

УДК 004.457

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РАСПРЕДЕЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ В КОНВЕРГЕНТНОЙ СЕТИ

Клепиков А.К.

*ФГБОУ ТГПУ им Л.Н. Толстого, Тула, Россия (301267, пос. Октябрьский, Киреевский район, Тульская область, д. 65, кв. 3), e-mail: don-klepikov@yandex.ru*

Обозначена актуальность использования облачных технологий в информационно-вычислительной среде предприятий, в том числе предприятий финансовой сферы. Предложен способ интеграции облачных технологий в сети предприятий с использованием монитора-распределителя данных с целью образования конвергентных вычислительных сетей. Рассмотрен монитор-распределитель данных в конвергентных вычислительных сетях предприятий, с уклоном в специфику деятельности предприятий финансовой сферы. Показана структура модулей программного распределителя данных. Указан основной алгоритм работы монитора-распределителя. Рассмотрены основные технологии, с помощью которых разработан монитор-распределитель. Рассмотрен процесс получения статистических данных по распределению информационных объектов для предоставления пользователю в визуальном виде. Рассмотрены вспомогательные средства, использованные для построения аналитических схем и диаграмм. Сделаны выводы по использованию монитора-распределителя данных в конвергентных вычислительных сетях.

Ключевые слова: облачные технологии, конвергентные вычислительные сети, распределитель данных, информационно-вычислительная среда, предприятия финансовой сферы, алгоритм распределения данных в конвергентной вычислительной сети.

## SOFTWARE DISTRIBUTION COMPUTATIONAL PROBLEMS IN A CONVERGED NETWORK

Клепиков А.К.

*FSEI TSPU L.N. Tolstoy, Tula, Russia (301267, pos. The Octyabriskiy, Kireyevsky district, Tula region, d. 65, q. 3), e-mail: don-klepikov@yandex.ru*

Denotes the relevance of the use of cloud technologies in information and enterprise computing environment, including the financial sector enterprises. We propose a way to integrate cloud computing in the enterprise network using the monitor, the data distributor to form converged networks. Considered monitor Distribution of data in computer networks converged companies, with a bias in the specifics of the financial sector companies. Shows the structure of software modules data distributor. Set the basic algorithm of the monitor-distributor. The basic technology, which is designed to monitor Splitter. The process of obtaining statistical data on the distribution of information objects to provide the user with a visual form. Considered aids used for the construction of analytical charts and diagrams. The conclusions on the use of the monitor, the data distributor in converged computing networks.

Keywords: cloud computing, converged computing network distributor data, information and computing environment, enterprise finance, distribution algorithm converged data in a computer network.

В настоящее время информационно-вычислительные сети (ИВС) предприятий финансовой сферы (ПФС) активно развиваются в направлениях автоматизации процессов обработки данных. Все большее количество населения использует онлайн-сервисы, предоставляемые ПФС. Онлайн-сервисы являются частью автоматизированных банковских систем (АБС), которые на данный момент являются важнейшим звеном в процессе автоматизации информационной структуры банка.

Активное использование ИВС приводит к множественным процессам обработки больших объемов данных, хранение и обработка части которых, с точки зрения временных и

экономических затрат, неэффективна. В связи с чем, новой тенденцией в развитии ИВС ПФС является использование облачных провайдеров, с целью включения в состав ИВС вычислительных облачных серверов. Образовывая такие сети, ПФС создают конвергентные вычислительные сети (КВС) [6].

При организации КВС важным звеном выступает программное обеспечение (ПО), которое способствует объединению сетей ПФС виртуальных облачных сетей в одну общую сетевую структуру с определяемыми политиками доступа, задаваемыми администратором сети. Другой важнейшей задачей является распределение вычислительных задач в виде информационных объектов (ИО) между серверами КВС: серверами ВС ПФС и облачными виртуальными серверами [2, 4].

Для распределения ИО, с целью выполнения информационно-вычислительных работ, ИВР предлагается использовать алгоритм рационального распределения вычислительных задач, который позволяет производить распределение поступившей вычислительной задачи  $X_k$ ,  $k=\{1,2,3,\dots,K\}$ , поступающей от АРМ  $f_g$ ,  $g=\{1,2,3,\dots,G\}$  на один из вычислительных серверов множества серверов ИВС ПФС  $L=\{l_1, l_2, l_3,\dots, l_n\}$  и множества облачных вычислительных серверов  $Cl=\{cl_1, cl_2, cl_3,\dots, cl_m\}$ .  $SRV=\{L,Cl\}$  – образует КВС ПФС,  $j=\{l_1, l_2, l_3,\dots, l_n, cl_1, cl_2, cl_3,\dots, cl_m\}$  – общее число серверов КВС ПФС,  $t$  – определяет интервал времени опроса серверных станций.  $P$  – шлюзовой сервер ВС ПФС,  $Z_{x_k}$  – набор пакетов данных принадлежащих задачи (ИВР). Алгоритм состоит из следующих шагов:

**Шаг 1.** Получить задачу  $X_k$  от АРМ  $f_g$ , в виде набора пакетов данных  $Z_{x_k}$  по задаче  $X_k$ .

**Шаг 2.** Определить объемы вычислительных ресурсов, необходимых для выполнения поступившей задачи  $W_{x_k} = \{c_{x_k}, m_{x_k}, l_{x_k}, h_{x_k}, s_{x_k}\}$ .

**Шаг 3.** Из множества серверов  $SRV=\{L,Cl\}$  определить сервера, для которых вычислительная мощность процессора, объем ОЗУ, скорости передачи данных по сети, объем свободного пространства на ПЗУ удовлетворяют требованиям, необходимым для решения задачи. Для каждого  $SRV_j$  проверяем условия  $c_{srv_j} > c_{x_k}$ ,  $m_{srv_j} > m_{x_k}$ ,  $l_{srv_j} > l_{x_k}$ ,  $h_{srv_j} > h_{x_k}$ ,  $s_{srv_j} > s_{x_k}$ , то добавляем новый сервер в множество доступных для решения задачи серверов  $A\{\}$ , если множество  $A$  пусто, то переход к шагу 4, иначе к шагу 5.

**Шаг 4.** Ожидаем интервал времени  $t$ . Переход к шагу 3.

**Шаг 5.** Выбираем сервер  $a_{best}$  из множества  $A$ , для которого прогнозируемая временная оценка времени исполнения ИВР минимальна.

**Шаг 6.** Получение ip адреса сервера  $a_{best}$ .

**Шаг 7.** Изменение ip адреса получателя в  $Z_{x_k}$  на ip адрес сервера  $a_{best}$ .

**Шаг 8.** Изменение ip адреса отправителя в  $Z_{x_k}$  на адрес шлюзового компьютера  $P$ .

**Шаг 9.** Отправка ИВР  $Z_{x_k}$  на сервер  $a_{max}$ .

**Шаг 10.** Добавление в журнал ведения логов информации об отправленной задаче: наименование задачи, компьютер отправителя, компьютер получателя.

**Шаг 11.** Ожидание поступления ответных данных по задаче  $Z_{x_k}$  от сервера  $a_{best}$ .

**Шаг 12.** Если данные от сервера  $a_{best}$  поступили, то выполняется переход к шагу 13, иначе переход к шагу 11.

**Шаг 13.** Изменение в  $Z_{x_k}$  ip адреса отправителя на ip адрес  $P$ .

**Шаг 14.** Изменение в  $Z_{x_k}$  ip адреса получателя на адрес  $f_g$ .

**Шаг 15.** Отправить полученные данные на  $f_g$ .

**Шаг 16.** Переход к шагу 1.

На основании вышеизложенной модели был разработан монитор-распределитель данных в конвергентных вычислительных сетях TiCloudBalancer, который разрабатывался с использованием ОС Microsoft Windows 8, в среде MS Visual Studio 2013 с использованием языков программирования C++ и C#. Выбор языка программирования C++ был обусловлен следующими критериями:

- скорость работы системных программ написанных с помощью языка C++;
- возможность работать напрямую с потоками данных, передаваемыми по сети с помощью сокетов;
- возможность работать напрямую с участками памяти компьютера;
- наличие большого количества библиотек, позволяющих расширить возможности языка без снижения общей скорости работы программы;
- возможность низкоуровневой работы с передаваемыми по технологии TCP/IP данными.

Использование языка C# в проекте было обусловлено необходимостью применения визуальной среды для предоставления администратору доступа к меню управления и мониторинга работы системы распределения ИВР. Также в проекты, написанные с использованием языка C#, могут быть внедрены модули и библиотеки, написанные с использованием языка C++, что позволяет совместить скорость работы программы с большими возможностями по визуализации происходящих в распределителе процессов с применением современной среды разработки MS Visual Studio 2013. Среда MS Visual Studio

2013 позволяет разрабатывать приложения, используя языки C, C++ и C#, используя современные средства построения визуального интерфейса windows presentation foundation (WPF). Перечисленные выше позволяют создавать системное ПО, производить его рефакторинг, дорабатывать, компилировать, используя различные версии встроенных библиотек, что является основополагающим при выборе среды разработки для создания монитора-распределителя ИВР.

Монитор-распределитель представляет собой системное ПО, построенное с помощью модулей: главный (шлюзовой) модуль, серверный модуль, клиентский (АРМ) модуль [1].

Монитор-распределитель позволяет производить распределение нагрузки только для приложений, построенных на основе клиент-серверной архитектуры [3] (рис. 1), где возможно разделение серверной части, выполняющей роль обработчика запросов, и клиентской части, исполняющей роль приложения, позволяющего пользователю взаимодействовать с системой по средствам графического интерфейса [5].



*Рис. 1. Структура ПО распределителя нагрузки TiCloudBalancer*

Монитор-распределитель TiCloudBalancer реализован на основании проекта с открытым исходным кодом Balancer в среде разработки Visual Studio 2013 с использованием языка программирования C#. Приложение построено с использованием объектно-ориентированного подхода. Каждый модуль представляет собой законченный программный продукт, поставляемый в виде скомпилированной программы и устанавливаемый с помощью стандартного установщика InstallShieldWizard.

После установки, монитора-распределитель использует порт 5747 для передачи данных по сети между модулями, установленными на серверных станциях, шлюзовом компьютере и АРМ, и становится доступным как служба Windows Cloud Balancer ARM Module для АРМ станций, Cloud Balancer Server Module – для серверных станций, Cloud Balancer Gate Module – для шлюзового компьютера.

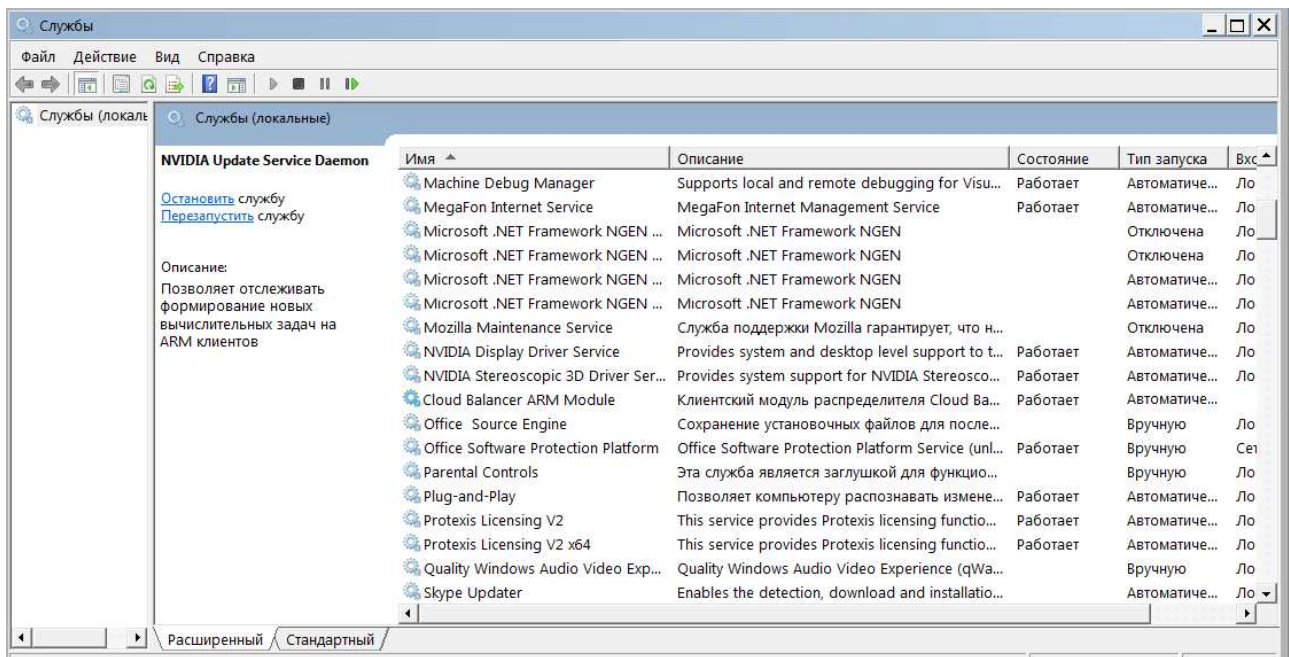


Рис. 2. АРМ модуль в списке процессов клиентского компьютера сети ПФС

Монитор-распределитель позволяет выводить статистику по проведенным распределениям ИО по вычислительным серверам КВС ПФС, а также отображать ход распределения в режиме реального времени. Подобные функциональные особенности доступны в связи с использованием языка С# и технологии WPF, с помощью которой происходит вывод оконного интерфейса с целью получения информации о работе системы (рис. 3).

В процессе работы распределителя формируются массивы данных, в которых хранится вся информация по поступившим, обработанным заявкам, времени обработки заявок, а также информация о заявках, которые на данный момент находятся в стадии обработки.

Информация по уже завершенным процессам обработки данных хранится в лог-файлах шлюзового модуля на шлюзовом компьютере, для доступа к лог-файлам используется метод:

```
public void ReadLogFile(string[] logData1)
{
    this.logData = logData1;
}
```

Листинг 1. Метод получения данных для формирования аналитического представления данных о происходящих процессах обработки данных в КВС с использованием монитора-распределителя.

Для визуализации и представления аналитической информации использовались элементы управления для построения диаграмм Microsoft Chart Controls, доступные для технологии WPF и NET FRAMEWORK 4.0 и выше [7]. Для использования библиотеки

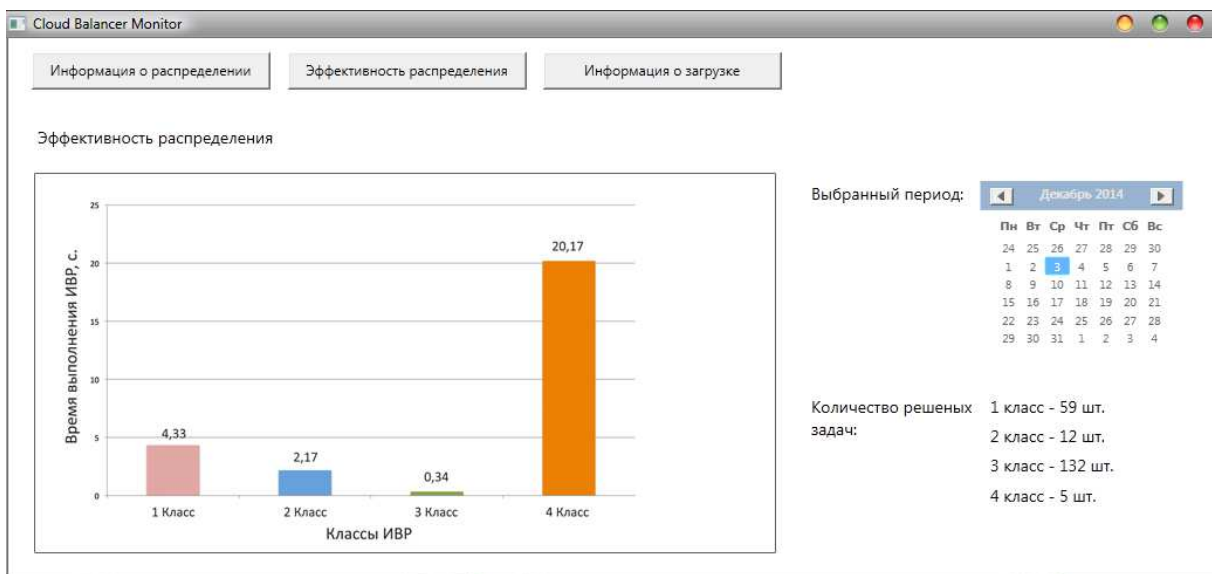
отображения графиков был использован метод ReadLogs(), основные особенности которого изложены в листинге 2. Сущность метода ReadLogs() заключается в чтении и обработке сформированных строк, представляющих собой информацию о происходящих в мониторе-распределителе событиях.

```
using (StreamReader sr = new StreamReader(path))
{
    string line;
    int y = 0;
    System.DateTime x;

    chart2.Series["Series1"].XValueType = ChartValueType.DateTime;
    chart2.ChartAreas[0].AxisX.LabelStyle.Format="yyyy-MM-dd HH:mm:ss";
    chart2.ChartAreas[0].AxisX.Title = "Дата, время";
    chart2.ChartAreas[0].AxisY.Title = "Количество обработанных задач";

    while (!sr.EndOfStream)
    {
        line = sr.ReadLine();
        string[] strArr = line.Split('\t');
        x = DateTime.Parse(strArr[0]);
        y = Convert.ToInt32(strArr[1]);
        chart2.Series["Series1"].Points.AddXY(x.ToOADate(), y);
    }
    sr.Close();
}
```

Листинг 2. Метод получения данных для формирования аналитического представления данных о происходящих процессах обработки данных в КВС с использованием монитора-распределителя.



*Рис. 3. Вывод аналитической информации в мониторе-распределителе данных*

Для отображения информации о задачах, которые на данный момент проходят этап выполнения, использовался метод ReadActiveData(), который позволяет получить данные из массивов, проводимых ИВР в КВС.

**Выводы:**

Монитор-распределитель данных в конвергентных вычислительных сетях позволяет производить рациональное распределение ИО по вычислительным серверам ИВС с подключенной облачной инфраструктурой. Основные возможности разработанного ПО состоят в распределении вычислительных задач, также существуют и дополнительные функциональные особенности, которые позволяют производить анализ происходящих в КВС процессов с помощью визуального интерфейса, содержащего вывод необходимой информации о состоянии работы монитора-распределителя.

Разработанный алгоритм позволяет производить рациональное распределение ИО по параметрам времени выполнения ИВР и стоимости ИВР на вычислительном сервере, что способствует повышению эффективности функционирования КВС ПФС в целом.

### Список литературы

1. Клепиков, А.К. Программный распределитель вычислительных задач с функцией мониторинга / А. К. Клепиков, А.Н. Привалов // Известия ТулГУ. Технические науки. – Тула, 2013. – № 1. – С. 274-280.
2. Кульба В.В., Ковалевский С.С., Косяченко С.А., Сиротюк В.О. Теоретические основы проектирования оптимальных структур распределённых баз данных. [Текст] / В.В. Кульба, С.С. Ковалевский, С.А. Косяченко, В.О. Сиротюк. – М.: Синтег, 1999. – 660 с.
3. Ладыженский, Г.М. Технология «клиент-сервер» и мониторы транзакций / Г.М. Ладыженский // «Открытые системы». – Москва, 1994. – № 3. – С. 4-10.
4. Привалов, А.Н. Моделирование информационных процессов тренажерных систем [Текст] / Е.В. Ларкин, А.Н. Привалов. – LAP LAMBERT Academic Publishing, 2012. – 240 с.
5. Романченко, В. Распределенные облачные (Cloud) вычисления. [Электронный ресурс] //URL: <https://sites.google.com/site/moiknigiilekcii/lekcii/infonnatika/lekcia-no25/cloud> (дата обращения: 10.01.2015).
6. HP, HP помогает Альфа-банку перевести сеть хранения на 16-гигабитный Fibre Channel [Электронный ресурс] // URL: <http://habrahabr.ru/company/hp/blog/142297/> (дата обращения: 10.01.2015).

7. Microsoft, Using Chart Controls [Электронный ресурс] // URL: [http://msdn.microsoft.com/en-us/library/dd489238\(v=vs.140\).aspx](http://msdn.microsoft.com/en-us/library/dd489238(v=vs.140).aspx) (дата обращения 10.01.2015).

**Рецензенты:**

Привалов А.Н., д.т.н., профессор кафедры информатики и информационных технологий, ФГБОУ ТГПУ им. Л.Н. Толстого, г. Тула;

Добровольский Н.М., д.ф.-м.н., профессор кафедры алгебры, математического анализа и геометрии, г. Тула.