ТЕХНОЛОГИЯ ОРГАНИЗАЦИИ ГЕТЕРОГЕННЫХ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СРЕД

И.В. Бычков, А.С. Корсуков, Г.А. Опарин, А.Г. Феоктистов

Введение

По мере продвижения к информационному сообществу наблюдается значительное увеличение объемов получаемой, обрабатываемой и распространяемой информации. Значительную долю в этих объемах составляет информация вычислительного характера, то есть информация, получаемая с использованием различного рода вычислительных установок: персональных компьютеров, вычислительных кластеров, вычислительных сетей и др. При этом обычные средства вычислительной техники уже не справляются со сложными ресурсоемкими информационными задачами.

Постоянный рост требований, исходящих от фундаментальных и прикладных научных приложений к компьютерным ресурсам (быстродействию, оперативной памяти, дисковой памяти и т.п.), привел к необходимости объединения компьютерных ресурсов сети Интернет и их совместного эффективного использовании. В последнее время для решения ресурсоемких фундаментальных и прикладных задач создаются различные распределенные и параллельные вычислительные системы и среды (см., например, [1-8]). В частности, активно применяется технология создания специальной вычислительной сети, получившей название Grid [9]. На сегодняшний день Grid является наиболее актуальной формой интеграции информационно-вычислительных ресурсов сети Интернет при решении больших ресурсоемких задач [10]. Применяемое в настоящее время программное обеспечение для организации Grid (Globus Toolkit [11], The Virtual Data Toolkit [12], gLite [13] и др.) не обеспечивает простого и удобного доступа пользователей к ресурсам Grid, а также не всегда позволяет реализовать выполнение ряда нетривиальных задач[1].

В работе предложен способ создания гетерогенных распределенных вычислительных сред[2] (РВС), предоставляющих широкий выбор средств для решения ресурсоемких научно-исследовательских задач различных типов и обеспечивающих возможность интеграции с другими РВС. Рассмотрен программный комплекс, предназначенный для инструментальной поддержки основных этапов построения и применения РВС.

Разработанное ПО предоставляет пользователям возможность самостоятельно (без участия высококвалифицированных системных программистов) описывать исследуемую предметную область и программно-аппаратную часть РВС, а также формировать задания для решения своих задач и проводить вычислительные эксперименты в РВС. Тем самым сокращаются сроки и повышается эффективность проведения сложных научных и прикладных задач.

Модель РВС

Предложенная авторами модель РВС [14] обеспечивает накопление знаний о вычислительных ресурсах, а также администрирование, планирование и применение этих ресурсов при решении пользовательских задач.

В рассматриваемой модели выделим следующие основные уровни:

- пользовательский уровень, включающий в себя: способы и средства доступа к вычислительным узлам РВС; множество пользователей РВС, их классификацию по категориям и правам доступа к РВС; множество заданий, запускаемых пользователями в РВС;
- уровень ПО, отражающий свойства и характеристики программных приложений, запуск которых требуется при выполнении заданий;
- аппаратный уровень, определяющий характеристики и комплектующие вычислительных узлов и коммутирующих устройств РВС;
- уровень планирования вычислений и загрузки ресурсов, обеспечивающий выбор необходимых ресурсов РВС для запуска заданий;
- вычислительный уровень, отражающий процесс выполнения заданий на конкретных вычислительных узлах РВС.

Разные уровни модели РВС содержат свои наборы объектов. Для каждого объекта определены его атрибуты. Особенностью этой структуры является то, что она позволяет на разных ее уровнях определять и совместно использовать различные модели распределенных вычислений, такие как модели программирования приложений, модели планирования вычислительных процессов, модели планирования загрузки ресурсов.

Схема взаимодействия пользователя с узлами РВС

[1] К такого рода задачам можно отнести, например, получение вычислительных услуг нетиражируемых программных комплексов (размещенных в узлах Grid) или выполнение набора взаимозависимых и частично упорядоченных заданий.

[2] Под РВС следует понимать как отдельные вычислительные кластеры, так и Grid.

Рассмотрим схему взаимодействия пользователя с узлами РВС, представленными вычислительными кластерами. В рамках приведенной схемы в качестве интеллектуальной управляющей надстройки к СУПЗ, установленных в узлах РВС, используется программная система Web-Interface Manager (WIM).

Запуск пользовательского задания и его выполнение на узлах вычислительного кластера происходит поэтапно. Прежде чем приступить к работе с кластером, пользователь должен пройти процедуру регистрации, если же у него имеется учетная запись, то необходимо пройти процедуру авторизации. Следующим этапом будет отображение веб-формы, с помощью которой пользователь сможет выбрать тип исполняемой задачи и необходимый режим запуска. Каждому режиму запуска задания соответствует определенная веб-форма, которая содержит набор необходимых полей, на основе которых система WIM составляет паспорт задания. Такие поля обычно содержат информацию о запускаемой пользовательской программе, ее исходные данные и другую специфическую информацию, характерную для каждого режима запуска.

Таким образом, после заполнения пользователем полей в веб-форме, информация о задании передается по протоколу HTTP на главный узел кластера, где установлен веб-сервер Apache. Система WIM получает поступившие данные от веб-сервера, обрабатывает их и составляет паспорт задания в формате СУПЗ. Подготовленный паспорт заданий, программа на выполнение и ее исходные данные передаются запускающей службе СУПЗ. Главная служба СУПЗ, основываясь на поставляемой службой сбора информации о загруженности кластера, производит выбор подходящего вычислительного узла (узлов), на котором будет выполняться задание пользователя. Выполнение задания обеспечивает выполняющая служба СУПЗ, установленная на каждом вычислительном узле кластера. В процессе выполнения задания пользователь имеет возможность совершать следующие операции: управлять заданием, просматривать ход его выполнения и просматривать отчет об ошибках. После того как задание выполнено, пользователь может скачивать файлы с результатами счета и просматривать историю выполнения задания. Операции, выполняемые пользователем в веб-интерфейсе, обрабатываются системой WIM и конвертируются в команды, понятные СУПЗ. Ответы на посланные команды перехватываются и транслируются системой WIM пользователю.

Схема взаимодействия пользователя с узлами РВС представлена на рис. 1.

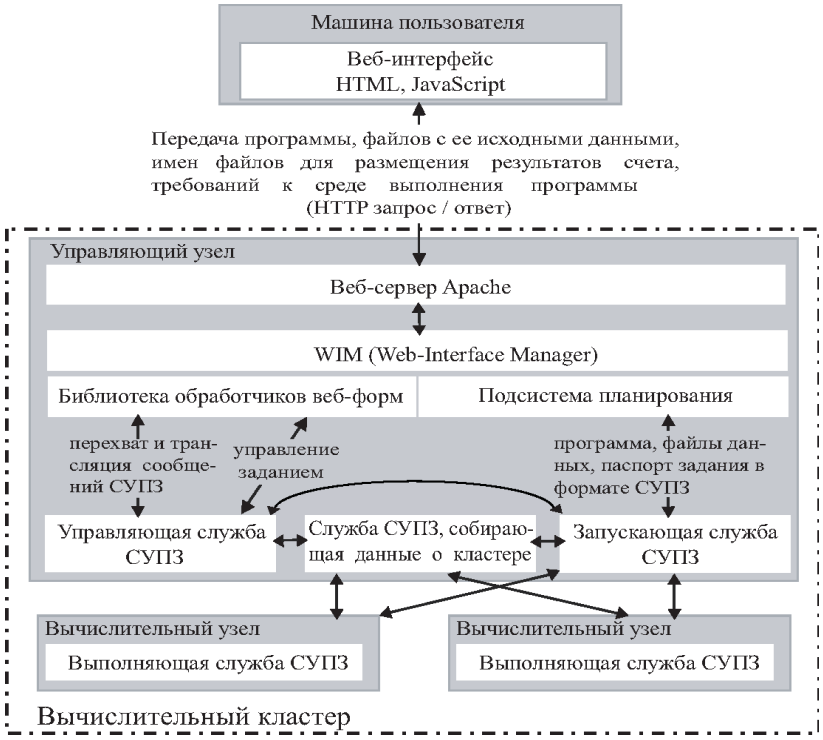

Рис. 1. Схема взаимодействия пользователя с узлами РВС

Архитектура инструментального комплекса

Инструментальный комплекс (ИК) DIStributed Computing ENvironment Toolkit (DISCENT) предназначен для организации и применения РВС. ИК DISCENT состоит из трех основных компонентов (рис. 2): конструктора, базы данных и системы WIM.

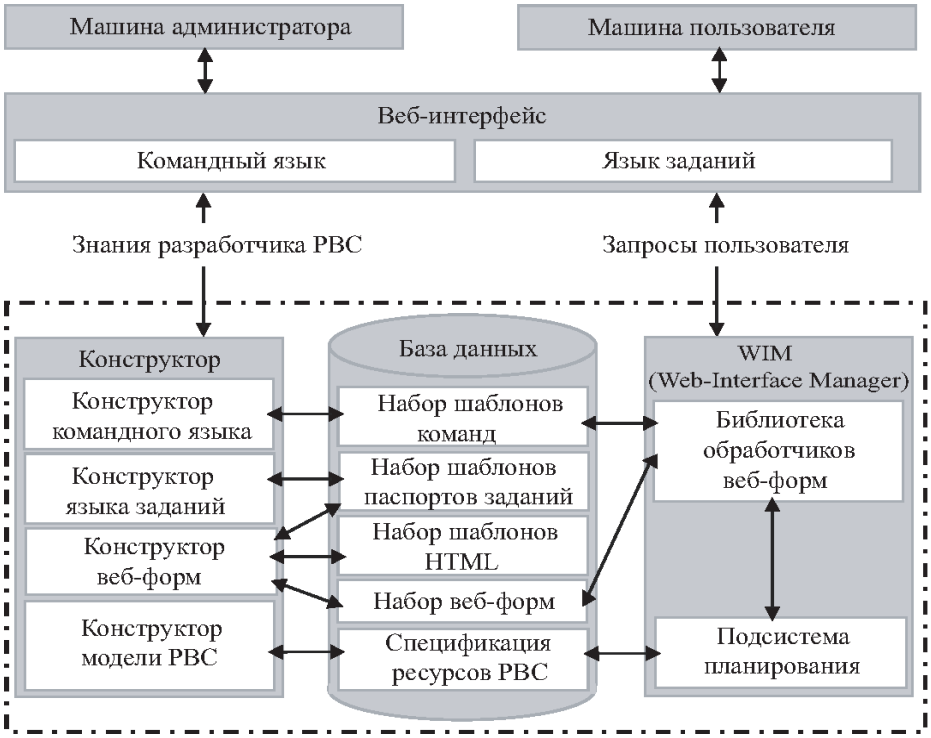

Рис. 2. Архитектура ИК DISCENT

Конструктор включает четыре подсистемы.

- *Конструктор командного языка* служит для создания, редактирования или удаления шаблонов команд для различных СУПЗ.
- *Конструктор языка заданий* используется для построения и модификации шаблонов паспортов заданий для различных СУПЗ.
- *Конструктор веб-форм* предназначен для создания шаблонов веб-форм, используемых для заполнения паспортов заданий. Создание веб-формы паспорта задания выполняется путем размещения на форме и задания свойств графических элементов HTML, соответствующих параметрам формируемого паспорта задания.
- *Конструктор модели РВС* применяется для описания и модификации данных о вычислительных ресурсах РВС. Такая информация необходима подсистеме планирования для эффективного распределения различных типов задач по вычислительным узлам РВС.

Язык заданий определяет взаимодействие пользовательских задач с системой WIM и предназначен для формирования паспорта задания, включающего в себя уникальное имя задания, тип решаемой задачи, исполняемые программы, исходные данные, минимальные и/или желаемые требования к вычислительным ресурсам РВС и др.

Командный язык представляет собой совокупность команд, предназначенных для управления пользовательскими заданиями в узлах РВС. С помощью командного языка система WIM может быть легко и гибко настроена для взаимодействия с используемой СУПЗ за счет применения унифицированных шаблонов описания команд СУПЗ.

Основным назначением базы данных является накопление информации о вычислительных ресурсах РВС, средствах создания и обработки паспортов заданий. Эта информация используется планировщиками системы WIM при формировании потоков заданий. Содержимое базы данных включает наборы шаблонов для командного языка и языка заданий, наборы графических элементов HTML и веб-форм, спецификацию ресурсов РВС.

Система WIM обеспечивает выполнение заданий пользователя путем распределения указанных в заданиях приложений на узлы РВС, подходящие для их запуска. Основными составляющими системы являются библиотека обработчиков веб-форм и подсистема планирования.

Библиотека обработчиков веб-форм предназначена для выполнения следующих функций: вывода пользователям требуемых веб-форм; обработки поступивших из этих форм данных; формирования паспортов заданий; предоставления пользователю информации о состоянии заданий и результатах счета; добавления, удаления и приостановления заданий пользователей. Данная библиотека состоит из набора скриптов языка PHP (обработчиков веб-форм), каждый из которых реализует алгоритмы для работы с одной или несколькими веб-формами на стороне веб-сервера. Имя обработчика веб-формы содержится на ней в специальном скрытом поле. Обработчики веб-форм имеют доступ к шаблонам команд, хранящимся в базе данных. В шаблонах команд для

каждой операции управления заданием, размещенной на веб-форме, определяется команда СУПЗ, с помощью которой данная операция будет выполнена в РВС.

Подсистема планирования служит для обработки, постановки в очередь и распределения заданий, соответствующих различным типам задач. Данная подсистема включает три планировщика.

1. *Планировщик стандартных заданий MST* (*Manager of Standard Tasks*) используется для распределения стандартных заданий (скомпилированных программ пользователей), для выполнения которых используется штатный набор функций СУПЗ.
2. *Планировщик распределенных приложений MDA* (*Manager of Distributed Applications*) предназначен для отправки заданий на вычислительные узлы РВС, на которых установлены приложения, необходимые для выполнения заданий. Планировщик MDA взаимодействует с планировщиком MST с целью резервирования за заданием выбранных ресурсов перед передачей этого задания обработчику веб-форм для запуска задания.
3. Планировщик взаимосвязанных заданий MIT (Manager of Interrelated Tasks) ориентирован на распределение заданий для решения взаимосвязанных задач (такие задачи требуют выполнения набора взаимозависимых заданий, включенных в процесс решения одной общей задачи) и контроль процесса их выполнения. Планировщик MIT, используя информационно-логические связи между объектами модели РВС, выполняет частичное упорядочение подзадач общего задания, находит необходимые для выполнения задания вычислительные ресурсы, взаимодействует с планировщиком MST с целью резервирования за заданием выбранных ресурсов, передает задание обработчику веб-форм для его запуска и осуществляет дальнейший контроль выполнения этого задания.

Как правило, резервирование ресурсов увеличивает время ожидания заданий в очереди [15]. Однако без применения такого способа время ожидания возрастает в значительно большей степени для заданий, требующих заранее установленных приложений в узлах РВС для их выполнения, и заданий для решения взаимосвязанных задач.

При распределении заданий по узлам РВС планировщики системы WIM используют спецификацию ресурсов РВС из базы данных.

Технология построения гетерогенных РВС

Технология организации РВС с помощью ИК DISCENT включает следующие основные этапы:

– *Определение вычислительных узлов РВС,* которые будут задействованы в качестве ее ресурсов для выполнения заданий пользователей. Помимо этого необходимо выделение рабочей станции (или машины) с выходом в Интернет, которая будет выполнять роль веб-сервера.

– *Установка и настройка ПО.* На данном подэтапе осуществляется инсталляция системного ПО, необходимого для функционирования ИК DISCENT. В частности, необходима установка и настройка следующих пакетов: веб-сервер Apache версии 2.0.43 (или выше), интерпретатор языка программирования для веб PHP версии 5 (или выше). Затем выполняется установка и конфигурация ИК DISCENT, включающая копирование исходных файлов данного ИК на веб-сервер РВС и задание значений переменных окружения (например, путь к СУПЗ).

– *Построение модели РВС* включает формирование множества пользователей РВС, заинтересованных в использовании вычислительных ресурсов РВС; классификацию пользователей по видам или категориям (например, администраторы, операторы, преподаватели, студенты, сторонние пользователи и др.); наделение их соответствующими правами использования ресурсов РВС; составление спецификации вычислительных ресурсов РВС; определение основных типов задач, которые будут решаться в РВС.

– *Параметризация и настройка шаблонов команд для каждой СУПЗ, используемой в РВС.* На данном подэтапе создаются шаблоны команд, для того чтобы система WIM могла взаимодействовать с различными СУПЗ.

– *Параметризация и настройка шаблонов паспортов заданий.* Для каждого типа пользовательского задания, которое будет обрабатываться определенной СУПЗ, необходимо создать шаблон паспорта задания, который должен учитывать специфику запуска выбранного типа заданий и особенностей СУПЗ.

– *Создание веб-форм* производится администратором РВС с помощью конструкторов, входящих в состав ИК DISCENT и позволяющих в графическом интерактивном режиме формировать необходимые поля (текстовые поля, выпадающие меню и т.д.) на веб-формах, которые будут выполнять запуск определенных типов заданий.

– *Отладка и тестирование режимов запуска пользовательских заданий.* На этом подэтапе происходит прогон специально подготовленных тестовых примеров для каждого режима запуска заданий, определяется их работоспособность и отказоустойчивость. В случае появления сбоев в работе системы WIM происходит поиск возникших ошибок, выявление и устранение их причин.

– *Регистрация пользователей* включает создание учетных записей пользователей, которые будут использовать ресурсы РВС.

– *Формирование паспортов и запуск заданий.* На данном подэтапе для осуществления запуска задания пользователи должны выбрать тип запускаемого задания и заполнить все необходимые поля в веб-форме, предложенной системой WIM.

– *Мониторинг узлов РВС, распределение заданий в узлах РВС* выполняют планировщики системы WIM. С помощью командного языка планировщики различных типов заданий могут получать информацию у СУПЗ о состоянии загруженности узлов РВС. Спецификацию вычислительных ресурсов планировщики получают из базы данных.

– *Обмен данными между пользователями и СУПЗ* осуществляют обработчики веб-форм, которые получают данные (например, программу на выполнения, ее исходные данные и др.) с машины пользователя с помощью метода «upload». Данные же с СУПЗ (например, файл с результатами счета, файл отчета СУПЗ) обработчики получают из определенной директории, место расположение которой указывается в паспорте задания для СУПЗ. После того, как задание выполнено, обработчики веб-форм помещают результаты счета в уникальный веб-каталог и высвечивают пользователю гиперссылку, по которой эти данные могут быть сохранены на жесткий диск.

Вычислительный эксперимент

ИК DISCENT был использован для организации экспериментальной Grid ИДСТУ СО РАН и решения в ней ряда практических задач. С целью более полного и детального анализа эффективности функционирования Grid, в том числе и разработанных средств управления заданиями, был проведен вычислительный эксперимент по моделированию процессов формирования и обработки непрерывных потоков заданий в течение длительного периода времени.

Grid представляет собой стохастическую динамическую вычислительную сеть, состояние которой в некоторый момент времени определяется распределением в ней потоков заданий [16]. Для получения оценки эффективности различных способов распределения потоков заданий (E, %) генерируется поток заданий, осуществляется их распределение и выполнение в Grid двумя способами (с помощью GridWay и GridWay + WIM) и сравниваются следующие показатели эффективности функционирования вычислительных ресурсов Grid: число заданий с нулевым временем ожидания (T_0, сек.), среднее время ожидания задания в очереди (T_q, сек.), среднее число заданий в очереди (C_q, ед.), среднее время пребывания задания в Grid (T_g, сек.), коэффициент полезного использования ресурсов Grid (K_g, %), количество заданий в потоке (C_s, ед.), общее время решения заданий потока (T_s, сек.).

Поток заданий в Grid характеризуется следующими свойствами: неоднородностью (задания соответствуют разным типам задач и отличаются друг от друга по своей специфике); отсутствием обратной связи (число заданий, поступивших за один промежуток времени, не зависит от числа заданий, поступивших за другой промежуток времени); неординарностью (возможно поступление двух и более заданий в один и тот же момент времени); стационарностью (число событий, поступивших за определенный промежуток времени, зависит от длины этого промежутка и не зависит от момента его начала).

Для формирования потока заданий и отправки их на выполнение в Grid ИДСТУ СО РАН разработан специальный генератор. В качестве приложений в заданиях используются программы-имитаторы, выполняющие в Grid реальную загрузку вычислительных ресурсов и обмен заданными объемами данных. Генератор заданий был запущен на удаленной машине в Институте вычислительного моделирования СО РАН, которая играла роль сторонней Grid, а также на независимых рабочих станциях в Интернет, представляющих машины пользователей веб-интерфейса к Grid ИДСТУ СО РАН. Результаты вычислительных экспериментов (см. таб. 1) показывают устойчивое преимущество второго способа распределения заданий (GridWay+WIM) по всем показателям эффективности функционирования ресурсов Grid.

Таблица 1. Результаты вычислительных экспериментов

Grid	2 кластера с ОС Linux (168 выч. ядер)			2 кластера с ОС Linux, кластер с ОС Windows (184 выч. ядер)			2 кластера с ОС Linux, кластер с ОС Windows (184 выч. ядер)		
	GridWay	GridWay+ WIM	E (%)	GridWay	GridWay+ WIM	E (%)	GridWay	GridWay+ WIM	E (%)
T_0	49060	52620	7,26%	24340	26680	9,61%	17980	19920	10,79%
T_q	87260	80100	–8,21%	44020	39740	–9,72%	28240	25540	–9,56%
C_q	7360	7260	–1,36%	3760	3620	–3,72%	2380	2280	–4,20%
T_g	123920	118440	–4,42%	62580	58460	–6,58%	39220	35420	–9,69%
K_g	93,34%	94,13%	0,85%	92,41%	95,12%	2,93%	93,74%	94,97%	1,31%
C_s	18688	18688	–	9344	9344	–	157	157	–

T_s	4917080	4716820	–4,07%	2486420	2336520	–6,03%	1585160	1459720	–7,91%

Пример

В качестве примера применения созданной Grid рассмотрим процесс моделирования работы супермаркета с помощью системы GPSS [17].

Постановку задачи моделирования работы супермаркета в общем случае можно представить следующим образом: $S=<A, T, C, B, CR, F, \tau \rightarrow k, P, T_0>$. Здесь исходными данными являются: A – количество мест на парковке супермаркета; $T=t\pm\Delta t$ – время подхода покупателей; C – количество тележек; B – количество ручных корзин; CR – количество кассовых аппаратов; F – вероятностное распределение потока покупателей; τ – моделируемый промежуток времени работы супермаркета. Требуется найти: k – коэффициент загрузки всех касс; $P=\{P_{min}, P_{mid}, P_{cur}\}$, где P_{min}, P_{mid}, P_{cur} – максимальное, среднее и текущее число покупателей в каждой очереди; $T_{обсл}=\{T_{can}, T_{que}\}$, где T_{can}, T_{que} – среднее время обслуживания в каждом канале и среднее время нахождения покупателя в каждой очереди. В качестве единицы измерения модельного времени берется 1 секунда.

Время прогона 1 варианта GPSS-модели процесса работы супермарекета в течение одной смены на одном ПК составляет 1 секунду. Если изменять время прихода покупателей и количество используемых парковочных мест, касс, тележек и ручных корзин, то возникает необходимость в многовариантных расчетах. Общее время прогона (таб. 2) 200 вариантов модели (время решения задачи) на ПК составляет 200 сек., а в Grid – 278 сек. Это объясняется наличием накладных расходов по времени на передачу паспорта задания программы и данных между вычислительными узлами Grid.

Таблица 2. Время решения задачи

	τ (ч.)				
	8	168	720	2160	8760
Время решения задачи на ПК (сек.)	200	200	400	600	1800
Время решения задачи на 28 процессарах Grid (сек.)	278	285	315	331	352

Увеличение моделируемого периода с 1 смены до 1 недели (1 месяца, 1 квартала и т.д.) соответственно приводит к росту времени решения задачи как на ПК, так и в Grid. Причем, при моделировании работы супермаркета в течение месяца время решения задачи в Grid уже меньше по сравнению с временем решения задачи на локальном компьютере.

Таким образом, при многовариантных расчетах длительного периода моделирования общее время решения задачи на ПК резко возрастает (рис. 3) в этом случае целесообразно использовать вычислительные возможности Grid.

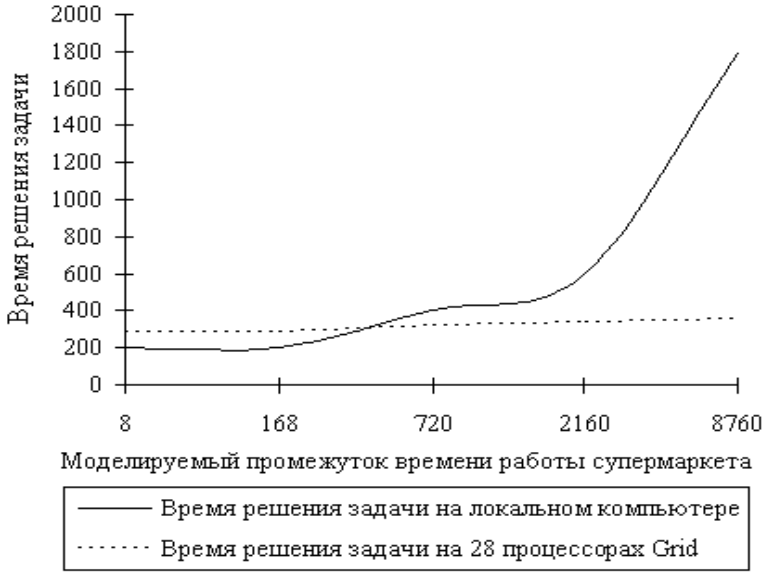

Рис.3. График времени решения задачи

Заключение

Разработан способ организации распределенных вычислений, отличающийся от известных обеспечением таких дополнительных возможностей, как формирование потоков заданий в зависимости от типов решаемых задач и распределенное управление этими потоками в процессе выполнения заданий. В основе этого способа лежат специализированные языковые и инструментальные средства, предназначенные для быстрой и гибкой настройки веб-ориентированного доступа к ресурсам гетерогенной РВС.

ЛИТЕРАТУРА:

1. В.Б. Бетелин, Е.П. Велихов, А.Г. Кушниренко "Массовые суперкомпьютерные технологии – основа конкурентоспособности национальной экономики в XXI веке" // Информационные технологии и вычислительные систем. – 2007. – № 2. –С. 3-10.
2. В.С. Бурцев "Параллелизм вычислительных процессов и развитие архитектуры суперЭВМ" – М.: ИВВС РАН, 1997. – 152 с.
3. В.В. Воеводин, Вл.В. Воеводин "Параллельные вычисления" – СПб.: БХВ-Петербург, 2002. – 608 с.
4. С.В. Емельянов, А.П. Афанасьев, В.В. Волошинов, Я.Р. Гринберг, В.Е. Кривцов, О.В. Сухорослов "Реализация Grid-вычислений в среде IARnet" // Информационные технологии и вычислительные системы. – М.: Институт микропроцессорных вычислительных систем РАН, 2005, №2, С.61-75.
5. В.Н. Коваленко, Д.А. Корягин "Организация ресурсов ГРИД" / – М., 2004. – 25 с. – (Препринт / ИПМ им. Келдыша РАН; № 63).
6. В.В. Корнеев "Параллельные вычислительные системы" – М.: Нолидж, 1999. – 320 с.
7. А.О. Лацис "Как построить и использовать суперкомпьютер" – М.: Бестселлер, 2003. – 274 с.
8. Г.А. Опарин, Феоктистов А.Г. "Инструментальная распределенная вычислительная САТУРН-среда" // Программные продукты и системы. – 2002. – №2. – С. 27-30.
9. I. Foster, C. Kesselman, S. Tuecke "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" // Intern. J. of High Performance Computing Applications. – 2001. – Vol. 15, № 3. – P. 200-222.
10. А.П. Демичев, В.А. Ильин, А.П. Крюков "Введение в грид-технологии". – М., 2007. – 87 с. – (Препринт / НИИЯФ МГУ; №11/832).
11. Globus Toolkit Homepage. – [http://globus.org/toolkit/].
12. The Virtual Data Toolkit. – [http://vdt.cs.wisc.edu/].
13. EGE gLite. – [http://glite.web.cern.ch/glite/].
14. А.Г. Феоктистов, А.С. Корсуков "Разработка Grid-системы с децентрализированным управлением потоками заданий" // Вестник НГУ. Серия: Информационные технологии. – 2008. – Т. 6, вып. 3. – С. 147-154.
15. В.В. Топорков "Модели распределенных вычислений". – М.: ФИЗМАТЛИТ, 2004. – 320 с.
16. Ю.С. Попков "Макросистемы и GRID-технологии: моделирование динамических стохастических сетей" // Проблемы управления. – 2003. – № 8. – С. 10-20.
17. Е.М. Кудрявцев "GPSS World. Основы имитационного моделирования различных систем". – М.: DMK Press, 2003. – 320с.