

В.А. СТАРЫХ, Д.А. ВАРЁНОВ, С.Ю. ПЛОСКОВ, А.Ю. КУЗНЕЦОВ  
V.A. STARYKH, D.A. VARENOV, S.Y. PLOSKOV, A.Y. KUZNETSOV

ПОСТРОЕНИЕ И ИССЛЕДОВАНИЕ РАСПРЕДЕЛЁННОЙ  
ИНФОРМАЦИОННОЙ СИСТЕМЫ ВЕБ-ПРИЛОЖЕНИЙ И СЕРВИСОВ НА  
БАЗЕ КЛАСТЕРНОЙ АРХИТЕКТУРЫ.

CONSTRUCTION AND RESEARCH OF THE DISTRIBUTED INFORMATION  
SYSTEM OF WEB APPLICATIONS AND SERVICES BASED ON CLUSTER'S  
ARCHITECTURE.

Аннотация

Данная статья посвящена исследованию подходов к решению проблемы неэффективного использования вычислительных ресурсов аппаратно-программных комплексов при использовании веб-приложений и сервисов. Представлены результаты исследования распределенных вычислительных систем, а также возможных вариантов реализации распределенной вычислительной системы веб-приложений и сервисов на базе кластерной архитектуры. Полученные результаты могут быть полезны для разработчиков управляющих приложений и сервисов на базе свободного ПО, а также администраторов, выполняющих настройку оборудования и программного обеспечения, в задачах балансировки нагрузки кластерных решений. Статья подготовлена на основе результатов работ, выполняемых авторами по условиям государственного контракта в рамках реализации мероприятия 1.2.2. ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы.

**Ключевые слова:** *распределённая вычислительная система, аппаратно-программный комплекс, веб-приложение, сервис, кластер, архитектура, управление, балансировка, оптимизация, производительность.*

Annotation

This article is devoted to research of approaches to solving the problem of irrational and inefficient computational resources usage of hardware-software complex with the use of WEB-applications and services. In this article are given the results of research of distributed calculating systems as well as available approaches to construction of distributed calculating system of WEB-applications and services based on cluster's architecture. The received results can be useful to developers of operating applications and services based on the open source software, and also for the managers who are carrying out adjustment of the equipment and the software, in problems of balancing for clusters. Article is prepared on the basis of results of the works which are carried out by authors on conditions of the state contract within the conditions of realisation part 1.2.2. of the Federal Program «Scientific and scientific and pedagogical shots of innovative Russia» on 2009 – 2013.

## 1. Введение

К сегодняшнему дню многие ВУЗы по всей стране столкнулись с проблемой нерационального и неэффективного использования имеющихся вычислительных ресурсов аппаратно-программных комплексов. На данный момент типовой аппаратно-программный комплекс образовательного учреждения (АПК ОУ) представляет собой как функционально, так и структурно сложную распределенную вычислительную систему. Под распределенной системой в рамках данной работы понимается набор независимых серверов, представляющий их пользователям единой системой обработки, объединенной средствами коммутации.

Несмотря на то, что вариантов использования АПК ОУ существует множество, анализ тенденций развития информационных систем образовательных учреждений показывает, что наиболее частым является использование, как в учебном процессе, так и для решения научных и административно-хозяйственных задач образовательного учреждения веб-приложений и сервисов [1-4]. Результаты исследования прикладного программного обеспечения уровня веб-приложений, используемого в зарубежных учебных заведениях, проведенного британским Центром изучения технологий обучения и развития [5], показали постоянный рост популярности веб-приложений и сервисов. Результаты данного исследования можно принять как типовые по отношению и к российским учебным заведениям. Круг задач, для решения которых используется данное прикладное ПО, также постоянно расширяется. Наиболее популярные веб-приложения и сервисы, используемые в различных учебных и научных заведениях, представлены в таблице 1.

Таблица 1 – Веб-приложения и сервисы, используемые в высших учебных и научных заведениях

Наименование	Описание
Twitter	Разработан компанией Twitter inc. Сервис позволяет пользователям отправлять короткие текстовые заметки (до 140 символов), используя веб-интерфейс, SMS, службы мгновенных сообщений или сторонние программы-клиенты.
Delicious	Разработан компанией Yahoo! Inc. Сервис предоставляет зарегистрированным пользователям услуги по хранению и публикации закладок на страницы сети Интернет. Пользователи Delicious могут просматривать имеющиеся закладки, упорядочивая их по популярности и присваиваемым меткам (ключевым словам).
SlideShare	Разработан компанией SlideShare inc. Сервис предназначен для распространения и публикации презентационных материалов, тексто-графических документов и видеозаписей.
Moodle	Разработан Martin Dougiamas. Представляет собой прикладное программное обеспечение, предназначенное для создания динамических учебных сред – система управления обучением (LMS). Данный продукт распространяется бесплатно, по лицензии GNU GPL. Moodle ориентирован прежде всего на организацию взаимодействия между преподавателем и учениками, хотя подходит и для организации традиционных дистанционных курсов, а также поддержки очного обучения.
WikiSpaces	Разработан компанией Talgient LLC. Сервис для публикации документов и совместной работы над тексто-графической документацией. Предоставляет зарегистрированному пользователю (или группе пользователей) персональную Wiki страницу для публикаций.
Illuminate	Разработан компанией Illuminate Inc. Приложение ориентировано на проведение образовательных веб-конференций и семинаров с

	возможностью трансляции аудиоконтента и видеоизображений посредством сети Интернет.
Lectora	Разработан компанией Trivantis Corporation. Динамическая учебная среда (LMS), ориентированная на разработку учебных курсов, лекционного и справочного материала, а также на организацию взаимодействия между преподавателем и учениками.
TeacherTube	Разработан компанией TeacherTube LLC. Сервис обеспечивает условия для размещения, поиска и получения видеoinформации образовательной направленности в режиме свободного обмена.
BaseCamp	Разработан компанией 37signals. Сервис позволяет организовать управление проектами и другую совместную работу сотрудников.

Популярность таких приложений объясняется тем, что хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети, что дает возможность обучающимся получать доступ к учебным материалам и выполнять практические задания удаленно на любом компьютере имеющем доступ в сеть Интернет. Сотрудники образовательного учреждения могут работать с документами и взаимодействовать между собой без привязки к рабочему месту.

В основном веб-приложения в образовании используются для создания электронных библиотек учебных заведений, разнообразных хранилищ электронных образовательных ресурсов, систем тестирования и удаленного выполнения лабораторных и практических работ, сред виртуализации и дистанционного обучения, а также систем совместной работы с электронными документами и научно-образовательными ресурсами, функционирующими в составе открытых информационно-образовательных сред различного масштаба и назначения. [6-8]

Дальнейший рост популярности Интернет-взаимодействия преподавателей, сотрудников и студентов вуза неизбежно приведет к нехватке вычислительных ресурсов АПК ОУ. Приобретение дополнительных вычислительных мощностей сопряжено с ростом затрат на их эксплуатацию, что может стать значительной дополнительной нагрузкой для бюджета учебного заведения. Исходя из этого, рост производительности АПК ОУ должен происходить, в основном, не за счет увеличения количества оборудования, а за счет его более эффективного использования.

На данный момент не существует готовых решений (моделей, алгоритмов), гарантирующих требуемую производительность веб-приложений при минимальном количестве задействованных вычислительных ресурсов. Как правило, каждое решение – уникальный проект, требующий больших финансовых вложений. Эффективность использования веб-приложений и всего АПК ОУ в целом зависит от многих факторов, но решающим, на наш взгляд, являются архитектура, модель построения и алгоритм функционирования распределенной вычислительной системы. Под алгоритмом функционирования распределенной вычислительной системы в данной работе понимается формализованное описание порядка совместной работы совокупности распределенной операционной системы (ОС) и виртуальных ОС.

Современные тенденции развития распределенных вычислительных систем предполагают использование механизма кластеризации в качестве базовой архитектуры. Механизм кластеризации в свою очередь предполагает взаимодействие компонентов распределенной вычислительной системы на уровне виртуализации операционной системы.

Таким образом, данная работа посвящена разработке *типовой модели* и вариантов ее оптимизации - распределенной вычислительной системы основанной на кластерной архитектуре для развертывания веб-приложений и сервисов обработки научно-образовательных ресурсов. *Типовая модель* позволит образовательным и научным учреждениям сократить затраты на создание и внедрение различных информационных систем на базе имеющегося оборудования за счет создания высокопроизводительной

распределенной вычислительной системы.

## 2. Исследование подходов к построению распределенной вычислительной системы веб-приложений и сервисов обработки научно-образовательных ресурсов

Существует множество возможных вариантов реализации распределенных систем (к примеру, кластеры, мейнфреймы, GRID-системы), но далеко не все из них отвечают требованиям, предъявляемым к вычислительной системе обработки научно-образовательных ресурсов [6-9], на основе веб-технологий:

- увеличение производительности веб-приложений и сервисов за счет рационального использования вычислительных ресурсов АПК ОУ;
- обеспечение отказоустойчивости и доступности приложений в режиме 24/7;
- осуществление централизованного резервирования данных и сервисов;
- гибкое управление вычислительными ресурсами при изменении нагрузки на приложения;
- увеличение производительности АПК ОУ без изменения архитектуры информационных систем организации;
- снижение трудозатрат и стоимости эксплуатации АПК ОУ;
- осуществление миграции веб-приложений, сервисов и данных между различными вычислительными платформами.

Для анализа характеристик различных кластерных архитектур была разработана их классификация, а также исследованы возможные варианты реализации распределенных вычислительных систем на базе кластерной архитектуры.

По функциональному признаку кластеры можно разделить на "Высокоскоростные", "Системы Высокой Готовности", а также "Смешанные Системы".

Экспериментальное исследование, проведенное авторами данной статьи в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы показало, что кластеры смешанного типа являются оптимальными для создания распределенной вычислительной системы обработки научно-образовательных ресурсов, поскольку они обеспечивают максимальную эффективность использования ресурсов АПК ОУ, а также полностью соответствуют всем предъявляемым требованиям. Исследование проводилось с использованием аппаратного обеспечения представленного в таблице 2 и объединенного с помощью гигабитной сети Ethernet.

Таблица 2 – Спецификация аппаратного обеспечения, использованного при проведении экспериментального исследования.

Количество	Наименование сервера	Характеристики
2	IBM System p 550Q	1024 MB DDR-2 SDRAM 73.4 SCSI Disk Drive Assembly IDE Slimline DVD-ROM Drive 2 процессора 1.65 GHz POWER5
2	IBM System p 520Q	1024 MB DDR-2 SDRAM 73.4 GB SCSI Disk Drive Assembly IDE Slimline DVD-ROM Drive 2 процессора 1.65 GHz POWER5
1	Консоль управления аппаратным обеспечением. (Hardware Management Console, HMC)	HMC 1:7310-CR3 Rack-mounted Hardw.Mgmt.Console T117 TFT 17-inch Color Display

Смешанные кластеры – это кластерные системы, принцип действия которых строится на распределении запросов на обслуживание через один или несколько входных узлов

(балансировщиков нагрузки), которые перенаправляют их на обработку на различные вычислительные узлы, исходя из уровня их загрузки. В таких системах достигается высокая надежность без ущерба для производительности. С точки зрения клиента вся система представляется как единый экземпляр программы (веб-приложения). Вычислительные узлы, входящие в состав данной системы работают идентично одиночным серверам, но в дополнении к ним обеспечивают балансировку нагрузки и передачу управления при сбое. [16-17]

Кластеры этого класса совмещают в себе многие черты высокопроизводительных кластеров и кластеров высокой готовности, но, в то же время, лишены их основных недостатков. В отличие от высокопроизводительных систем, предполагающих вычислительный характер решаемых задач смешанные системы обеспечивают высокую скорость операций ввода-вывода, что является критическим показателем для скорости работы веб-приложений и сервисов. Скорость операций ввода-вывода определяет максимальное количество одновременно работающих пользователей и время реакции веб-приложения на запрос пользователя. Схема стандартного пользовательского запроса на обслуживание представлена на рисунке 1.

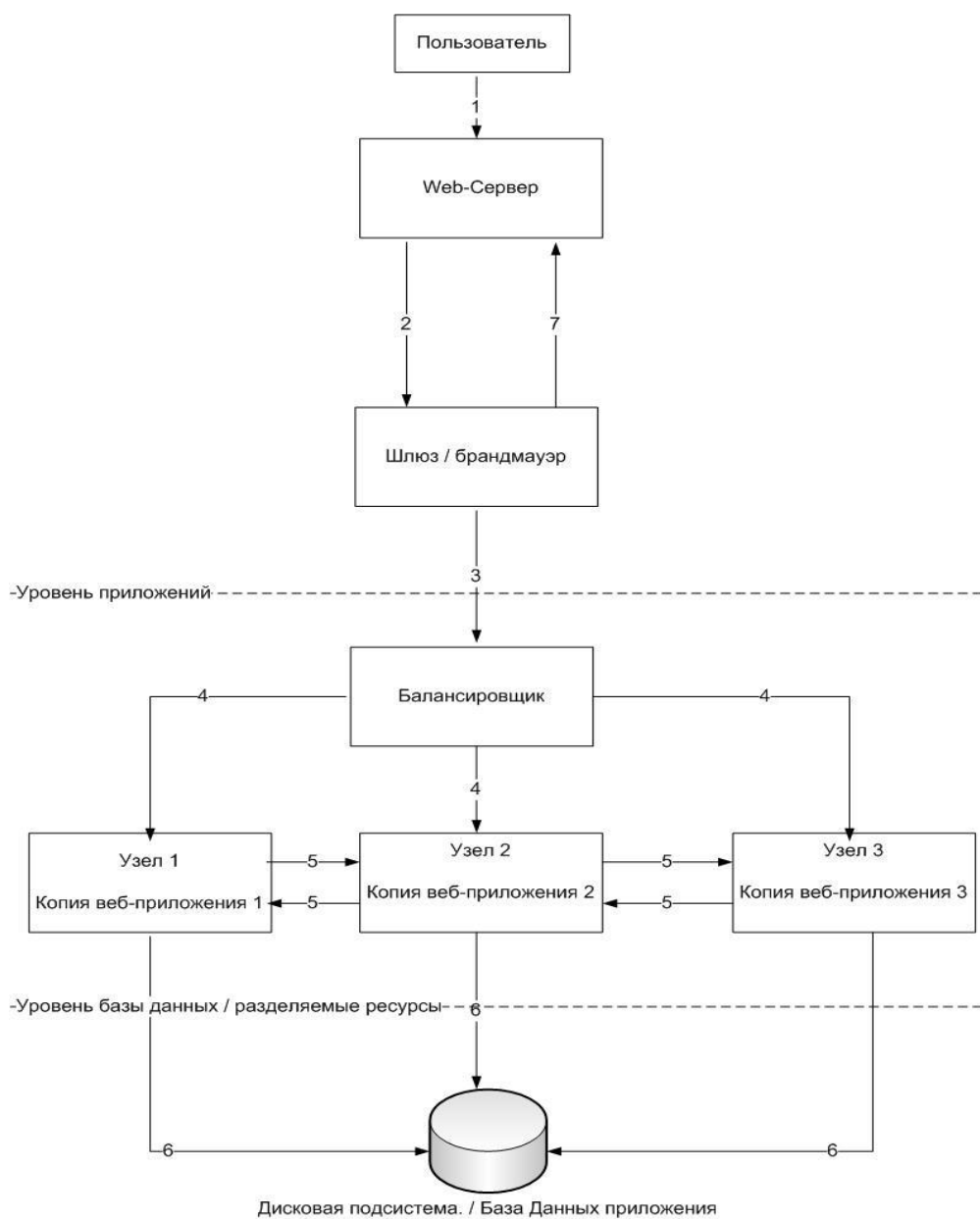


Рисунок 1 – схема обработки пользовательского запроса на обслуживание  
Цифрами на рисунке отмечены следующие итерации:

- 1 – Веб-браузер посылает пользовательский запрос на веб-сервер.
- 2 – Веб-сервер передает запрос на шлюз.
- 3 – Шлюз передает запрос на балансировщик нагрузки.
- 4 – Балансировщик нагрузки определив, какой именно из узлов распределенной системы будет заниматься его обработкой, передает его на соответствующий узел.
- 5 – Передача служебных сообщений о загрузке узлов системы. При необходимости обработка запроса пользователя может быть проведена на другом, менее загруженном узле системы.
- 6 – Запрос веб-приложения к базе данных для выборки необходимых данных для обработки запроса.
- 7 – Доставка обработанного запроса.

Большая часть узлов кластера высокой готовности находится в режиме ожидания, на случай выхода из строя одного из активных в данный момент серверов. Их использование для решения других задач невозможно, поэтому эффективность АПК на базе кластеров данного типа невысока. Кластеры смешанного типа лишены данного недостатка. [18-19]

Ниже на рисунке 2 представлена архитектура кластера смешанного типа.

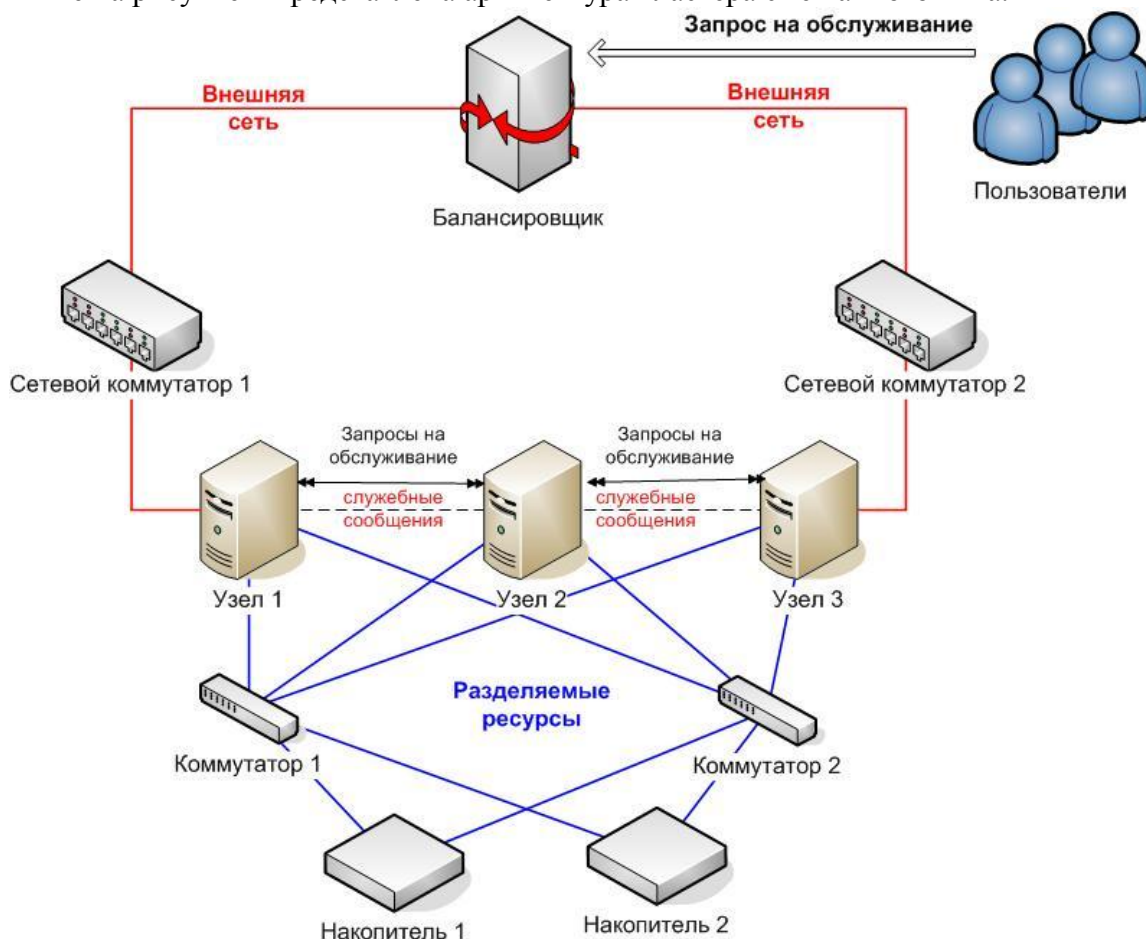


Рисунок 2 – Архитектура кластера смешанного типа

Рассматриваемая типовая архитектура смешанного кластера состоит из трех узлов (серверов), взаимодействующих с разделяемыми ресурсами. В качестве разделяемых ресурсов выступают накопители, представляющие собой дисковые массивы. На одном из узлов устанавливают балансировщик нагрузки, который управляет перераспределением нагрузки между узлами кластера. Управление кластером осуществляется с помощью служебных сообщений, содержащих информацию о состоянии узла.

В данной архитектуре принципиальным является отказоустойчивость канала связи, достигаемая за счет резервирования каналов связи и коммуникационного оборудования (на рисунке коммутатор 2 и сетевой коммутатор 2 дублируют основные).

Кластерная архитектура данного типа обладает следующими преимуществами:

- возможность равномерной или определяемой правилами балансировки запросов между узлами кластера;
- высокая отказоустойчивость за счет использования нескольких узлов и резервных каналов;
- простое увеличение производительности кластера за счет увеличения количества узлов;
- возможность обслуживания и замены узлов кластера без остановки работы приложения.

### **3. Исследование вариантов реализации распределенных вычислительных систем на базе кластерной архитектуры смешанного типа**

Варианты реализации распределенных вычислительных систем на базе кластерной архитектуры смешанного типа могут быть различными, в зависимости от комбинации и типов используемых компонент кластера. Рассмотрим зависимость характеристик кластера смешанного типа от вариантов его реализации.

**Типы узлов кластера.** Как и всякие распределенные системы, кластеры смешанного типа разделяются на однородные (все узлы кластера имеют одинаковую архитектуру и производительность) и гетерогенные (используются узлы различной архитектуры и производительности). Обычно, когда говорят о вычислительных кластерах, подразумевают однородные кластеры. Однако часто при наращивании кластера приходится использовать серверы, различающиеся не только по производительности, но и по архитектуре. Поэтому постепенно однородный вычислительный кластер может стать неоднородным. Эта неоднородность создает следующие проблемы:

- различие в производительности усложняет задачу распределения работ между процессорами;
- различие в архитектуре требует подготовки разных выполняемых файлов для разных узлов, а в случае различий в представлении данных, может потребовать и преобразования их форматов при передаче сообщений между узлами;

**Использование оперативной памяти.** По варианту использования кластером оперативной памяти они разделяются на кластеры с общей (разделяемой) памятью, с распределенной памятью (UMA-системы) и кластеры с физически распределенной, а логически общедоступной памятью (гибридные системы, NUMA-системы). Кластер с общей памятью обычно имеет высокую пропускную способность памяти при передаче информации между процессорами, но при условии, что не происходит одновременного обращения нескольких процессоров к одному и тому же элементу памяти. Кластеры с такой памятью носят название вычислительных систем с одинаковым временем доступа к памяти или сильносвязанными.

Для кластеров с распределенной памятью характерно наличие большого числа быстрых каналов, которые связывают отдельные части этой памяти с отдельными процессорами. Обмен информацией между частями распределенной памяти осуществляется относительно медленно. Кластеры с распределенной памятью называются также слабосвязанными.

Кластеры с гибридной памятью имеют оперативную память, которая физически распределена по различным частям системы, но логически разделяема (образует единое адресное пространство). Такая память называется еще логически общей (разделяемой) памятью (logically shared memory). В отличие от UMA-систем, в NUMA-системах время доступа к различным частям оперативной памяти различно.

**Тип коммуникационной среды.** По количеству уровней иерархии коммуникационной среды различают кластеры с одноуровневой коммутационной сетью (один уровень коммутации) и кластеры с иерархической коммутационной сетью. В кластерах с иерархической коммутационной сетью группы процессоров объединены с

помощью коммутации одного типа, а внутри каждой группы используется другая система коммутации.

**Архитектуры с разделяемыми дисками и без разделения ресурсов.** Кластер без распределения ресурсов не использует общей системы хранения данных. При таком подходе каждый узел имеет свои дисковые накопители, которые не используются совместно узлами кластерной системы. Фактически, на аппаратном уровне разделяются только коммуникационные каналы.

Кластеры с разделяемыми ресурсами состоит из общей системы хранения данных и узлов кластера, которые распределяют доступ к общим данным. При высокой мощности системы хранения данных, при работе с задачами, ориентированными на их обработку, архитектура с общими дисками является более эффективной. В этом случае не нужно держать несколько копий данных и в то же время, при выходе из строя узла, задачи могут быть мгновенно доступны для других узлов. [18-20]

**Тип масштабирования системы.** Рост вычислительной мощности можно обеспечить за счет применения горизонтального масштабирования кластера (наращивание мощностей распределенной кластерной системы за счет добавления новых узлов.) и вертикального (возможность наращивания вычислительной мощности системы за счет модернизации существующих узлов кластерной системы) либо осуществлять масштабирование, комбинируя оба этих варианта. [21]

**Оптимизация работы кластера.** Анализ функциональных характеристик смешанного кластера позволил сформулировать зависимости производительности веб-приложений и сервисов обработки научно-образовательных ресурсов от варианта реализации кластера, для чего разработаны математические модели для анализа работы алгоритма балансировки кластера, и выполнен анализ и оптимизация его работы в условиях стационарного и стохастического потока запросов.

Для вычисления величины интегрального критерия качества системы используется формула [22]:

$$S(N, \lambda_1, \lambda_2 \dots \lambda_N, b_1, b_2 \dots b_N) = \sum_{i=1}^N \left\{ \frac{a_i \lambda_i^2 b_i^2}{2(1 - \lambda_i b_i)} + d_i(b_i) \cdot (1 - \lambda_i b_i) \right\}, \quad (1)$$

где  $a_i$  - штраф за простой запросом в очереди к серверу  $i$  в течение единицы времени,  $d_i$  - штраф за простой сервера  $i$  в течение единицы времени,  $N$  - количество серверов,  $\lambda_i$  - интенсивность потока запросов на сервер  $i$ ,  $b_i$  - среднее время обслуживания запроса (производительность сервера).

Распределение нагрузки на сервера задается с помощью системы коэффициентов  $(\alpha_1, \alpha_2, \dots, \alpha_N)$  таких, что  $1 \geq \alpha_i \geq 0$ , ( $i = 1, 2, \dots, N$ ) и  $\sum_{i=1}^N \alpha_i = 1$ . Таким образом,

интенсивность потока запросов на сервер номер  $i$  равна  $\lambda_i = \alpha_i \Lambda$ , где  $\Lambda$  - интенсивность общего потока запросов на кластер.

Для заданного интегрального критерия качества и алгоритма распределения нагрузки, используется следующая формула:

$$S(\alpha_1, \alpha_2 \dots \alpha_N, b_1, b_2 \dots b_N) = \sum_{i=1}^N \left\{ \frac{a_i \alpha_i^2 \Lambda^2 b_i^2}{2(1 - \alpha_i \Lambda b_i)} + \frac{\delta_i}{b_i} (1 - \alpha_i \Lambda b_i) \right\}, \quad (2)$$

где  $d(b) = \frac{\delta}{b}$ ,  $b \in (0; \frac{1}{\Lambda})$ .

В общем случае простои серверов нежелательны, тогда при отсутствии запросов возможно выполнение фоновых задач, что определяет производительность сервера,



связанную с обработкой запросов (среднее время обработки запроса -  $b_i$  для сервера номер  $i$ ).

В связи с этим требуется решать задачу не только оптимального распределения потока на кластер, но и определения оптимальной производительности серверов, которая заключается в нахождении минимума:

$$\min_{\alpha_i, b_i} \{S(\alpha_1, \alpha_2 \dots \alpha_N, b_1, b_2 \dots b_N) = \sum_{i=1}^N \left\{ \frac{a_i \alpha_i^2 \Lambda^2 b_i^2}{2(1 - \alpha_i \Lambda b_i)} + \frac{\delta_i}{b_i} (1 - \alpha_i \Lambda b_i) \right\}\} \quad (3)$$

при ограничениях  $\sum_{i=1}^N \alpha_i = 1$ ,  $\alpha_i \geq 0$  и  $b \in (0; \frac{1}{\Lambda})$ .

Для решения задачи при детерминированном потоке запросов использовался метод неопределенных множителей Лагранжа, применение которого сводится к решению системы уравнений:

$$\left\{ \begin{array}{l} -a_1 \alpha_1 \Lambda^2 b_1^2 (\alpha_1 \Lambda b_1 - 2) / (2(\alpha_1 \Lambda b_1 - 1)^2) - \delta_1 \cdot \Lambda_1 + x = 0 \\ -a_2 \alpha_2 \Lambda^2 b_2^2 (\alpha_2 \Lambda b_2 - 2) / (2(\alpha_2 \Lambda b_2 - 1)^2) - \delta_1 \cdot \Lambda_1 + x = 0 \\ \dots \dots \dots \\ -a_n \alpha_n \Lambda^2 b_n^2 (\alpha_n \Lambda b_n - 2) / 2(\alpha_n \Lambda b_n - 1)^2 - \delta_n \cdot \Lambda + x = 0 \\ \sum_{i=1}^k \alpha_i - 1 = 0 \\ -a_1 \alpha_1^2 \Lambda^2 b_1 (\alpha_1 \Lambda b_1 - 2) / (2(\alpha_1 \Lambda b_1 - 1)^2) + \delta_1 (1 + \alpha_1 \Lambda) / b_1^2 = 0 \\ -a_2 \alpha_2^2 \Lambda^2 b_2 (\alpha_2 \Lambda b_2 - 2) / (2(\alpha_2 \Lambda b_2 - 1)^2) + \delta_2 (1 + \alpha_2 \Lambda) / b_2^2 = 0 \\ \dots \dots \dots \\ -a_n \alpha_n^2 \Lambda^2 b_n (\alpha_n \Lambda b_n - 2) / 2(\alpha_n \Lambda b_n - 1)^2 + \delta_n (1 + \alpha_n \Lambda) / b_n^2 = 0 \end{array} \right. \quad (4)$$

Поскольку реальные задачи имеют большую размерность, в качестве упрощения искомое решение задачи разбивается на два этапа. На первом этапе вычисляется оптимальное распределение входного потока между серверами  $\alpha_i, i = \overline{1, N}$ , а на втором этапе вычисляется оптимальная величина производительности для каждого сервера  $b_i, i = \overline{1, N}$ . Рассмотрены различные варианты применения предложенного подхода.

В реальных системах не всегда известно значение интенсивности входного потока запросов. Поэтому величина входного потока запросов рассматривается как случайная величина. В этом случае случайной величиной является и критерий качества работы системы, для оптимизации которого использован алгоритм управления, на основе метода Кифера-Вольфовица.

Показано, что увеличение штрафа за простой в очереди к серверу ведет к оттоку запросов от этого сервера, а увеличение штрафа за простой самого сервера ведет, наоборот, к притоку запросов на этот сервер. Также установлено, что увеличение производительности сервера (уменьшение времени обработки запроса на сервере) ведет к увеличению доли запросов на этот сервер по двум причинам: во-первых, возрастает штраф за простой сервера и это требует увеличения потока запросов на этот сервер для того, чтобы уменьшить вероятность простоя сервера; во-вторых, уменьшается время простоя в очереди к этому серверу, а значит и затраты, связанные с простоем.

Для вычисления оптимальной производительности сервера при заданном потоке запросов на сервер получено следующее соотношение:

$$\begin{aligned}
b_{n+1} = b_n - \frac{n^{-3/4}}{10} & \left[ \frac{a\Lambda^2 (b_n + 0,01 \cdot n^{-1/4})^2}{2(1 - \Lambda(b_n + 0,01 \cdot n^{-1/4}))} + \right. \\
& \frac{\delta}{(b_n + 0,01 \cdot n^{-1/4})} \cdot (1 - \Lambda(b_n + 0,01 \cdot n^{-1/4})) - \frac{a\Lambda^2 (b_n - 0,01 \cdot n^{-1/4})^2}{2(1 - \Lambda(b_n - 0,01 \cdot n^{-1/4}))} - \\
& \left. - \frac{\delta}{(b_n - 0,01 \cdot n^{-1/4})} \cdot (1 - \Lambda(b_n - 0,01 \cdot n^{-1/4})) \right]
\end{aligned} \tag{5}$$

Исследование процесса сходимости с использованием полученного соотношения выявило, что в связи с необходимостью выполнения условия существования установившегося режима для применения заданного критерия качества работы алгоритма, имеет место эффект подвижных границ, обусловленный тем, что от величины интенсивности входного потока зависит область допустимых значений производительности (среднего времени обработки запросов). Это может приводить к тому, что рассчитанное на предыдущем шаге значение производительности сервера, используемое для расчета величины производительности на текущем шаге, может оказаться за границами области допустимых значений. В связи с этим предложено и исследовано несколько подходов:

- масштабирование величины производительности сервера на текущем шаге, при котором используется следующее значение величины производительности сервера -  $b_i' = (\Lambda_i / \Lambda_{i+1}) b_i$ ; (6)

- усреднение (сглаживание) интенсивности входного потока, когда величина интенсивности входного потока измеряется несколько ( $K$ ) раз и при расчетах используется среднее

значение величины интенсивности входного потока -  $\Lambda_i = \frac{1}{K} \sum_{j=1}^K \Lambda_{i,j}$ . (7)

Также при исследовании влияния значений весовых коэффициентов на сходимость алгоритма, получено, что увеличение штрафа за простой сервера ведет к увеличению оптимального времени на обработку запроса для того чтобы обеспечить увеличение очереди к серверу и снизить вероятность его простоя. Увеличение штрафа за ожидание в очереди к серверу снижает влияние затрат на простой сервера и ведет к уменьшению оптимального времени на обработку запроса для того, чтобы уменьшить затраты, связанные с простоем в очереди к серверу. Влияние затрат, связанных с простоем сервера, при этом уменьшается.

Исследовано также влияние отношения штрафа за простой сервера, к штрафу за ожидание запросом в очереди к серверу на оптимальную производительность сервера, при известной величине интенсивности входного потока. Полученные результаты могут быть полезны для разработчиков управляющих приложений и сервисов на базе свободного ПО, а также администраторов, выполняющих настройку оборудования и программного обеспечения, в задачах балансировки нагрузки кластерных решений.

### Список литературы

1. Д.А. Варенов, А.А. Больных. Исследование подходов к разработке распределенной вычислительной системы на базе кластерной архитектуры для функционирования веб-приложений и сервисов обработки научно-образовательных ресурсов. //Труды XVII Всероссийской научно-методической конференции «Телематика'2010». Том 2. – СПб.: «Университетские телекоммуникации», 2010. – С. 378-379. – ISBN 978-5-7577-0354-1.
2. Варенов Д.А., Кузнецов А.Ю. Использование свободного программного обеспечения в составе аппаратно-программного комплекса Федерального центра информационно-образовательных ресурсов. //Седьмая конференция разработчиков свободных программ. Тезисы докладов. – М.: Институт Логики, 2010. – С. 12-14.
3. Варенов Д.А., «Использование репозитория свободного программного обеспечения в

- подготовке IT-специалистов.» Труды XV Всероссийской научно-методической конференции «Телематика 2008» Том 2. Спб.: «Университетские телекоммуникации» 2008, стр. 356-358.
4. Варенов Д.А., «Информационно-технологическое обеспечение проектирования прикладного программного обеспечения на основе открытых исходных текстов в рамках проекта OpenPower.» Инновации в экономике и социальной сфере. Тезисы докладов Всероссийской научно-практической конференции. - Н.Новгород, НГТУ им. Р.Е.Алексеева, 2008. - с. 229 – 230.
  5. Top 100 tools for learning [Электронный ресурс] — Электрон. текстовые дан.: Center for Learning & Performance Technologies, 2010. - Режим доступа: <http://www.c4lpt.co.uk/recommended/top100-2010.html>.
  6. Старых В.А., Открытая территориально-распределённая система управления информационными ресурсами. // "Информационные системы и технологии" ("ИСИТ"), № 6(62), 2010, 90-99 с.
  7. Варенов Д.А., «Разработка системы на базе архитектуры POWER по поддержке программистов и проектов на основе открытых исходных текстов в рамках проекта OpenPower». Сборник тезисов докладов III конференции «Свободное программное обеспечение в высшей школе». Москва, ALT Linux, 2008. - с. 65 — 67.
  8. Варенов Д.А., «Построение системы параллельных вычислений для научно-исследовательских и образовательных организаций а также малого и среднего бизнеса.» Материалы второй научно-технической конференции студентов, аспирантов и молодых специалистов Российского государственного университета инновационных технологий и предпринимательства. Москва, Российский государственный университет инновационных технологий и предпринимательства, 2006. с. 14 — 15.
  9. Эффективные кластерные решения [Электронный ресурс] / Владимир Савяк. – Электрон. текстовые дан. – М.: Ixbt.com, 2002. – Режим доступа: <http://www.ixbt.com/cpu/clustering.shtml>.
  10. Tom Dunn. Navy DoD Supercomputing Resource Center // HPCinsights. – 2009. – Spring 2009. – С. 19.
  11. Абламейко С.В., Абрамов С.М. Основные результаты суперкомпьютерной программы "СКИФ" Союзного Государства // Труды семинара "АКИИ'03: Третий расширенный семинар Использование методов искусственного интеллекта и высокопроизводительных вычислениях и в аэрокосмических исследованиях", М.: изд-во Физматлит, 2003 г. – С. 135-140. ISBN 5-940-52-065-9.
  12. Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.). – Челябинск: Издательский центр ЮУрГУ, 2010. – 723 с.
  13. Богачев К.Ю., Миргасимов А.Р. Об оптимизации вычислительных приложений для многопроцессорных систем с общей неоднородной памятью // Вычислительные методы и программирование. – 2010. – Т.11. – С. 193-197.
  14. Суперкомпьютерные технологии в науке, образовании и промышленности / Под редакцией: академика В.А. Садовниченко, академика Г.И. Савина, чл.-корр. РАН Вл.В. Воеводина. – М.: Издательство Московского университета, 2009. – 232 с., ил. ISBN 978-5-211-05719-7.
  15. Rajkumar Buyya. High performance cluster computing. – Melbourne, Australia: Prentice Hall, 1999. – 855 с.
  16. Proceedings of the twenty-second annual symposium on Principles of distributed computing / Annual ACM Symposium on Principles of Distributed Computing. – Boston, Massachusetts, USA. – 2003. – 380 с. ISBN:1-58113-708-7.
  17. Параллельные вычислительные технологии (ПаВТ'2009): Труды международной научной конференции (Нижний Новгород, 30 марта – 3 апреля 2009 г.). – Челябинск: Изд. ЮУрГУ, 2009. – 839 с.
  18. Камолкин А.В., Неверов В.С. Параллельные алгоритмы молекулярной динамики //

Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской суперкомпьютерной конференции (21-26 сентября 2009 г., г. Новороссийск). – М.: Изд-во МГУ, 2009. – 524 с. ISBN 978-5-211-05697-8.

19. Введение в грид-технологии / А.П. Демичев, В.А. Ильин, А.П. Крюков. – М.: Препринт НИИЯФ МГУ, 2007. – 87 с.

20. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ванн Стеен. – СПб.: Питер, 2003. – 877 с., ил. – (Серия “Классика computer science”). ISBN 5-272-00053-6.

21. Варенов Д.А. «Построение вычислительного кластера». Тезисы докладов федеральной школы-конференции по инновационному малому предпринимательству в приоритетных направлениях науки и высоких технологий. Москва, Российский государственный университет инновационных технологий и предпринимательства. 2005. с. 49 — 50.

22. Калашников Е.И. Адаптивные алгоритмы управления распределением нагрузки в многосерверных системах. Автореферат диссертации на соискание учёной степени к.т.н. МИЭМ, М: -2010, 17с.

**Старых Владимир Александрович**

ФГУ ГНИИ ИТТ «Информика» г. Москва

Кандидат технических наук, доцент, заместитель директора

Тел. (499) 155-87-28

[vstar@informika.ru](mailto:vstar@informika.ru)

**Starykh Vladimir A.**

State Institute of Information Technologies and Telecommunications “Informika”

Deputy Director

Tel. (499) 155-87-28

E-mail: [vstar@informika.ru](mailto:vstar@informika.ru)

**Варенов Дмитрий Анатольевич**

ФГУ ГНИИ ИТТ «Информика»

программист отдела программных систем

Тел. 155-87-30, (499) 155-83-1 , доб. 1106

[varenov@informika.ru](mailto:varenov@informika.ru)

**Varenov Dmitri A.**

State Institute of Information Technologies and Telecommunications “Informika”

Programmer of department of programmable systems

Tel. 155-87-30, (499) 155-83-11, add. 1106

E-mail: [varenov@informika.ru](mailto:varenov@informika.ru)

**Плосков Сергей Юрьевич**

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Российский государственный университет инновационных технологий и предпринимательства», г. Москва

Кандидат технических наук, старший преподаватель кафедры «Информационные Системы»

Тел. 155-87-30, (499) 155-83-11 , доб. 1106

[Sergploskov@yandex.ru](mailto:Sergploskov@yandex.ru)

**Ploskov Sergey Y.**

Russian State University for Innovation Technologies and Business

The senior teacher

Tel. 155-87-30, (499) 155-83-11 , add. 1106

E-mail: [Sergploskov@yandex.ru](mailto:Sergploskov@yandex.ru)

**Кузнецов Александр Юрьевич**

ФГУ ГНИИ ИТТ «Информика» г. Москва

Ведущий инженер Отдела программных систем

Тел.(499)155-87-30, (499) 155-83-11 , доб. 1121

[allexius@informika.ru](mailto:allexius@informika.ru)

**Kuznetsov Alexander Y.**

State Institute of Information Technologies and Telecommunications “Informika”  
The leading engineer  
Tel.(499)155-87-30, (499) 155-83-11 , add. 1121  
E-mail: [allexius@informika.ru](mailto:allexius@informika.ru)