

УДК 004.75

ИНТЕРФЕЙС ВЗАИМОДЕЙСТВИЯ ВЫЧИСЛИТЕЛЕЙ РВС*

П.В. МИЩЕНКО¹, Я.В. СИЛОВ²

¹ 630073, РФ, г. Новосибирск, пр. К. Маркса, 20, Новосибирский государственный технический университет, старший преподаватель кафедры вычислительной техники, аспирант. E-mail: mpv-777@ya.ru

² 630073, РФ, г. Новосибирск, пр. К. Маркса, 20, Новосибирский государственный технический университет, студент кафедры вычислительной техники. E-mail: ya.yuaann@yandex.ru

Статья посвящена актуальным вопросам создания отечественных программных средств реализации параллельных вычислений. Описывается опыт разработки интерфейса организации распределенных вычислительных систем с программируемой структурой (РВС ПС). В основе решения лежит концепция однородных вычислительных систем (ОВС), базирующаяся на модели коллектива вычислителей. Концепция ОВС была сформулирована в Институте математики СО РАН СССР под руководством Э.В. Евреина в 1962 году [1]. Подход особенно актуален в условиях распространенности информационных сетей, объединяющих многочисленные персональные компьютеры (ПК). Необходимо разработать программные средства, позволяющие объединять разрозненные вычислители в систему и выделять необходимое количество подсистем для решения конкретных прикладных задач. При разработке программного обеспечения (ПО) необходимо исключить вмешательство в аппаратную и программную части вычислителей. Программируемость структуры вычислительной системы обеспечивается на этапе объединения вычислителей в подсистему программными средствами интерфейса – Агента РВС ПС. Разработка Агента РВС ПС велась на языке программирования C# с применением Microsoft .NET Framework версии 4.5. Архитектура приложения была реализована в соответствии с шаблоном программирования MVC. Применение подобных шаблонов позволяет неограниченно расширять функционал приложения. Для реализации графического интерфейса приложения были использованы Windows Forms. Описываемое в статье программное обеспечение (Агент РВС ПС) было апробировано при организации экспериментальной РВС ПС кафедры вычислительной техники НГТУ на базе существующей сети персональных компьютеров. В настоящее время проводятся работы по расширению функциональных возможностей разработанного ПО.

Ключевые слова: распределенные вычислительные системы, параллельный алгоритм, коллектив вычислителей, стек TCP/IP, архитектура клиент–сервер, GUID, NAT, VPN

DOI: 10.17212/2307-6879-2016-1-110-122

* Статья получена 01 октября 2015 г.

ВВЕДЕНИЕ

Исследования в области распределенных вычислительных систем (РВС) ведутся с середины XX века. В Институте математики СО РАН СССР под руководством Э.В. Евреинова в 1962 году была сформулирована концепция однородных вычислительных систем [4, 5]. В России и за рубежом выполнен ряд фундаментальных работ, посвященных проблемам организации высокопроизводительных вычислительных средств [1–3, 6–7]. Одним из подходов организации распределенной вычислительной системы с программируемой структурой (РВС ПС) является построение систем на базе персональных компьютеров (ПК), объединенных в локальную или глобальную сеть [12].

Подход основан на модели коллектива вычислителей, под которым понимается совокупность вычислительных средств, динамично программным способом настраиваемая на решение конкретной задачи. В нашем случае РВС ПС представляет собой совокупность территориально разнесенных ЭВМ (или ВС), соединенных между собой сетью связей так, что между элементами системы обеспечиваются необходимые обменные взаимодействия, предоставляющие возможность реализации параллельных алгоритмов (Р-алгоритмов). Поскольку Р-алгоритм представляет собой совокупность последовательных алгоритмов (ветвей), в каждом из которых имеется по крайней мере один подалгоритм, его следует рассматривать как операнд обменных взаимодействий. При этом каждая ветвь параллельного алгоритма выполняется параллельно с другими ветвями. На фазе обмена реализуются необходимые обменные взаимодействия, которые определяются структурной схемой решаемой задачи. Несмотря на огромное разнообразие задач большой размерности (по объему данных, сложности расчетов, т. е. требуемому объему памяти и/или процессорному времени), все взаимодействия между ветвями параллельного алгоритма могут быть сведены к конечному ограниченному множеству основных (типовых) процедур: трансляционных, трансляционно-циклических, коллекторных, дифференцированных обменов, обменов сдвигами [14]. Эти обменные взаимодействия осуществляются с помощью набора системных операций: операций настройки, обмена (передачи и приема), обобщенного условного и безусловного переходов, синхронизации. Приведенный набор системных операций является полным, поскольку с его помощью можно осуществить самые разнообразные схемы организации параллельного вычислительного процесса [4, 5].

Организацию вышеупомянутых межмашинных взаимодействий предлагается осуществить с помощью наиболее популярного средства организации сетей – стека протоколов ТСП/IP. Данный стек протоколов применяется для построения как локальных, так и глобальных сетей, включая интернет [8, 9].

1. ПОСТАНОВКА ЗАДАЧИ

Для организации и согласованного функционирования системы необходимо разработать программное обеспечение (ПО РВС ПС), включающее в себя библиотеку и интерфейс взаимодействия вычислителей. Библиотека должна представлять собой инструментарий для реализации параллельных алгоритмов. Интерфейс – Агент РВС ПС – предназначен для управления запуском параллельных алгоритмов одновременно на нескольких компьютерах. Основными задачами Агента РВС ПС является организация системы и выделение подсистем для решения конкретных прикладных задач. Для выполнения этих функций разумным представляется организовать опрос всех ПК сети с целью определения ПК, доступных для вычислений. После организации подсистемы Агент должен выполнить рассылку необходимых для вычислений файлов – Р-алгоритма и исходных данных. По окончании вычислений Агент должен собрать отчеты о завершении счета от всех вычислителей и освободить их, присвоив соответствующий статус.

С целью осуществления контроля функционирования системы необходимо предусмотреть систему статусов, которая отображала бы степень участия конкретного ПК в процессе вычислений и предоставляла возможность обнаружения ошибок и сбоев. При разработке ПО необходимо исключить вмешательство в аппаратную и программную часть вычислителей. Структура РВС задается на этапе объединения вычислителей в подсистему посредством программных средств Агента РВС ПС.

Статья описывает опыт разработки Интерфейса организации распределенных вычислительных систем с программируемой структурой и выделения подсистем для решения конкретных прикладных задач.

2. ИНТЕРФЕЙС ВЗАИМОДЕЙСТВИЯ

Следует различать две стороны взаимодействия:

- 1) исполнитель (по умолчанию);
- 2) инициатор (по запросу пользователя).

При запуске агента машина, на которой он был запущен, автоматически становится исполнителем.

Сетевое взаимодействие в Агенте

Все сетевое взаимодействие в РВС определяется сетевыми настройкам. Задаются три параметра: IP-адрес узла – машины, на которой запущен Агент; маска сети; порт. Изначально, все машины, на которых запущен Агент, являются исполнителями. При этом с каждой из них можно запустить процесс организации подсистемы. В этом случае ПК становится инициатором.

С использованием IP-адреса узла и маски высчитывается адрес сети, который будет использоваться инициатором для поиска исполнителей в автоматическом режиме. Поиск ограничен одним сегментом сети в связи с тем, что для установки TCP-соединения требуется фиксированное время. Если машина ожидает подключения, то соединение будет установлено мгновенно. Как показали эксперименты, инициатор будет ожидать ответа 20 секунд. Также экспериментально было выяснено, что среднее суммарное время ожидания подключений будет более 7 минут на один сегмент сети. В связи с этим максимальное время ожидания установки соединения было снижено до 1,5 секунды, что позволило сократить среднее время поиска вычислителей PBC до 30 секунд. Для поиска машин вне сегмента сети доступна возможность подключения напрямую по IP-адресу. При поиске вручную соединение будет установлено за фиксированное время, как это заложено протоколом TCP [8, 9].

Для установки соединения рекомендуется использовать порт 56001, так как он не должен пересекаться с другими приложениями и сервисами.

Стоит отметить, что Агент является независимым от физической среды передачи данных и программных средств сетевого взаимодействия. Предполагается, что Агент способен работать как в локальных сетях, так и в сети интернет и с VPN-сетями. Однако, организация подсистем через сеть интернет представляется проблематичной ввиду сложности установки соединения с машиной за NAT без трансляции порт-адресов на маршрутизаторе. Решение этой проблемы может быть достигнуто двумя способами. Первое из них заключается в создании сетевого сервиса, который будет выступать в роли посредника при соединении через NAT. Второе решение проблемы с NAT – использование VPN-сетей.

Для организации подсистем и осуществления контроля функционирования PBC введена следующая система служебных сообщений:

- Hello – пакет приветствия, в ответ на который исполнители должны отправить информацию о своих технических характеристиках, полученных на основе теста. Подробное описание тестирования будет дано ниже.
- NotDeleteFiles – пакет, позволяющий отменить удаление файлов после завершения вычислений. Автоматическое удаление предусмотрено с целью предотвращения использования устаревших данных для вычислений. Пакет отправляется инициатором исполнителям перед отправкой файлов, связанных с вычислениями.
- RunFile – пакет, который идет перед передачей исполняемого файла (P-алгоритма). Пакет необходим для корректной интерпретации входного потока данных от инициатора. Помимо этого, он служит инструментом для идентификации файла P-алгоритма.
- Data – пакет, который идет перед передачей файла данных. Пакет необходим для корректного определения входного потока данных. Является при-

знаком того, что в потоке данных будет файл, относящийся к исходным данным для вычислений.

- StartCalc – пакет, уведомляющий о необходимости запуска Р-алгоритма. Отправляется инициатором всем исполнителям.
- FinishCalc – пакет, сообщающий об окончании вычислений на конкретном исполнителе. Отправляется исполнителями инициатору.
- StopCalc – пакет, уведомляющий о необходимости прервать вычисления. Отправляется инициатором всем исполнителям. В этом случае все исполнители прерывают выполнение исполняемого файла, полученного после RunFile, и разрывают соединение с инициатором.
- Free – пакет, уведомляющий о том, что исполнитель свободен и больше не относится ни к какой из подсистем РВС. Отправляется инициатором исполнителю.

Рассмотрим процесс функционирования со стороны исполнителя.

При каждом запуске приложения происходит тестирование системы, которое определяет ID машины, основанный на GUID Windows, вычисляет объем доступной RAM и получает значение тактовой частоты процессора.

GUID (Globally Unique Identifier) – статистически уникальный 128-битный идентификатор. Его главная особенность – уникальность, которая позволяет создавать расширяемые сервисы и приложения без опасения конфликтов, вызванных совпадением идентификаторов.

GUID считывается из системного реестра ОС Windows (regedit.exe) [15]. При применении на GUID хэш-функции мы получаем уникальный ID каждого исполнителя.

RAM вычисляется путем нагрузочного тестирования. Происходят попытки выделения по 128 МБ оперативной памяти до тех пор, пока не возникнет ошибка выделения памяти. Так как приложение 32-разрядное, то количество выделенной памяти будет примерно равным реально доступной RAM.

Тактовая частота процессора, как и GUID, берется из системного реестра ОС Windows.

Результаты тестов используются инициатором для построения подсистемы. Информацию о своей машине пользователь может посмотреть в окне «О программе» (рис. 1).

При первом запуске приложения у пользователя запрашиваются вышеописанные сетевые настройки (рис. 2). После нажатия кнопки «сохранить» автоматически запускается механизм ожидания подключений.

Каждый исполнитель выступает в роли TCP-сервера, который ожидает соединения на порт, заданный в настройках. Исполнитель может принять только одно соединение, что исключает вероятность того, что исполнитель будет участвовать в нескольких вычислениях одновременно. Участие в не-

скольких вычислениях возможно только в том случае, если приложение «Агент» будет запущено несколько раз на одной машине с разными значениями порта. Но такой подход не рекомендуется, так как параллельное участие в нескольких вычислениях негативно скажется на общей производительности исполнителя.



Рис. 1. Окно настроек

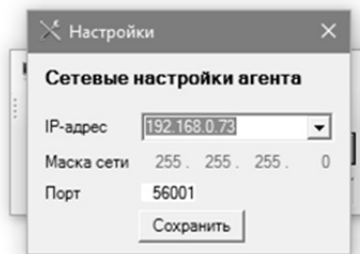


Рис. 2. Окно настроек

Как только на указанный порт придет TCP-соединение, исполнитель ожидает получения пакета Hello. Этот пакет служит для исполнителя сигналом о том, что необходимо отправить информацию о своих технических характери-

стиках (основанных на тесте, который выполнен при запуске агента) инициатору. После отправки технических характеристик исполнитель меняет свой Free статус на Wait. Исполнитель в состоянии Wait ожидает от инициатора команд или настроек, описанных ранее. При этом для других инициаторов исполнитель не будет доступен, пока не получит статус Free. Этот механизм предотвращает захват исполнителя несколькими инициаторами.

Полученные файлы помещаются в папку TEMP в директории, где был запущен Агент. После запуска со стороны инициатора вычислений исполнитель ожидает завершения работы исполняемого файла. По завершении инициатору отправляется сообщение FinishCalc (вычисления завершены). После прихода пакета Free исполнитель, в зависимости от полученных настроек, удаляет папку TEMP. После этого он закрывает соединение с инициатором и Агент перезапускается.

Со стороны инициатора функционирование выглядит несколько иначе.

Каждый исполнитель может стать инициатором по нажатию кнопки «Начать вычисления». В этом случае машина перестает ожидать входящего соединения и появляется графический интерфейс для организации подсистемы. На данный момент реализованы два способа поиска машин: автоматический и ручной (рис 3).

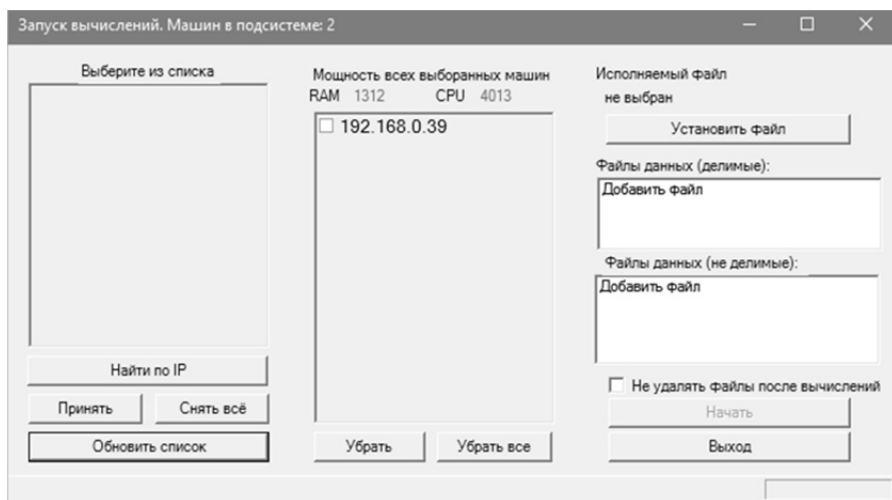


Рис. 3. Окно запуска вычислений

Автоматический режим заключается в попытке инициатора подключиться ко всем машинам в сегменте сети, заданной маской в настройках. Все маши-

ны, которые ответили на запрос соединения, – исполнители, готовые участвовать в вычислениях. Эти машины по мере установления соединений помещаются в список доступных для вычислений.

Ручной заключается в том, что необходимо открыть специальное окно для поиска машины по IP. После того как пользователь вводит IP искомой машины и нажимает «Соединиться», происходит попытка подключения. Если соединение установлено, машина появляется в списке доступных, а окно поиска автоматически закрывается. В противном случае выдается ошибка.

Подключение как в автоматическом режиме, так и в ручном подразумевает попытку подключиться к исполнителю как к ТСР-серверу.

Сразу после установления соединения с исполнителем инициатору отправляется пакет Hello, и сразу за ним ожидается ответ, в котором будет содержаться информация о технических характеристиках исполнителя. Для организации подсистемы необходимо выбрать машины из списка доступных путем выбора их с помощью CheckBox и нажатия «Принять». После выбора машин они появляются в соответствующем окне. Для всех выбранных машин отображается их суммарный объем оперативной памяти и тактовая частота процессора. Количество выбранных машин отображается в заголовке окна запуска вычислений.

Перед запуском можно выбрать следующие виды файлов: исполняемый файл вычислений, делимые файлы данных, неделимые файлы данных.

Исполняемый файл вычислений – Р-алгоритм, разработанный с использованием библиотеки. Его отсутствие не позволит запустить вычисления. Этот файл будет запущен на всех исполнителях и инициаторе. Он должен иметь расширение .exe.

Исходными данными могут быть файлы любых расширений, при этом стоит отметить, что в процессе функционирования могут отправляться так называемые делимые и неделимые файлы (рис. 3).

Делимые файлы данных – после выбора такого файла обязателен выбор соответствующего файла-разделителя. Файл-разделитель при запуске принимает два аргумента: название файла данных и количество частей, на которое необходимо его разделить. Количество частей будет равным количеству вычислителей подсистемы. Поэтому оператор на стороне инициатора должен учитывать число вычислителей в зависимости от решаемой задачи.

Файлы данных не являются обязательными для запуска вычислений, но они должны быть выбраны в зависимости от решаемой задачи.

Перед запуском вычислений нужно осуществить выбор о необходимости удаления всех файлов, задействованных в вычислениях. В случае соответствующего выбора по окончании счета файлы будут удалены.

По нажатию кнопки «Запустить вычисления» всем машинам, которые не были выбраны для вычислений, отправляется пакет Free (свободен). Это значит, что они будут доступны для организации других подсистем.

Перед созданием и делением файлов все выбранные файлы копируются в директорию TEMP в папке, откуда был запущен Агент.

После этого собирается файл `iplist.txt`, в который записаны IP-адреса всех машин в подсистеме. Этот файл добавляется в список неделимых файлов.

На стороне инициатора имеется список исполнителей, и для каждого из них имеется список файлов, которые необходимо отправить каждому из исполнителей. У каждого исполнителя список файлов изначально пуст. В первую очередь всем исполнителям в список добавляются исполняемый файл вычислений и неделимые файлы данных. Дальше начинается деление делимых файлов данных на части. Каждая часть попадает соответствующему исполнителю в список файлов. После деления файлов всем исполнителям отправляется информация о том, необходимо ли удалять принятые файлы после вычислений.

После этого начинается передача всех файлов. Для передачи одного файла исполнителю отсылается сообщение `RunFile` или `Data` в зависимости от того, передается исполняемый файл вычислений или файл данных. После того как последний файл передан последнему исполнителю, инициатор отправляет всем исполнителям сообщение `StartCalc` («Запустить вычисления»). Сразу после отправки этого сообщения инициатор запускает вычисления и на своей машине. После запуска вычислений инициатор ожидает от всех исполнителей сообщения о том, что вычисления завершены.

После того как ответ о завершении всех вычислений получен от всех, подсистема реорганизуется (всем исполнителям отправляется пакет `Free`) и все ее члены снова становятся доступными для других вычислений.

ЗАКЛЮЧЕНИЕ

Был разработан Агент, с помощью которого можно управлять запуском параллельных алгоритмов одновременно на нескольких компьютерах. Агент способен заниматься организацией системы и выделением подсистем для решения конкретных прикладных задач. Для выполнения этих функций организован опрос всех ПК сети с целью определения ПК, доступных для вычислений. После организации подсистемы Агент способен выполнить рассылку необходимых для вычислений файлов – Р-алгоритма и исходных данных. По окончании вычислений Агент собирает отчеты о завершении счета от всех вычислителей и освобождает их, присвоив соответствующий статус.

С использованием разработанного программного обеспечения на кафедре вычислительной техники НГТУ организован фрагмент экспериментальной PBC ПС – научно-образовательная вычислительная система «Пара» (НОВС «Пара»). Система использует ПК лаборатории сетевых средств и технологий, связанных сетью Fast Ethernet. Каждый ПК идентифицируется в системе по IP-адресу. На каждом из компьютеров установлено системное программное обеспечение PBC ПС. Благодаря многозадачности операционной системы компьютеры, участвующие в вычислениях, могут использоваться и для других целей во время вычислений.

СПИСОК ЛИТЕРАТУРЫ

1. *Ефимов А.В., Мамойленко С.Н., Перышкова Е.Н.* Организация функционирования распределенных вычислительных систем при обработке масштабируемых задач // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2011. – № 2 (15). – С. 51–60.
2. *Мамойленко С.Н.* Организация функционирования распределенных вычислительных систем в мультизадачных режимах: дис. ... д-ра техн. наук: 05.13.15. – Новосибирск, 2012. – 223 с.
3. *Павский В.А.* Анализ функционирования вычислительных и сложных технологических систем: дис. д-ра техн. наук: 05.13.15. – Новосибирск, 2007. – 316 с.
4. *Евреинов Э.В.* Однородные вычислительные системы, структуры и среды. – М.: Радио и связь, 1981. – 208 с.
5. *Хорошевский В.Г.* Архитектура вычислительных систем. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. – 520 с.
6. *Хорошевский В.Г.* Распределенные вычислительные системы с программируемой структурой // Вестник СибГУТИ. – 2010. – № 2 (10). – С. 3–41.
7. Масштабируемый инструментарий параллельного мультипрограммирования пространственно-распределенных вычислительных систем / В.Г. Хорошевский, М.Г. Курносков, С.Н. Мамойленко, К.В. Павский, А.В. Ефимов, А.А. Пазников, Е.Н. Перышкова // Вестник СибГУТИ. – 2011. – № 4. – С. 3–18.
8. *Олифер В.Г., Олифер Н.А.* Компьютерные сети: принципы, технологии, протоколы. – 3-е изд. – СПб.: Питер, 2006. – 958 с.
9. *Куроуз Дж., Росс К.* Компьютерные сети: многоуровневая архитектура Интернета. – 2-е изд. – СПб.: Питер, 2004. – 765 с.

10. *Мищенко П.В., Губарев В.В., Мищенко В.К.* Распределенная вычислительная система кафедры вычислительной техники НГТУ // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов: сборник статей Всероссийской научно-практической конференции, Барнаул, 27 февраля 2015 г. – Барнаул, 2015. – С. 150–155.

11. *Рихтер Д.* CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. – М.: Питер, 2013. – 896 с.

12. *Мищенко П.В., Белобородова З.Е.* Современные подходы к реализации высокопроизводительных вычислений // Наука и мир. – 2014. – № 12 (16). – С. 54–57.

13. *Мищенко П.В.* Исследование и разработка методов организации высокопродуктивных вычислительных систем // Инновации на основе информационных и коммуникационных технологий, ИНФО–2015: материалы международной научно-практической конференции, Сочи, 1–10 октября 2015 г. – М.: НИУ ВШЭ, 2015. – С. 209–210.

14. *Губарев В.В., Мищенко В.К., Мищенко П.В.* Средства межмашинного взаимодействия для реализации параллельных алгоритмов в распределенных вычислительных системах // Многоядерные процессоры, параллельное программирование, ПЛИС, системы обработки сигналов: сборник статей Всероссийской научно-практической конференции, Барнаул, 28 февраля 2013 г. – Барнаул, 2013. – С. 149–152.

15. *Кокорева О.И.* Реестр Windows 7. – СПб.: БХВ-Петербург, 2010. – 704 с.

Мищенко Полина Валерьевна, аспирант, старший преподаватель кафедры вычислительной техники Новосибирского государственного технического университета. E-mail: mpv-777@ya.ru

Силов Ян Викторович, студент кафедры вычислительной техники Новосибирского государственного технического университета. E-mail: ya.yuaann@yandex.ru

Compute nodes interaction interface *

P.V. Mishchenko¹, Y.V. Silov²

¹ Senior Lecturer, Department of Computer Engineering, graduate student

² Student of computer engineering

This article considers up-to-date problems of parallel computing systems development in Russia. The study describes an engineering of an interface for HPC organization and for its partition to specific subsystems focused on applied tasks. The software has been tested at the of computer engineering department of Novosibirsk State Technical university.

The experience of the development of the Interface organization of distributed computing systems with programmable structure and allocation of subsystems for specific applications. The software has been tested at the department of computer engineering of NSTU. Considerable work is currently underway to expand the software functionality.

Keywords: distributed computing systems, parallel algorithm, computing community, TCP/IP, client-server architecture, GUID, NAT, VPN

DOI: 10.17212/2307-6879-2016-1-110-122

REFERENCES

1. Efimov A.B., Mamoilenko C.H., Peryshkova E.H. Organizatsiya funktsionirovaniya raspredelennykh vychislitel'nykh sistem pri obrabotke masshtabiruemykh zadach [Organization of functioning of distributed computing systems with scalable processing tasks]. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*, 2011, no. 2 (15), pp. 51–60.
2. Mamoilenko S.N. *Organizatsiya funktsionirovaniya raspredelennykh vychislitel'nykh sistem v mul'tizadachnykh rezhimakh*. Diss. dokt. tekhn. nauk [Organization of functioning of the distributed computing system in multi-tasking mode. Dr. eng. sci. diss.]. Novosibirsk, 2012. 223 p.
3. Pavskii V.A. *Analiz funktsionirovaniya vychislitel'nykh i slozhnykh tekhnologicheskikh sistem*. Diss. dokt. tekhn. nauk [Analysis of the functioning of the computing and complex technological systems. Dr. eng. sci. diss.]. Novosibirsk, 2007. 316 p.
4. Evreinov E.V. *Odnorodnye vychislitel'nye sistemy, struktury i sredy* [Homogeneous computing systems, environments, and structure]. Moscow, Radio i svyaz' Publ., 1981. 208 p.
5. Khoroshevskii V.G. *Arkhitektura vychislitel'nykh sistem* [Architecture of computer systems]. Moscow, Bauman MSTU Publ., 2008. 520 p.

* Receive 01 October 2015.

6. Khoroshevskii V.G. *Raspredelennye vychislitel'nye sistemy s programmiruemoi strukturoi* [Distributed computing systems with programmable structure]. *Vestnik SibGUTI*, 2010, no. 2 (10), pp. 3–41.
7. Khoroshevskii V.G., Kurnosov M.G., Mamoienko S.N., Pavskii K.V., Efimov A.V., Paznikov A.A., Peryshkova E.N. Masshtabiruemyi instrumentarii parallel'nogo mul'tiprogramirovaniya prostranstvenno-raspredelennykh vychislitel'nykh sistem [Scalable toolkit for parallel concurrent multiprogramming spatially-distributed computing systems]. *Vestnik SibGUTI*, 2011, no. 4, pp. 3–18.
8. Olifer V.G., Olifer N.A. *Komp'yuternye seti: printsipy, tekhnologii, protokoly* [Computer networks. Principles, technologies, protocols]. 3rd ed. St. Petersburg, Piter Publ., 2006. 958 p.
9. Kurose J., Ross K. *Computer networking: a top-down approach featuring the Internet*. 2nd ed. Boston, MA, Addison Wesley, 2003. 752 p. (Russ. ed.: Kurouz Dzh., Ross K. *Komp'yuternye seti: mnogourovnevaia arkhitektura Interneta*. 2nd ed. St. Petersburg, Piter Publ., 2004. 765 p.).
10. Mishchenko P.V., Gubarev V.V., Mishchenko V.K. [The distributed computing system of the Department of NSTU computing]. *Mnogoyadernye protsessory, parallel'noe programirovanie, PLIS, sistemy obrabotki signalov*: sbornik statei Vserossiiskoi nauchno-prakticheskoi konferentsii [Multi-core processors, parallel programming, FPGA, signal processing system: a collection of articles All-Russian scientific-practical conference], Barnaul, 27 February 2015, pp. 150–155.
11. Richter J. *CLR via C#. Programirovanie na platforme Microsoft .NET Framework 4.5 na yazyke C#* [CLR via C#. Programming Microsoft .NET Framework 4.5 platform in C#]. Moscow, Piter Publ., 2013. 896 p.
12. Mishchenko P.V., Beloborodova Z.E. Sovremennye podkhody k realizatsii vysokoproizvoditel'nykh vychislenii [Modern approaches to the implementation of high performance computing]. *Nauka i mir – Science and world*, 2014, no. 12 (16), pp. 54–57.
13. Mishchenko P.V. [Research and development of methods of organization of high-productivity computing systems]. *Innovatsii na osnove informatsionnykh i kommunikatsionnykh tekhnologii INFO–2015*: materialy mezhdunarodnoi nauchno-prakticheskoi konferentsii [Materials of XII International Scientific and Practical Conference "Innovations based on information and communication technologies" INFO–2015], Sochi, 1–10 October 2015, pp. 209–210.
14. Gubarev V.V., Mishchenko V.K., Mishchenko P.V. [Tools machine to machine to implement parallel algorithms in distributed computing systems]. *Mnogoyadernye protsessory, parallel'noe programirovanie, PLIS, sistemy obrabotki signalov*: sbornik statei Vserossiiskoi nauchno-prakticheskoi konferentsii [Multi-core processors, parallel programming, FPGA, signal processing system: a collection of articles All-Russian scientific-practical conference], Barnaul, 28 February 2013, pp. 149–152.
15. Kokoreva O.I. *Reestr Windows 7* [Register Windows 7]. St. Petersburg, BHV-Petersburg Publ., 2010. 704 p.