СИСТЕМА РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ В МАТЕМАТИЧЕСКИ ОДНОРОДНОМ ПОЛЕ ИСЧИСЛЕНИЯ ДРЕВОВИДНЫХ СТРУКТУР

С.Е. Артамонов, Ю.С. Затуливетер, В.А. Козлов, В.С. Подлазов, В.В. Сергеев, А.В. Топорищев, Е.А. Фищенко

Глобальная компьютерная среда (ГКС) уже включает несколько миллиардов стационарных и мобильных компьютерных устройств связанных сетями и продолжает быстро расти. Посредством протоколов TCP/IP эти устройства (ПК, смартфоны, рабочие станции, суперкомпьютеры и др.) обмениваются данными, образуя сильносвязное информационное пространство. Высокий технический уровень компьютерных устройств связанных сетями предопределяет колоссальный совокупный функциональный и вычислительный потенциал ГКС. Так, пространство WWW связывает уже более 1,5 млрд. ПК. Только эта среда уже обладает суммарной производительностью более 1-2 Эфлопс (10^{18} флопс), общей оперативной памятью более 1-2Эбайт (10^{18} байт) и сотнями экзабайт долговременной памяти. Параллельно с этой средой развиваются сети мобильной связи (более 4,5 млрд. абонентов). В стандартах 3G и 4G они быстро интегрируются в WWW, увеличивая не только вычислительный потенциал ГКС, но и привнося новые функциональные возможности, связанные с перемещениями вычислительных узлов (универсальные платежные терминалы, сенсоры, мониторинг, навигация и многое др.)

Ключевая проблема дальнейшего развития ГКС в том, что с увеличением размеров ее быстро растущий вычислительный потенциал используется для переработки быстро растущих потоков и объемов глобально распределенной информации лишь в ничтожной своей части. Действительно, сетевые вычислительные ресурсы ГКС изначально дезинтегрированы и потому впрямую не доступны для бесшовной реализации универсально программируемых распределенных вычислений. Фундаментальная причина отсутствия системной целостности ГКС в том, что свойство универсальной программируемости, которым в рамках модели Дж. фон Неймана обладают современные компьютеры, изначально замкнуто во внутрикомпьютерных ресурсах.

В рамках существующих системных правил для распространения свойства программируемости на сетевые ресурсы дополнительно требуется создание в них весьма сложных и трудоемких промежуточных программных слоев (middleware), которые интегрируют выделенные ресурсы локальных и глобальных сетей в системы распределенных вычислений того или иного назначения.

Такие решения строятся и развиваются в рамках разнообразных вариантов Grid-технологий [1-3]. Разрабатываемые с их применением системы распределенной обработки данных в большинстве случаев имеют корпоративное назначение и представляют собой замкнутые вычислительной среды, ориентированные на предоставление ограниченных наборов сервисов. Они строятся по разным моделям и сетевым архитектурам, закрепляемым в дополнительных программных надстройках над стандартными операционными системами. В результате получаются разнородные и громоздкие многослойные программные решения, с жесткими, как правило, узкопрофильными структурами, с чрезмерным числом степеней свободы выведенных на человека. Трудоемкость, а значит и себестоимость, а также сроки их разработки, весьма значительны. Эксплуатация требует сложного администрирования на системных и сетевых уровнях.

Такие решения, как правило, трудно адаптируются к структурным изменениям внешнего контекста глобального информационного пространства, тенденции развития которого, по понятным причинам, не подчиняется внутрикорпоративным интересам. В результате длительных сроков разработки таких систем информационная среда, в которой они должны функционировать, может существенно изменить свои свойства, что приводит к обесцениванию результатов усилий многих разработчиков, поскольку решения часто устаревают еще на этапах осуществления проектов.

Глобализация распределенных вычислений в сетевых ресурсах идет двумя встречными направлениями. Первое — «снизу-вверх» (от практики), второе — «сверху-вниз» (от теории).

Первое, в настоящее время доминирующее, направление представлено Grid-технологиями [1-3]. Многочисленные и разнородные версии Grid развиваются в рамках классической модели универсальных вычислений Дж. фон Неймана, которая лежит в основе массового производства компьютеров и технологий индустриального программирования. Это путь постепенного, ограниченного рамками классической модели вычислений, наращивания средств индустриального программирования в области сетевых вычислений. Интеграция сетевых ресурсов осуществляется в ходе разработки промежуточных программных слоев посредством силового преодоления комбинаторной сложности интеграции крайне разнородных данных, программ, процессов и систем.

За полтора десятилетия развития Grid-технологий проявились их принципиальные ограничения и недостатки:

- отсутствие единой математической модели, которая могла бы стать основой формирования в сетевых ресурсах глобального универсально программируемого алгоритмического пространства распределенных вычислений;

- высокие уровни неоднородности и сложности системного ПО, формирующего сетевую инфраструктуру распределенных вычислений, которые быстро нарастают с увеличением количества вовлекаемых компьютеров;
- отсутствие единого адресного пространства оперативной памяти сетевых ресурсов и средств «бесшовного» программирования в нем структурно сложных распределенных вычислений.

Данное направление глобализации распределенных вычислений опирается на индустриальные Grid-стандарты и реализующие их промежуточное ПО [3]. Однако, в силу комбинаторной сложности задач интеграции глобально распределенных данных, программ, процессов и систем (по причине крайней их разнородности), никакие согласительные процедуры не могут охватывать совокупные ресурсы быстро растущих и развивающихся глобальных сетей. Поэтому, как нам представляется, временны*е* горизонты направления «снизу-вверх», развивающегося step-by-step в условиях исчерпания системообразующего потенциала классической модели вычислений, ограничены и оно не может составлять стратегической альтернативы встречному направлению «сверху-вниз».

Цель встречного направления глобализации пространства распределенных вычислений «сверху-вни» – устранение указанных ограничений посредством общей модели универсально программируемых распределенных вычислений с единой, математически однородной формой представления структурно-сложной компьютерной информации (данных и программ).

Новая модель строится посредством минимальной коррекции классической модели универсальных вычислений, что на уровне постулатов позволяет устранить первопричины непрерывного воспроизводства разнородных форм представления данных и программ и распространить свойство универсальной программируемости с внутренних ресурсов компьютеров на сетевые.

Обновленная модель представляет собой исчисление древовидных структур, которое становится основой математически однородного поля компьютерной информации и пространства структурно-сложных распределенных вычислений [4]. В этой модели проблемы программирования, интеграции и масштабирования компьютерных решений и распределенных процессов перестают зависеть от размера компьютерной среды и технических особенностей компьютеров и сетей.

В работе представлена экспериментальная реализация системы распределенных вычислений, осуществляемых в модели исчисления древовидных структур. Распределенные вычисления осуществляются в предположении недетерминированной доступности компьютеров в сетях. При этом целостность распределенного процесса сохраняется при отказах или при непредсказуемом включении/выключении компьютеров. Для присоединения компьютеров к распределенной системе требуется установка простейшей программы сетевой связи, которая передается при регистрации IP-адресов компьютеров.

Экспериментальная система испытана на задаче поиска оптимальных конфигураций коммутирующих сетей на основе комбинаторных методов построения квазиполных графов (симметричных блок-схем). Алгоритм решения задачи допускает разбиение на многие слабосвязанные фрагменты. Это позволяет исполнять их одновременно на многих компьютерах связанных через Интернет.

Представленная в данной работе система распределенных вычислений построена в двухуровневой сетевой архитектуре «master-slaves», которая имеет схожие черты с системой метакомпьютинга X-Com [5,6]. Предлагаемая система принципиальное отличается наличием единого адресного пространства распределенных вычислений, охватывающего оперативную память компьютеров, осуществляющих распределенные вычисления, в котором обеспечивается возможность «бесшовного» программирования алгоритмов. Система реализована в системе программирования Парсек, реализующей исчисление древовидных структур в сетевых ресурсах, как единая программа, подпрограммы которой запускаются и параллельно исполняются на отдаленных компьютерах. Особенность данной системы, также в том, что любое подмножество компьютеров, входящее в пул, может одновременно иметь статус «master», с которого можно запускать и одновременно исполнять на всем пуле компьютеров разные задачи.

Программная архитектура автоматически реконфигурируемой системы отказоустойчивых распределенных вычислений

Для распространения свойств универсальной программируемости системы ПАРСЕК [7,8] с внутренних ресурсов компьютеров на распределенные вычислительные ресурсы локальных и глобальных сетей с использованием библиотеки функций управления протоколом TCP/IP реализован базис управления распределенными вычислениями в сетевой архитектуре Peer-to-Peer. Особенность построенного решения – организация распределенных структурно-сложных вычислений в едином адресном пространстве, охватывающем оперативную память компьютеров, предоставляющих ресурсы через сети.

Распределенные вычисления осуществляются в предположении недетерминированности доступа к компьютерным ресурсам в сетях. При этом целостность распределенного процесса сохраняется при отказах или при непредсказуемом включении/выключении компьютеров. Для присоединения компьютеров к распределенной системе требуется установка простейшей программы сетевой связи, которая передается при регистрации IP-адресов компьютеров.

Функционально полный набор действий по управлению распределенными ресурсами приведен на рис.1.

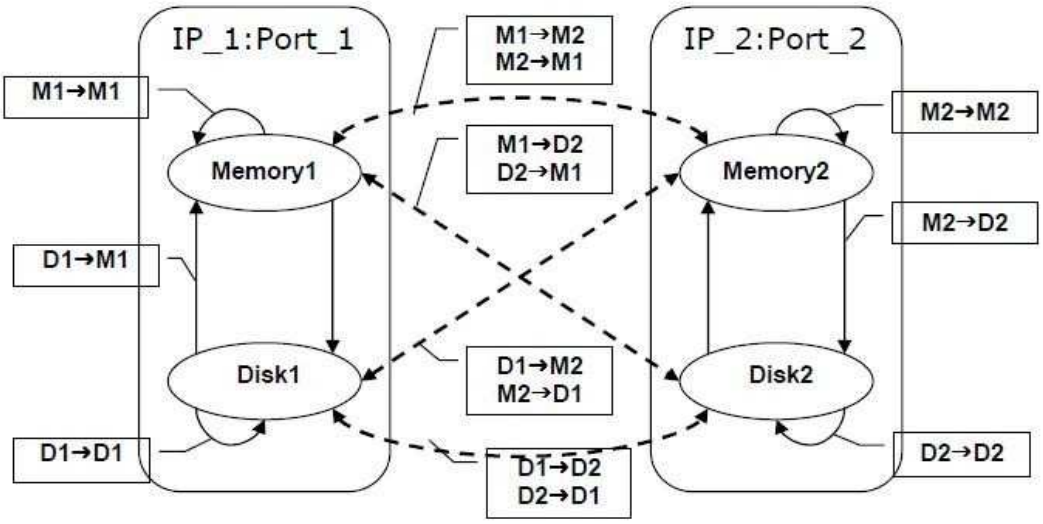

Рис. 1. Концептуальная схема сетевой архитектуры системы распределенных вычислений

В таблице 1 иллюстрируется представление двоичных деревьев в едином адресном пространстве распределенной оперативной памяти.

Таблица 1. Формат единого адресного пространства распределенной оперативной памяти

Формат сквозного адреса (80bit):		
IP адрес (32bit)	IP порт (16bit)	Адрес в ОЗУ (32bit)

На рис.2. приведена структура программных модулей, реализующих программную архитектуру распределенной обработки.

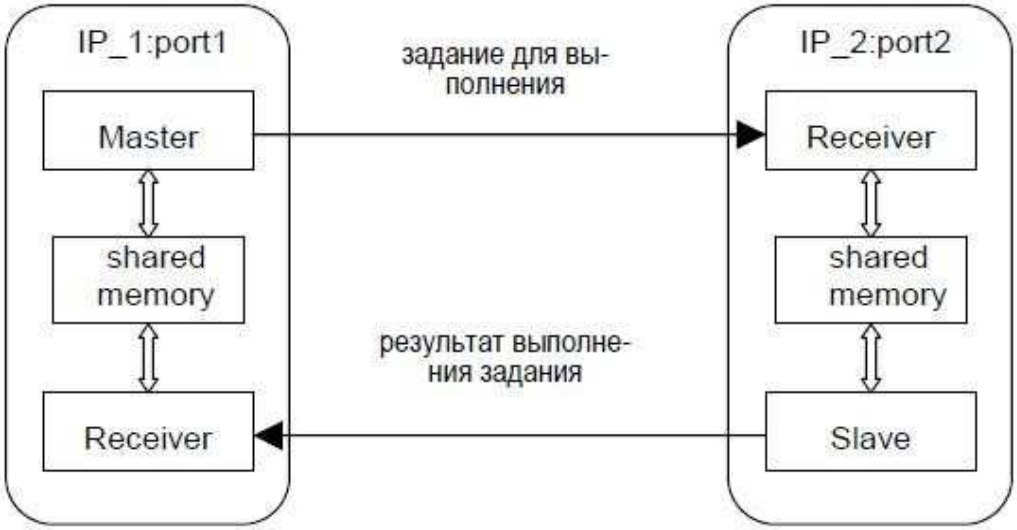

Рис. 2. Структура программных модулей для распределенных вычислений

На каждом компьютере запускаются два процесса, взаимодействующих через системный буфер памяти:

1. обязательный резидентный процесс Receiver ("стандартная" программа приема сообщений с дальних компьютеров);
2. процесс прикладной программы.

Выделяется компьютер со статусом "Master", остальным придается статус "Slave":

- на "Master" прикладной процесс исполняет головную часть прикладной программы;
- на "Slave" прикладной процесс реализует программу универсального интерпретатора базисных функций языка ПАРСЕК, с добавленными на этапе трансляции функциями прикладной программы, назначенными для отдаленного исполнения.

Каждый компьютер может находиться в одном из трех состояний: "Свободен", "Выполняется задание", "Недоступен".

Алгоритм управления подзадачами реализуется на компьютере со статусом "Master". Он периодически

опрашивает незадействованные компьютеры в пуле зарегистрированных компьютеров. Каждый не ответивший на опрос компьютер определятся как "Недоступный". Ответившие – как "Свободный".

При запуске задачи формируется очередь исполняемых подзадач (фрагментов) подзадач к пулу доступных компьютеров, а именно к компьютерам, находящимся в состоянии "Свободен".

При наличии свободных компьютеров алгоритм производит автоматический запуск очередной подзадачи на очередной отдаленный свободный со статусом "Slave". Получивший подзадачу компьютер переводится в состояние "Выполняется". В ходе выполнения каждая подзадача периодически сохраняет свое текущее состояние в файле на компьютере со статусом "Master". В случае отказа или выключения компьютера со статусом "Slave" это сохраненное состояние позволяет возобновить процесс на другом отдаленном компьютере со статусом "Slave" (посредством пересылки сохраненного файла-состояния и запуска подзадачи).

По мере завершения исполнения фрагментов задачи компьютеры переводятся из состояния "Выполняется" в состояние "Свободен", что позволяет повторно использовать компьютеры со статусом "Slave" для подзадач, остающихся в очереди на исполнение.

По мере исчерпания очереди подзадач, процесс распределенной обработки задачи завершается.

Программирование автоматически реконфигурируемой системы отказоустойчивых вычислений в системе ПАРСЕК

Модель обменов через сокеты: обмены в протоколе TCP/IP (точка-точка, широковещание); установка и отключение каналов; статический набор каналов; динамический набор каналов.

Функция Receiver: установка каналов связи; сокеты и буферирование (режим циклического обегания); буфер разделяемой памяти; синхронизация буфера при приеме строки данных из сети; синхронизация буфера при передаче данных прикладному процессу.

Функция Sender: инициация канала связи; подготовка пакета для передачи строки данных; фазы синхронизации передачи данных в отдаленный компьютер; передача длинной строки (с превышением размера буфера).

Модуль "Master": в главный раздел `rout помещается ведущая программа приложения; в раздел `subrout помещаются все определяемые функции приложения, в том числе для отдаленного исполнения.

Модули "Slave": в главном разделе `rout резидентно размещена головная часть программы интерпретатора отдаленного исполнения встроенных и определяемых функций; в раздел `subrout резидентно помещены внутренние функции обработки деревьев, добавляются на этапе трансляции функции приложения.

Пространство IP-адресов локальных и глобальных сетей по различным причинам изначально не являются равнодоступным. Это и активное использование "непрямых" (динамических) IP-адресов, разнообразные, несогласованные между собой, политики провайдеров, сетевые защитные экраны и т.п.

Для организации распределенных вычислений через Интернет требуется единое сетевое пространство IP-адресов компьютеров, входящих в различные локальные сети и сети различных провайдеров. Единое пространство виртуальных IP-адресов, обеспечивающее их равнодоступность можно формировать посредством VPN технологии. В данной работе использована VPN технология с открытой лицензией.

Пространство равнодоступных IP-адресов, созданных по VPN-технологии (рис. 3), представлено на сайте http://pardon.idm.ru/.

Рис. 3. Скриншот страницы сайта

На сайте приведены виртуальные IP-адреса зарегистрированных компьютеров, образующих пул компьютеров разработанной системы распределенных вычислений. Для присоединения компьютеров, имеющих выход в Интернет к данному пулу требуется пройти процедуру регистрации, в ходе которой потенциальный

участник получает для своего компьютера виртуальный IP-адрес. Для присоединения зарегистрированного компьютера к пулу необходимо установить на нем программу ресивера, дистрибутив которой передается при регистрации.

В макетной версии системы распределенной обработки допускается присоединение до 256 компьютеров.

Задание на исполнение может быть запущено с любого компьютера, входящего в пул. Для этого требуется установка мониторной программы "Мастер". После запуска программы "Мастер" на экране появляется его окно, показано на скриншоте (компьютеры идентифицируются IP-адресом и, через двоеточие, номером порта), см. рис. 4.

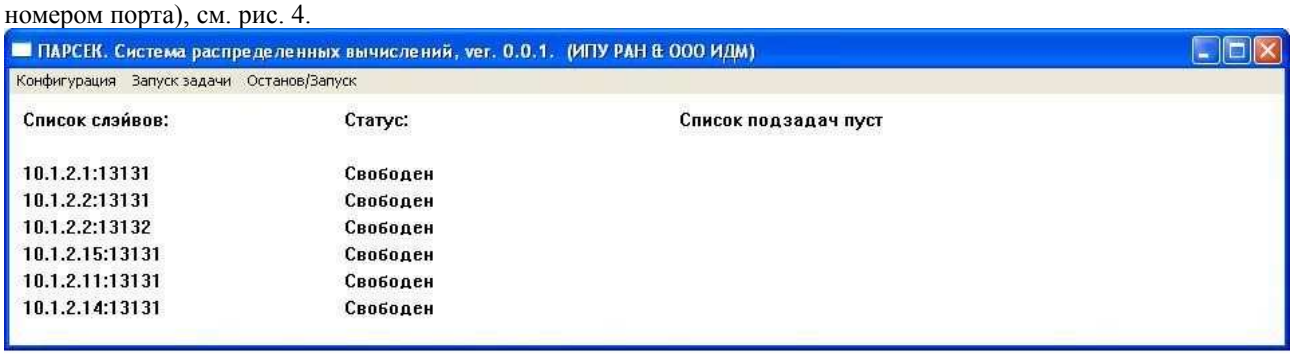

Рис. 4. Идентификация пула компьютеров

Меню управления заданиями имеет следующие опции: конфигурация, включающая параметры задачи (число фрагментов, время) и доступные компьютеры (редактирование списка доступных компьютеров); запуск задачи, включающая инициализацию (первый шаг запуска задания – осуществляется компоновка выбранного задания), старт вычислений (запуск на выполнение); останов/запуск (включение/выключение компьютеров из числа доступных).

Решение задачи поиска оптимальной конфигурации кольцевых коммутаторов

Повышение надежности распределенных вычислений в условиях непредсказуемых выключений компьютеров сетевой среды экспериментально подтверждено на прогонах задачи большой вычислительной сложности, в которой отыскиваются оптимальные конфигурации коммутирующих сетей на основе комбинаторных методов построения квазиполных графов (симметричных блок-схем). Метод решения этой задачи изложен в работе [9].

Алгоритм решения задачи допускает разбиение на многие слабосвязанные фрагменты. Это позволяет исполнять их одновременно на многих компьютерах сетей, что обеспечивает сокращение времени счета пропорциональное числу вовлеченных компьютеров.

При выборе опции "Параметры задачи" в окне Мастера в меню Конфигурация появляется меню выбора одного из заданий: число фрагментов, время (см. рис. 5).

В данном случае выбрана задача, состоящая из 12 параллельных фрагментов, оценочное время выполнения каждого – 5 мин.

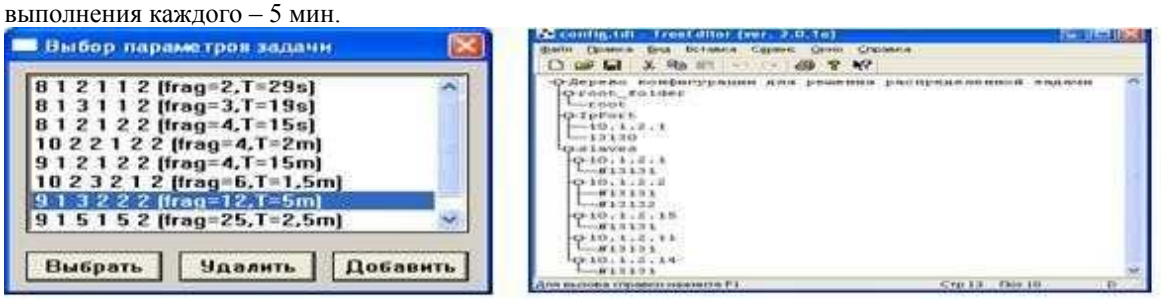

Рис. 5. Опция "Параметры задачи"

На скриншоте (рис. 6) показано состояние процесса запуска после захода в меню "Запуск задачи" и выбора опции "Инициация задачи".

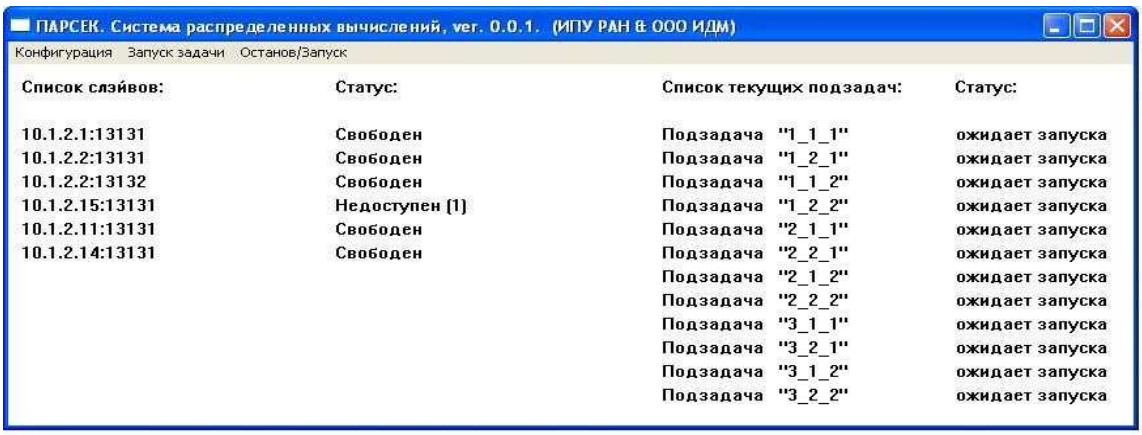

Рис. 6. Скриншот меню "Запуск задачи", опция "Инициация задачи"

Дальнейший ход процесса исполнения очереди текущих подзадач, а также окончание задачи, "опустошение" списка подзадач и выдача окна результатов показаны на рис. 7.

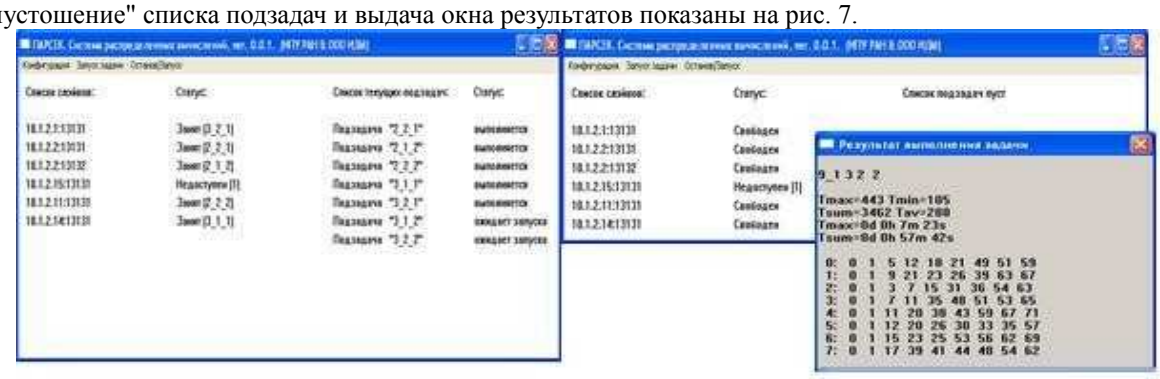

Рис. 7. Скриншоты выполнения и окончания задачи

Выводы

В ходе испытаний экспериментальной системы распределенных вычислений, построенной в новой модели на основе исчисления древовидных структур, подтверждена осуществимость "бесшовного" программирования распределенных вычислений в едином адресном пространстве математически однородного поля компьютерной информации, свободно распространяемого на оперативную память компьютеров, связанных глобальными сетями.

При этом показано:

- распределенные вычисления осуществляются в ресурсах общедоступных сетей без добавления каких-либо новых системных программных слоев промежуточного ПО;
- система программирования ПАРСЕК в едином формализме исчисления древовидных структур распространяет свойство универсальной программируемости структурно-сложных распределенных вычислений на ресурсы глобальных сетей;
- трудоемкость «бесшовного» программирования распределенных вычислений в системе ПАРСЕК практически не зависит от количества задействованных компьютеров, связанных сетями;
- эффективность системы выражается относительно малой долей расхода времени на управление распределенными процессами в условиях непредсказуемого отключения отдаленных компьютеров.

При полном отсутствии атрибутов коммерчески значимых программных продуктов, функциональные возможности пробной системы отчасти сопоставимы с известной системой X-COM [5,6]. Цель дальнейшего развития – построение предкоммерческой версии системы ПАРСЕК, которая позволит программировать распределенные вычисления в свободно масштабируемой сетевой архитектуре «Peer-to-Peer».

ЛИТЕРАТУРА:

1. I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations // International J. Supercomputer Applications, 15(3), 2001.
2. I. Foster, C. Kesselman, (Editors), The Grid: Blueprint for a New Computing Infrastructure (2nd Edition),

San Mateo, CA: Morgan Kaufmann, 2004.
3. В.Н. Коваленко, Д.А. Корягин. Грид: истоки, принципы и перспективы развития // Информационные технологии и вычислительные системы. 2008, №4. С. 38 — 50.
4. Ю.С. Затуливетер. На пути к глобальному программированию // Открытые системы. 2003, №3. С. 46- 47. -URL: http://www.osp.ru/os/2003/03/182704/.
5. Вл.В. Воеводин Решение больших задач в распределенных вычислительных средах //Автоматика и Телемеханика. 2007, N5, С. 32-45. -URL: http://parallel.ru/news/x-com.pdf, http://www.mathnet.ru/links/84cdf4f2446515ec09a6a695a40335d0/at983.pdf
6. Вл.В. Воеводин, Ю.А. Жолудев, С.И. Соболев, К.С. Стефанов Эволюция системы метакомпьютинга X-Com. -URL:http://www.ict.edu.ru/vconf/files/11865.pdf.
7. Ю.С. Затуливетер, Т.Г. Халатян ПАРСЕК - язык компьютерного исчисления древовидных структур с открытой интерпретацией. Стендовый вариант системы программирования / -М., 1997. (Препр. ИПУ РАН). – 71 с.
8. Ю.С. Затуливетер, А.В. Топорищев Язык Парсек: программирование глобально распределенных вычислений в модели исчисления древовидных структур // Проблемы управления. №4. 2005. С.12-20.
9. М.Ф. Каравай, П.П. Пархоменко, В.С. Подлазов Комбинаторные методы построения двудольных однородных минимальных квазиполных графов (симметричных блок-схем) // АиТ. №. 2. 2009. С. 153-170.