистраль сети РОКСОН, со скоростью передачи данных 100 Мбит/с.

Пользователям предоставляются следующие услуги: удаленный телекоммуникационный доступ к вычислительным ресурсам кластера и интегрированным ресурсам; обучение и помощь в разработке прикладных программ, для чего создан учебный сервер — http://parallel.edu.nw.ru/; пользование библиотеками научных программ (параллельные процедуры линейной алгебры, сеточных

Труды СПИИРАН. 2003. Вып. 1. Т. 3. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2003. Issue 1. V. 3. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

методов, методов Монте-Карло, генетических алгоритмов, рендеринга изображений, квантовой и молекулярной химии и др).

На кластере СПИИРАН с помощью задачи вычисления ключа дешифровки строки методом подбора произведены измерения зависимости времени решения задачи от числа процессоров, выделенных ей (рис. 6).

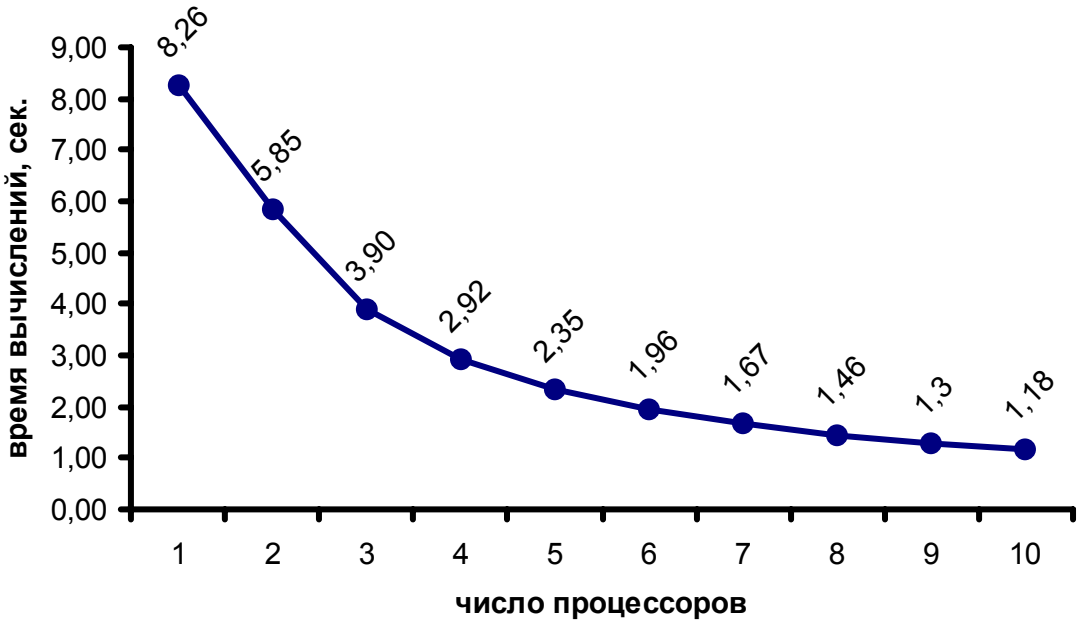

Рис. 6. Зависимость производительности от числа процессоров на кластере СПИИРАН

С целью объединения ресурсов и их более эффективного использования созданный кластер, по мере развития, предполагается интегрировать с существующими в системе РАН и университетскими кластерными центрами (в частности, ИПМ, ФТИ, МСЦ, СПбГУ).

4. Заключение

Подавляющее большинство кластеров в академических и учебных учреждениях имеют упрощенные системы управления распределением ресурсов, эксплуатирующие их по принципу “один процессор для одной задачи”, без учета реальной загрузки, как процессора, так и остальных ресурсов, таких как оперативная память и каналы межмодульного обмена. Эксплуатация показала неэффективность такого подхода, так как использование процессора при обслуживании одной задачи составляет 30-40%; остальное время процессор простаивает. Другие пользователи в это время ожидают своей очереди. В случае запуска двух или более задач загрузка отдельно взятых процессоров может увеличиться до 80%, но суммарная загрузка всех процессоров, в этом случае, весьма далека от желаемой (40% и ниже). Обусловлено это тем, что имеющиеся стандартные средства не позволяют равномерное распределение ресурсов не только динамически, в процессе счета, но и на этапе запуска задачи. Следствием чего может быть ситуация, когда часть процессоров сильно перегружена, при недостаточной загрузке остальных. Очевидно, что необходимо создание системы управления, которая будет распределять процессоры задачам более гибко, с учетом реального использования ресурсов в каждый конкретный момент, а также данных по динамике выполнения конкретных задач, полученных в

Труды СПИИРАН. 2003. Вып. 1. Т. 3. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2003. Issue 1. V. 3. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

результате мониторинга предыдущих запусков, что, безусловно, ускорит прохождение задач. Такая система управления позволит поднять суммарную загрузку процессоров до 70-80%, а время ожидания в очереди снизится [6].

Потребность в создании системы управления кластером, способной распределять и динамически перераспределять ресурсы кластерных комплексов, как централизованных, так и распределенных сетевых, в том числе временных, для реализации конкретных проектов, является актуальной не только в настоящее время, но и в перспективе. Решение проблемы управления распределением ресурсов необходимо искать с учетом общих тенденций, и, в частности, с использованием стандартных протоколов и интерфейсов проекта “Глобус”.

Литература

[1] *Шевель А.* Технология GRID //«Открытые системы». №2 (58). Открытые системы. — М.: 2001. — С. 36-39.

[2] *Воеводин В. В., Воеводин Вл.* Параллельные вычисления — СПб.: БХВ-Петербург, 2002. — 608 с.

[3] *Бараш Л.* Grid Computing — новая парадигма Internet-вычислений. //«Компьютерное обозрение». №32, 22–29 августа 2001, http://www.itc.ua/7249

[4] *Chisholm G.H.* Toward a Verifiable Approach to the Design of Concurrent Computations // Argonne National Laboratory, 1993. http://www-unix.mcs.anl.gov/mpi/

[5] *Abramov A.G., Ivanov N.G., Smirnov E.M.* Numerical study of high-Ra Rayleigh-Benard mercury and water convection in confined enclosures using a hybrid RANS/LES technique // Department of Aerodynamics, Saint-Petersburg State Polytechnic University, 2002 (submitted for publication).

[6] *Петров М.Ю.* Применение кластеров высокопроизводительных параллельных вычислений в научных исследованиях // Материалы конференции «Региональная информатика — 2002» (СПб, 26-28 ноября 2002 г.) — СПб.: 2002. — ч. 2, с. 47.