

УДК 004.75

## Программная организация высокопродуктивных распределенных вычислительных систем с программируемой структурой\*

П.В. МИЩЕНКО<sup>1</sup>, А.В. БЕЛЯШОВ<sup>2</sup>

<sup>1</sup> 630073, РФ, г. Новосибирск, пр. Карла Маркса, 20, Новосибирский государственный технический университет, старший преподаватель. E-mail: mpv-777@ya.ru

<sup>2</sup> 071100, РФ, г. Курчатов, РГП «Институт геофизических исследований», заведующий отделением. E-mail: abelyashov@igr.kz

Рассматривается актуальная задача разработки отечественных методов и средств организации функционирования распределенных вычислительных систем для малобюджетных предприятий. Особое внимание уделено системности подхода к разработке программных средств, характеризующейся комплексированием программ организации функционирования РВС с модулями расширения функциональных возможностей, наборами параллельных алгоритмов. Функционирование РВС увязывается с критерием продуктивности реализации вычислений, основанным на соотношении «цена – производительность – качество». В основе решения лежит концепция однородных вычислительных систем, сформулированная в Институте математики СО РАН СССР под руководством Э.В. Евреина в 1962 году [1]. Подход актуален в условиях распространенности информационных сетей, объединяющих многочисленные ЭВМ. Описываемый в статье метод и программное обеспечение организации распределенных ВС были апробированы при организации РВС кафедры вычислительной техники НГТУ на базе существующей сети персональных компьютеров. Разработанное программное обеспечение для функционирования системы включает в себя библиотеку и сетевое приложение. Библиотека представляет собой инструментарий для реализации параллельных алгоритмов. Сетевое приложение предназначено для управления запуском параллельных алгоритмов одновременно на нескольких компьютерах: настройки системы, организации подсистем, распределения данных между вычислителями, вывода информации о состоянии ПК. В настоящее время ведется разработка модуля балансировки вычислительной нагрузки и модуля обеспечения продолжения счета при отказе одного или нескольких вычислителей. В статье приводятся полученные результаты численных экспериментов по оценке эффективности на тестовом примере решения системы линейных алгебраических уравнений. Оценены предельные значения коэффициента ускорения.

**Ключевые слова:** высокопродуктивные вычислительные системы, параллельный алгоритм, мелкозернистое распараллеливание, крупноблочное распараллеливание, эффективность, масштабируемость, системное программное обеспечение, система линейных алгебраических уравнений

DOI: 10.17212/1814-1196-2016-2-115-128

---

\* Статья получена 17 декабря 2015 г.

## ВВЕДЕНИЕ

Технологии параллельных вычислений в настоящее время получили широкое распространение в различных отраслях науки и промышленности. При этом вопросы эффективности той или иной технологии и систем, построенных с учетом энергетических, экономических и других критериев, все больше принимаются во внимание. В последнее время все больше ученых инициирует актуализацию задачи повышения качества функционирования высокопроизводительных систем. В дополнение к термину HPC (от англ. High Performance Computing – высокопроизводительные вычисления) появились такие понятия, как HPTC (от англ. High Performance Technical Computing – высокопроизводительные технические вычисления) и HPCS (High-Productivity Computing Systems – высокопродуктивные вычислительные системы).

Одна из современных тенденций развития высокопроизводительных вычислительных систем (ВВС) связана с проблемой оценки затрат вычислительных кластеров в условиях их коммерческого использования. Возникновение проблемы обусловлено отсутствием инструкций нормативного и методического обеспечения, которые были бы одинаково прозрачны с точки зрения бухгалтерского, управленческого учета и экономического анализа. Еще одна существующая проблема связана с ограниченностью загрузки ВВС, не отвечающей их потенциальным вычислительным мощностям. На практике последние часто используются лишь на несколько процентов, что приводит к неоправданно большим эксплуатационным расходам на содержание дорогостоящих вычислительных систем [6]. Уникальные и зачастую весьма дорогостоящие системы, созданные при значительной поддержке государства, оказываются недоступными для индивидуальных компаний, предприятий и др. В то же время существует масса малобюджетных организаций. Как правило, они располагают парком персональных компьютеров (ПК), используемых в повседневной работе сотрудников. Стоит заметить, что обычно эта техника работает в течение рабочего дня (8–10 часов), а в остальное время, включая выходные дни, она не используется. Совокупная же производительность этих ПК может быть сравнима с производительностью вычислительного кластера стоимостью в миллионы, десятки миллионов рублей [6]. Этот факт не мог остаться незамеченным и породил такое направление распределенных вычислений, как *Volunteer computing* – «добровольные вычисления». С каждым годом растет число участников проекта BOINC [7, 8], допускающего две возможные формы участия. Первая и основная заключается в предоставлении своих вычислительных средств для расчетов по выбранному волонтером проекту, а вторая – дает возможность создания собственного проекта. Наиболее популярным и востребованным среди программных средств разработки параллельных алгоритмов, ориентированных на использование ВС на основе персональных компьютеров, является интерфейс передачи сообщений MPI. Стандарт имеет почти 25-летнюю историю. За это время появилось большое количество бесплатных и коммерческих реализаций, был существенно расширен функционал. Однако, не стоит оставлять без внимания актуальность разработки отечественных программных продуктов.

Отмеченные факторы определяют выбор пути в направлении повышения продуктивности использования вычислительных ресурсов путем инте-

грации их в распределенные вычислительные системы с программируемой структурой (РВС ПС) [1–3]. Под последними подразумевается композиция множества элементарных машин (ЭМ) и сети межмашинных связей. Взаимодействие может осуществляться в локальной сети или сети Интернет по стеку ТСР/IP [9]. Структура вычислительной системы определяется на этапе организации подсистемы посредством программного обеспечения.

Сказанное в равной степени относится и к учебному процессу в вузах, где интеграция машин способствует внедрению современных методов обучения и повышению качества учебного процесса по ИТ-технологиям. Один из методов к организации РВС ПС, программные средства его реализации, а также результаты их апробации на кафедре вычислительной техники НГТУ представлены в настоящей работе.

## 1. ПОСТАНОВКА ЗАДАЧИ

Задача организации вычислительного процесса на РВС может быть охарактеризована различными признаками [1–5]. Имеется  $N$  элементарных машин, соединенных сетью связи. Необходимо организовать вычислительный процесс исходя из цели повышения продуктивности вычислительной системы. Предлагается критерий продуктивности вычислений, определяемый метрикой

$$K = R / C, \quad (1)$$

где  $R$  – количество требуемых результатов вычислений заданного качества  $Q$  в единицу времени  $T$  при себестоимости получения результатов  $C$ . Качество  $Q$  может оцениваться, в частности, точностью вычислений: например, точностью решения системы линейных уравнений. Под стоимостью подразумеваются эксплуатационные затраты, например, коэффициент эффективности использования электроэнергии (Power Usage Effectiveness – PUE). Рассчитывается как отношение всей потребленной объектом энергии к количеству, расходуемому на работу непосредственно вычислительных устройств. В идеальном случае коэффициент должен стремиться к единице. Однако, реально цифры могут отличаться в разы, поскольку помимо оборудования ИТ присутствует масса другого мощного оборудования, в первую очередь системы охлаждения и кондиционирования. На отвод тепла от активного оборудования приходится затрачивать дополнительную мощность, и значение PUE может составлять 3 (энергосберегающие мероприятия не проводятся, системы изначально неоптимально спроектированы), 2 (есть большие резервы для энергосбережения), 1.5 (резервы по-прежнему есть: на каждый киловатт мощности активного оборудования приходится 0.5 кВт энергии на охлаждение и сопутствующие цели). Интегрально продуктивность зависит от большого числа векторных и скалярных параметров:

$$K = F\{S, Q, R, C\}. \quad (2)$$

Для распараллеливания вычислений применяется крупноблочная методика, ориентированная на разбиение сложной задачи на крупные блоки-подзадачи, между которыми существует слабая связность. Такие подзадачи называют ветвями параллельного алгоритма (Р-алгоритма). Все ветви иден-

тичны, каждая реализуется своим вычислителем. В алгоритмах, построенных на основе такого подхода, операции обмена между подзадачами составляют незначительную часть по сравнению с общим числом операций в каждой подзадаче. Одним из подходов к крупноблочному распараллеливанию является распараллеливание по циклам. При этом подходе процесс решения задачи представляется в виде параллельных ветвей, полученных расщеплением цикла на части, число которых в пределе равно числу повторений цикла. На входе и выходе из цикла параллельный процесс сводится к последовательному процессу вычислений, доля которого в общем времени незначительна [14].

В качестве условий при оценивании продуктивности (2) предполагается, что на вход системы может поступить одна задача или набор задач  $I = \{I_i\}$ ,  $i = 1, \dots, L$ , где  $L$  – число задач в наборе. Для каждой задачи  $I_i$  требуется выделить множество ЭМ (подсистему ЭМ необходимого размера). Каждая из задач описывается вектором параметров  $p_i = \{p^1, p^2, \dots, p^q\}$ , где  $q$  – число параметров. Допускается, что в наборе присутствуют задачи с различным количеством параметров  $q$ . Каждой ветви параллельной программы выделяется не более одной ЭМ.

## 2. ПОДХОД К ОРГАНИЗАЦИИ ВЫСОКОПРОДУКТИВНЫХ РВС

Решение вычислительных задач с учетом условия (2) предполагает наличие сетевой вычислительной структуры, позволяющей реализовать требования *продуктивности, масштабируемости, параллелизма, крупноблочного распараллеливания, равноценности прав доступа пользователей* к вычислительной системе.

Этим требованиям отвечает взятая за основу авторами концепция РВС с программируемой структурой [1, 2]. Основные принципы ее организации.

- Распределенность (рассредоточенность) – допускает включение в систему территориально удаленных вычислителей.
- Децентрализованное управление и функциональная однородность – обеспечивают возможность управления вычислительным процессом, ввода исходных данных и вывода результатов с любого/на любой ПК вычислительной системы.
- Живучесть – достигается путем перераспределения функций вышедшего из строя вычислителя между остальными.
- Переменность (адаптивность) структуры – изменение структуры и связей между элементами системы с целью достижения наибольшей адекватности между системой и конкретной прикладной задачей.

По степени гранулярности (отношение объема вычислений к объему коммуникаций) различают следующие подходы к распараллеливанию вычислений: рассмотренное выше крупноблочное и мелкозернистое. Мелкозернистое распараллеливание ориентировано на разбиение алгоритма на предельно простые блоки и требует выделения для каждого этапа вычислений максимально возможного количества одновременно выполняемых блоков. Процесс такого распараллеливания весьма трудоемок, в сравнении с крупноблочным подходом требует частых обменов данных между блоками. Получаемые при

этом параллельные алгоритмы характеризуются структурной неоднородностью, разными объемами операций на различных этапах вычислений [2]. Рисунок 1 демонстрирует совмещение крупноблочной и мелкозернистой схем распараллеливания при организации параллельных вычислений на РВС ПС.

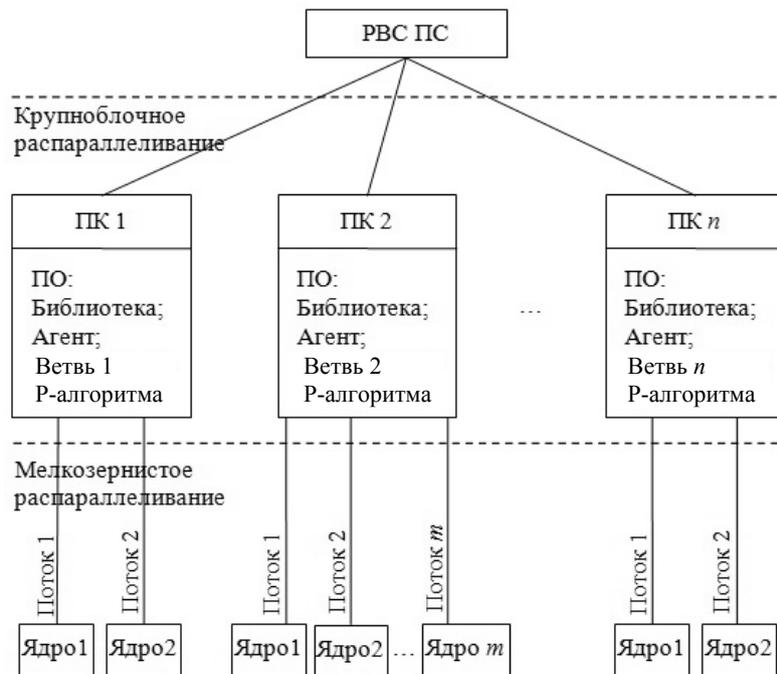


Рис. 1. Совмещение крупноблочной и мелкозернистой схем распараллеливания

Выделим два условных уровня: верхним будем считать уровень параллелизма в рамках всей РВС ПС, а нижним – уровень параллелизма внутри вычислителя. Исходная задача подлежит разбиению на крупные слабо связанные блоки-подзадачи (ветви), которые впоследствии распределяются между вычислителями. В такой форме реализуется крупноблочная схема распараллеливания, лежащая в основе вышеописанного подхода и работающая на верхнем уровне. Если рассматривать распараллеливание на нижнем уровне системы (внутри отдельного ПК между ядрами процессора), разбиение производится на предельно простые блоки – потоки. Таким образом, подход к распараллеливанию, применяемый на этом уровне системы, принимает вид мелкозернистого параллелизма. Указанные принципы организации программного обеспечения (ПО) положены в основу разрабатываемой авторами РВС ПС.

### 3. СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Разработанное программное обеспечение РВС ПС (ПО РВС ПС) включает в себя библиотеку и сетевое приложение. Библиотека представляет собой инструментарий для реализации параллельных алгоритмов. В качестве средства для организации РВС ПС используется платформа .NET, в частно-

сти, язык С#. Библиотека предоставляет разработчику параллельного алгоритма универсальный набор системных операций.

- *Обобщенный условный переход* – управляет счетом в зависимости от обобщенного признака, выработанного всеми ПК системы. Поэтому данная операция также является и точкой синхронизации кода, так как для получения результирующего признака необходимо, чтобы каждая машина выполнила функцию условного обобщенного перехода. Команда может использоваться для синхронизации вычислителей.

- *Обобщенный безусловный переход или прерывание* – приостанавливает действие параллельного алгоритма на ПК, участвующем в безусловном обобщенном переходе, и обеспечивает выполнение переданного кода.

- *Обмен* – передает данные в подсистему. Все передачи данных происходят асинхронно. С помощью данной функции можно обеспечить сбор результатов в подсистеме.

Межмашинные взаимодействия могут осуществляться по реализованным в библиотеке схемам: *трансляционной* (одна машина транслирует данные всем), *коллекторной* (все машины пересылают данные в одну), *конвейерной* (каждая – своим соседям), *дифференцированной* (любая – любой). Результатом разработки параллельного алгоритма с использованием библиотеки является исполняемый файл (\*.exe). Запуск его осуществляется под управлением операционной системы Windows.

Сетевое приложение *Агент* РВС ПС предназначено для управления запуском параллельных алгоритмов одновременно на нескольких компьютерах, а именно: настройки системы, организации подсистем, распределения данных между вычислителями, вывода информации о состоянии ПК. Более подробно процесс организации подсистемы состоит в следующем. Для развертывания системы требуется произвести установку «Agent.exe» на каждом из компьютеров. С этой целью достаточно скопировать данный файл и запустить его на выполнение. При первом запуске пользователь выбирает режим работы клиента: VPN или LAN-режим. В дальнейшем изменить режим пользователь может в настройках агента. LAN-режим не требует никаких дополнительных настроек (используются системные настройки). VPN-режим требует указать IP VPN-сервера, логин и пароль для подключения к сети. После настройки режима программа будет работать абсолютно одинаково в обоих режимах.

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

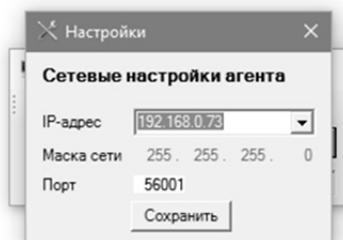


Рис. 2. Окно настроек

Следует различать две стороны взаимодействия: *Исполнитель* (по умолчанию) и *Инициатор* (по запросу пользователя). При запуске приложения «Агент» РВС ПС, машина, на которой он был запущен, автоматически становится исполнителем. С любого ПК может быть запущен процесс организации подсистемы. В этом случае ПК становится инициатором.

Каждый исполнитель выступает в роли ТСП-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 нескольких 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 одновременное участие в нескольких вычислениях негативно скажется на общей производительности исполнителя.

С использованием IP-адреса узла и маски высчитывается адрес сети, который будет использоваться инициатором для поиска исполнителей в автоматическом режиме (рис. 3).

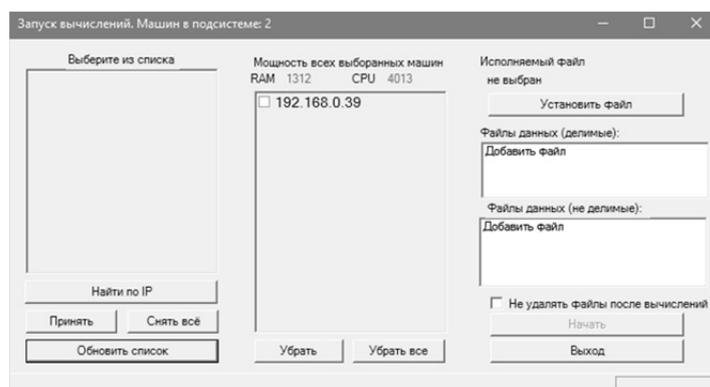


Рис. 3. Окно запуска вычислений

После завершения опроса доступных для вычислений ПК собирается файл `iplist.txt`. Список содержит номера, IP-адреса узлов, маски, порты всех машин в подсистеме. Изначально список IP-адресов доступен только для инициатора вычислений, а после распределения задачи большой размерности между узлами он доступен всем вычислителям. Пользователю предоставляется возможность выбора необходимого количества ПК для решения задачи.

Рассмотрим введенную *систему статусов*. Все ПК, на которых запущен *Агент*, имеют по умолчанию статусы «свободен». Как только эти машины зарезервированы для вычислений, они получают статус «занят» (что отражает требование к РВС: каждой ветви вычислений – один ПК). Если же машина отключилась, то она помечена статусом «отключен». Если же ПК перестал отвечать в процессе вычисления, то всем машинам в подсистеме отправляет пакет «error» с кодом 101, и включается режим «ожидание». Если машина вернулась в сеть, то всем клиентам отправляется пакет `continue` с кодом 101, иначе отправляется пакет `stop` и следом за ним пакет `update`, уведомляющий о том, что связь с ПК была потеряна. Для организации нового вычисления на любом из ПК можно открыть *Агент* РВС ПС и в нем организовать новое

вычисление. Программное обеспечение прошло процедуру государственной регистрации [16,17]. В настоящее время ведутся работы по разработке модуля балансировки вычислительной нагрузки и модуля обеспечения продолжения счета при отказе вычислителя. Для обеспечения отказоустойчивости распределенной вычислительной системы кафедры ВТ НГТУ выбран метод локальных контрольных точек (КТ). КТ каждого из задействованных в вычислениях узлов будут дублироваться в память соседних вычислителей. Соседние ПК будут определяться по номерам из списка IP-адресов (iplist.txt). При таком методе число максимально допустимых отказов растет, так как даже при отказе части узлов их вычисления можно будет продолжить на избыточных узлах. Нагрузка на сеть при этом будет зависеть от размера контрольных точек [10–13, 18].

С использованием ПО РВС ПС предприятия могут развернуть ВС на основе корпоративной сети, включающей сети филиалов, или организовать вычисления в локальной сети, что позволит не прибегать к услугам сторонних организаций и избежать передачи данных «на сторону». Также ПО может быть использовано как средство организации проектов «Добровольные вычисления».

#### 4. АПРОБАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

На кафедре вычислительной техники НГТУ организован фрагмент экспериментальной РВС ПС – научно-образовательная вычислительная система «Пара» (НОВС «Пара») [15]. 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 РВС ПС. Благодаря многозадачности операционной системы компьютеры, участвующие в вычислениях, могут использоваться и для других целей во время вычислений.

После запуска головной программы на любой из машин перед ее пользователем появится список машин, входящих в систему. Всем ПК, включенным в состав системы, в зависимости от степени их участия в процессе функционирования присваивается текущий статус. Система выводит информацию о быстродействии процессора и объеме оперативной памяти каждой машины. Имеется возможность выбрать необходимое количество свободных машин исходя из размерности задачи. После этого необходимо выбрать на диске программу (.exe) и файл (или несколько файлов) с данными. В процессе вычислений ПК будут взаимодействовать друг с другом самостоятельно согласно написанному параллельному алгоритму, а по окончании отправят сообщение о том, что работа завершена. Программа даст возможность вернуть компьютерам состояние «свободен», а также удалить ненужные данные. Отметим также, что система может работать как в режиме решения одной задачи большой размерности, так и в режиме решения набора или потока задач. Количество задач и их размерность ограничиваются лишь количеством ПК в системе/подсистеме и их ресурсами.

В качестве тестовой задачи для иллюстрации работы программной системы организации вычислений использован алгоритм решения системы линейных алгебраических уравнений (СЛАУ) методом наискорейшего спуска.

Краткое описание метода работы может быть представлено следующим образом.

1. Каждое уравнение  $n$  решается относительно  $X_n$ .
2. Задается начальное приближение  $x(0) = [b_1/a_{11}, \dots, b_i/a_{ii}, \dots, b_n/a_{nn}]$ .
3. Затем начинается итерационный процесс, в котором  $x$  выражается через  $x_{\text{пред.итер}}$  по формуле

$$x_n = \frac{b_n - (a_{n1}x_1 + a_{n2}x_2 + \dots + a_{n(n-1)}x_{n-1})}{a_{nn}}. \quad (4)$$

Оканчивается этот процесс при достижении заданной точности  $\text{eps}$ .

Применение крупноблочного метода в значительной степени упрощает процесс распараллеливания задачи: на первом этапе каждый из  $m$  компьютеров решает уравнение с частью данных  $N/m$ . На втором этапе аналогично задается начальное приближение. После каждой итерации ПК обмениваются полученными в результате вычислений данными.

Любой параллельный алгоритм (Р-алгоритм) представляет собой композицию связанных ветвей, где каждая ветвь – это последовательность как обычных операторов (арифметических и логических), так и операторов, реализующих взаимодействия (обмены информацией между ветвями).

Очевидно, что все взаимодействия между ветвями составляют накладные расходы, следовательно, чем меньше взаимодействий, тем выше эффективность реализации параллельного алгоритма. Приведем принятые показатели эффективности параллельного алгоритма. Коэффициентом накладных расходов называют

$$\varepsilon = \frac{t}{T}, \quad (5)$$

где  $t$  – время, затраченное ВС на организацию и реализацию обменов информацией между ветвями;  $T$  – время, необходимое на выполнение арифметических и логических операций при реализации Р-алгоритма.

*Коэффициент ускорения* для параллельного алгоритма вычисляется по формуле

$$\chi = \frac{\tau_1}{\tau_n}, \quad (6)$$

где  $\tau_1$  – время решения задачи на одном вычислителе;  $\tau_n$  – время выполнения соответствующего Р-алгоритма в системе из  $n$  вычислителей. Также используют *коэффициент эффективности* Р-алгоритма, представленный формулой

$$-E = \frac{\chi}{n}. \quad (7)$$

По результатам реализации вычислений было получено следующее. На решение задачи с размерностью матрицы  $10^3 \times 10^3$  элементов с  $\text{eps} = 10^{-7}$  на одном ПК затрачивается менее одной секунды. Для матрицы  $2.5 \cdot 10^4 \times 2.5 \cdot 10^4$  компьютеру с процессором Intel i5 (средний современный процессор) потребуется уже 16.5 минуты, а  $5 \cdot 10^4 \times 5 \cdot 10^4$  уже почти 10 часов. Такое значительное увеличение времени объясняется как раз «эффектом памяти», матрица  $2.5 \cdot 10^4 \times 2.5 \cdot 10^4$  элементов занимает в памяти около 6 Гб и полностью помещается в 8 Гб оперативной памяти тестовой машины, в то

время как матрица  $5 \cdot 10^4 \times 5 \cdot 10^4$  элементов занимает уже 25 Гб. Системе приходится создавать файл подкачки, куда отправляются лишние данные, и скорость решения значительно уменьшается.

Проанализируем результаты выполнения программы с использованием системы РВС ПС. Параллельная программа, решающая СЛАУ (матрица  $1.2 \cdot 10^4 \times 1.2 \cdot 10^4$  элементов, точность  $E = 10^{-7}$ ), была выполнена на компьютерах от одного до восьми. Использовались машины с процессорами Intel Core 2, имеющие 2 Гб оперативной памяти, работающие на 32-разрядной системе Windows. Каждая машина выводила чистое время вычисления в миллисекундах (без времени загрузки данных в память и ожидания подключения других машин). Результаты усреднялись и округлялись до секунд.

В табл. 1 показаны результаты выполнения алгоритма на системе из  $m \in [1; 8]$  компьютеров,  $N = 12\,000$ ,  $E = 10^{-7}$ .

Таблица 1

Результаты работы параллельного алгоритма

$N$ (машин)	1	2	3	4	5	6	7	8
$\tau_n$ , с	43	26	17	13	11	10	10,5	11
$\chi$ (коэффициент ускорения)		1,65	2,53	3,31	3,91	4,3	4,1	3,91
$E$ (коэф. эффект.)		0,83	0,84	0,83	0,78	0,72	0,59	0,49
$\varepsilon$ (коэффициент накладных расходов)		0,21	0,19	0,21	0,28	0,40	0,71	1,05

Рисунок 4 демонстрирует время работы программы в зависимости от количества машин, на которых выполнялось вычисление.

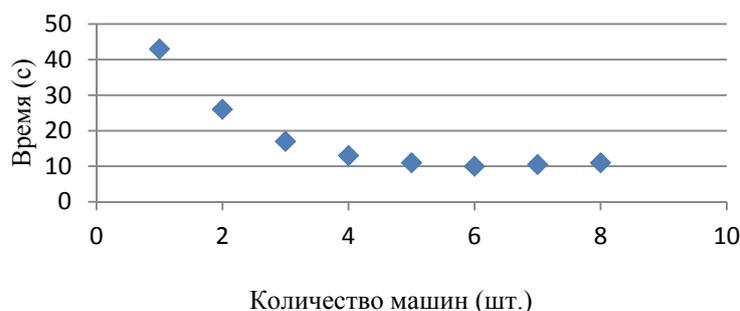


Рис. 4. Время работы программы в зависимости от количества машин

Как видно, наименьшее время решения достигается при реализации вычислений на шести машинах. В дальнейшем время начинает постепенно увеличиваться, что объясняется увеличением накладных расходов (время, которое система тратит на приемы, передачи, синхронизацию и т. д.). Рисунок 5 демонстрирует изменение коэффициента ускорения. Данный коэффициент демонстрирует достигнутый выигрыш по времени по сравнению с одиночным вычислителем. В данной ситуации достигнут коэффициент 4.3 на шести вычислителях. Распределенная вычислительная система позволяет добиться ускорения решения алгоритма по сравнению со временем решения последо-

вательного алгоритма. При наличии достаточного количества компьютеров можно масштабировать систему под любой объем данных, а следовательно, работать только с быстрой памятью.

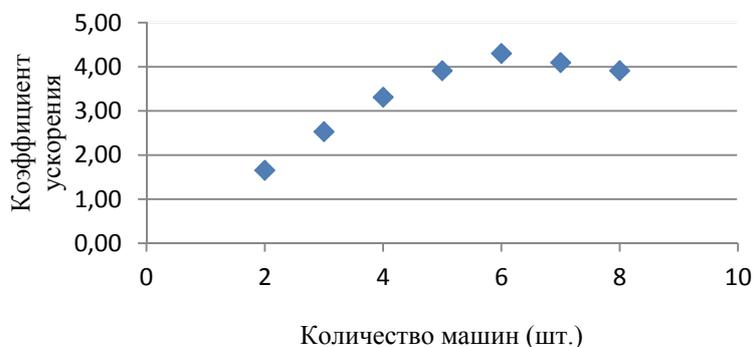


Рис. 5. Зависимость коэффициента ускорения от количества машин

## ЗАКЛЮЧЕНИЕ

В интересах учебного процесса на кафедре ВТ НГТУ разработан фрагмент распределенной вычислительной системы с программируемой структурой (РВС ПС). Данная статья отражает разработанное оригинальное программное обеспечение для функционирования системы, включающее в себя библиотеку и сетевое приложение. Библиотека представляет собой инструментарий для реализации параллельных алгоритмов. Сетевое приложение предназначено для управления запуском параллельных алгоритмов одновременно на нескольких компьютерах, а именно настройки системы, организации подсистем, распределения данных между вычислителями, вывода информации о состоянии ПК.

Получены результаты численных экспериментов по оценке эффективности на тестовом примере решения системы линейных алгебраических уравнений, представленных матрицей размера  $1.2 \cdot 10^4 \times 1.2 \cdot 10^4$  элементов при точности решения  $E = 10^{-7}$ . Оценены предельные значения коэффициента ускорения.

Описанный метод и программное обеспечение отличаются доступностью для предприятий. Организация РВС ПС может базироваться на парке имеющихся ПК, подключенных к локальной или глобальной сети. Программное обеспечение сопровождается подробными руководствами, набором алгоритмов, методическими указаниями по организации обучения. Данная разработка может использоваться для решения прикладных задач, в научных и образовательных целях.

## СПИСОК ЛИТЕРАТУРЫ

1. Евреинов Э.В. Однородные вычислительные системы, структуры и среды. – М.: Радио и связь, 1981. – 208 с.
2. Хорошевский В.Г. Архитектура вычислительных систем. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. – 520 с.

3. *Хорошевский В.Г.* Распределенные вычислительные системы с программируемой структурой // Вестник СибГУТИ. – 2010. – № 2 (10). – С. 3–41.
4. *Перышкова Е.Н., Мамойленко С.Н., Ефимов А.В.* Инструментарий решения масштабируемых задач на распределенных вычислительных системах // Восьмая Сибирская конференция по параллельным и высокопроизводительным вычислениям, Томск, 28–30 октября 2015 г. – Томск, 2015. – С. 143–150.
5. Алгоритмы отказоустойчивого управления ресурсами пространственно-распределенных вычислительных систем / А.Ю. Поляков, О.В. Молдованова, А.А. Пазников, М.Г. Курносов, С.Н. Мамойленко, А.В. Ефимов // Вестник СибГУТИ. – 2014. – № 4. – С. 11–29.
6. *Воеводин В.В., Воеводин Вл. В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2004. – 608 с.
7. *Durech J., Hanus J., Vanko R.* Asteroids@home – A BOINC distributed computing project for asteroid shape reconstruction // Astronomy and Computing. – 2015. – Vol. 13, N 1. – P. 80–84.
8. Asteroids@home – Customizable execution environments for evolutionary computation using BOINC + virtualization / F. Fernández De Vega, G. Olague, L. Trujillo, D. Lombrana González // Natural Computing. – 2013. – Vol. 12, N 2. – P. 163–177.
9. *Kurose J.F., Ross K.W.* Computer networking: a top-down approach. – Boston: Addison Wesley, 2009. – 864 p.
10. *Tanenbaum A.S., Steen M.* Distributed systems: principles and paradigms. – New Jersey: Prentice Hall PTR, 2002. – 803 p.
11. *Jalote P.* Fault tolerance in distributed systems. – New Jersey: Prentice Hall, 1994. – 448 p.
12. *Tel G.* Introduction to distributed algorithms. – Cambridge: Cambridge University Press, 2000. – 596 p.
13. *Vaidya N.H.* A case for two-level distributed recovery schemes // Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, Ottawa, Canada, 15–19 May 1995. – New York: ACM, 1995. – P. 64–73.
14. Considerations of implementation and application of high-performance computing for low-budget enterprises / V.K. Mishchenko, P.V. Mishchenko, V.V. Gubarev, M.A. Karneev // Proceedings XII international conference Actual Problems of Electronic Instrument Engineering, APEIE–2014, Novosibirsk, Russia, 2–4 October 2014: in 7 vol. – Novosibirsk, 2014. – Vol. 1, pt. 2. – P. 11–13.
15. *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.
16. Библиотека разработки параллельных алгоритмов для распределенных вычислительных систем с программируемой структурой (Библиотека РВС ПС): свидетельство о государственной регистрации программы для ЭВМ № 2015618556 / П.В. Мищенко, К.А. Бородин, М.А. Карнеев, В.К. Мищенко. – Заявка № 2015615598; заявл. 25.06.15; опубл. 20.09.15, Бюл. № 9. – 1 с.
17. Сетевое приложение для организации распределенных вычислительных систем с программируемой структурой (РВС ПС Агент): свидетельство о государственной регистрации программы для ЭВМ № 2015618557 / К.А. Бородин, П.В. Мищенко, М.А. Карнеев, В.К. Мищенко. – Заявка № 2015615602; заявл. 25.06.15; опубл. 20.09.15, Бюл. № 9. – 1 с.
18. *Бондаренко А.А., Якобовский М.В.* Обеспечение отказоустойчивости высокопроизводительных вычислений с помощью локальных контрольных точек // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. – 2014. – № 3. – С. 20–36.

*Мищенко Полина Валерьевна*, аспирант, старший преподаватель кафедры вычислительной техники факультета автоматики и вычислительной техники Новосибирского государственного технического университета. Основные направления научных исследований: информационные сети, параллельные вычисления. Имеет более 20 публикаций. E-mail: mpv-777@ya.ru

*Беляшов Андрей Владимирович*, кандидат геолого-минералогических наук, заведующий отделением Института геофизических исследований, г. Курчатов, Республика Казахстан. Основное направление научных исследований – алгоритмы и программы обработки и интерпретации геофизических данных. Имеет 36 публикаций. E-mail: abelyashov@igr.kz

## ***Program organization of high-performance distributed computing systems with a programmable structure \****

P.V. MISHCHENKO<sup>1</sup>, A.V. BELYASHOV<sup>2</sup>

<sup>1</sup>Novosibirsk State Technical University, 20, K. Marx Prospekt, Novosibirsk, 630073, Russian Federation. E-mail: mpv-777@ya.ru

<sup>2</sup>Institute of Geophysical Research, Republic of Kazakhstan, Kurchatov city, Meridian site, 071100, EKO. E-mail: abelyashov@nnc.kz

A pressing problem of developing domestic methods and tools for organizing distributed computing system (DCS) functioning for low-budget enterprises is considered. Special attention is paid to a system approach to program tool development. It is characterized by the integration of DCS organization program functioning with add-in modules of functionality expanding and sets of parallel algorithms. DCS functioning is correlated with the criterion of computing performance based on the price - productivity – quality relationship. This solution is based on the concept of homogeneous computing systems, which was suggested in the Institute of Mathematics of the USSR SB RAS under the leadership of E.V. Evreinov in 1962 [1]. This approach is quite timely under current conditions when information networks comprising numerous computers are widely spread. The method described in this paper and the software for the organization of distributed computing systems was tested during the DCS arrangement based on the existing network of personal computers at the NSTU Computer Science Department. The developed software for system functioning includes a library and a network application. The library is a tool for parallel algorithm realization. The network application is aimed at running parallel algorithms simultaneously on several computers. Its functions include setting up the system, organizing sub-systems, distributing data between computers, and displaying information about the PC status. Currently a module for balancing the computation load and a module for providing computing in the case of failure of one or several computers are being developed. The results of computing experiments to evaluate their efficiency based on the test case of solving a system of linear equations (SLE) are given in this article. Limiting values of speedup coefficients are estimated.

**Keywords:** high-performance computing systems, parallel algorithm, fine-grain parallelism, coarse-grain parallelism, distributed computing systems, efficiency, scalability, system software

DOI: 10.17212/1814-1196-2016-2-115-128

## **REFERENCES**

1. Evreinov E.V. *Odnorodnye vychislitel'nye sistemy, struktury i sredy* [Homogeneous computer systems, structures and environment]. Moscow, Radio i svyaz' Publ., 1981. 208 p.
2. Khoroshevskii V.G. *Arkhitektura vychislitel'nykh sistem* [Architecture of computer system]. Moscow, Bauman MSTU Publ., 2008. 520 p.
3. Khoroshevskii V.G. Raspredelemnnye vychislitel'nye sistemy s programmiruemoi strukturoi [Distributed computing systems with programmable structure]. *Vestnik SibGUTI*, 2010, no. 2 (10), pp. 3–41.
4. Peryshkova E.N., Mamoilenko S.N., Efimov A.V. [Tools of the solution of scalable tasks on the distributed computing systems]. *Vos'maya Sibirskaya konferentsiya po parallel'nym i vysokoproduktivnym vychisleniyam* [The Eighth Siberian Conference on Parallel and High-performance Computing], Tomsk, 28–30 October 2015, pp. 143–150.
5. Polyakov A.Yu., Moldovanova O.V., Paznikov A.A., Kurnosov M.G., Mamoilenko S.N., Efimov A.V. Algoritmy otkazoustoichivogo upravleniya resursami prostranstvenno-raspredelemnnykh vychislitel'nykh sistem [Algorithms of fault tolerance resource management of the distributed computing systems]. *Vestnik SibGUTI*, 2014, no. 4, pp. 11–29.

---

\* Received 17 December 2015.

6. Voevodin V.V., Voevodin V.I.V. *Parallel'nye vychisleniya* [Parallel computing]. St. Petersburg, BHV-Petersburg Publ., 2004. 608 p.
7. Durech J., Hanus J., Vanco R. Asteroids@home – A BOINC distributed computing project for asteroid shape reconstruction. *Astronomy and Computing*, 2015, vol. 13, no. 1, pp. 80–84.
8. Fernández De Vega F., Olague G., Trujillo L., Lombraña González D. Asteroids@home – Customizable execution environments for evolutionary computation using BOINC + virtualization. *Natural Computing*, 2013, vol. 12, no. 2, pp. 163–177.
9. Kurose J.F., Ross K.W. *Computer networking: a top-down approach*. Boston, Addison Wesley, 2009. 864 p.
10. Tanenbaum A.S., Steen M. *Distributed systems: principles and paradigms*. New Jersey, Prentice Hall PTR, 2002. 803 p.
11. Jalote P. *Fault tolerance in distributed systems*. New Jersey, Prentice Hall, 1994. 448 p.
12. Tel G. *Introduction to distributed algorithms*. Cambridge, Cambridge University Press, 2000. 596 p.
13. Vaidya N.H. A case for two-level distributed recovery schemes. *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, Ottawa, Canada, 15–19 May 1995, pp. 64–73.
14. Mishchenko V.K., Mishchenko P.V., Gubarev V.V., Karneev M.A. Considerations of implementation and application of high-performance computing for low-budget enterprises. *Proceedings XII international conference Actual Problems of Electronic Instrument Engineering, APEIE–2014*, Novosibirsk, Russia, 2–4 October 2014, vol. 1, pt. 2, pp. 11–13.
15. Mishchenko P.V., Gubarev V.V., Mishchenko V.K. [Distributed computing system in Computer engineering department of NSTU]. *Mnogoyadernye protsessory, parallel'noe programmirovaniye, PLIS, sistemy obrabotki signalov: sbornik statei Vserossiiskoi nauchno-prakticheskoi konferentsii* [Multicore processors, parallel programming, PLD and the signal processing systems: a collection of articles All-Russian scientific-practical conference], Barnaul, 27 February 2015, pp. 150–155. (In Russian)
16. Mishchenko P.V., Borodin K.A., Karneev M.A., Mishchenko V.K. *Biblioteka razrabotki parallel'nykh algoritmov dlya raspredelennykh vychislitel'nykh sistem s programmiruemoi strukturoi (Biblioteka RVS PS)* [Library of development of parallel algorithms for the distributed computing systems with programmable structure (RVS PS Library)]. The Certificate on official registration of the computer program. No. 2015618556, 2015.
17. Borodin K.A., Mishchenko P.V., Karneev M.A., Mishchenko V.K. *Setevoe prilozhenie dlya organizatsii raspredelennykh vychislitel'nykh sistem s programmiruemoi strukturoi (RVS PS Agent)* [Network application for the organization of the distributed computing systems with programmable structure (DCS PS the Agent)]. The Certificate on official registration of the computer program. No. 2015618557, 2015.
18. Bondarenko A.A., Yakobovskii M.V. Obespechenie otkazoustoichivosti vysokoproizvoditel'nykh vychislenii s pomoshch'yu lokal'nykh kontrol'nykh toчек [Fault tolerance for hpc by using local checkpoints]. *Vestnik YuUrGU. Seriya: Vychislitel'naya matematika i informatika – Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*, 2014, no. 3, pp. 20–36.