МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И КОМПЬЮТЕРНЫХ СЕТЕЙ

УДК 004.72:004.42

РАЗРАБОТКА ПРОГРАММНОГО КОМПОНЕНТА ПОСТРОЕНИЯ РАСПРЕДЕЛЕННОГО ОБЛАЧНОГО ХРАНИЛИЩА ДАННЫХ ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ

Д. А. Перепелкин, д.т.н., доцент, декан факультета вычислительной техники РГРТУ, Рязань, Россия; orcid.org/0000-0003-4775-5745, e-mail: dmitryperepelkin@mail.ru
А. Н. Сапрыкин, к.т.н., доцент кафедры САПР ВС РГРТУ, Рязань, Россия; orcid.org/0000-0000-0000-000X, e-mail: alexpizza@rambler.ru
М. А. Иванчикова, ассистент кафедры САПР ВС РГРТУ, Рязань, Россия; orcid.org/0000-0002-9615-2898, e-mail: ivanchikova.masha@yandex.ru
С. С. Косоруков, инженер кафедры САПР ВС РГРТУ, Рязань, Россия; orcid.org/0000-0000-0000-000X, e-mail: kosorukov.s.s@rsreu.ru

Современные тенденции увеличения числа подключенных к Интернету устройств, экспоненциальный рост объемов информации, развитие сетевых и облачных технологий меняют все сферы деятельности. ***Целью работы*** *является разработка программного компонента построения распределенного облачного хранилища данных образовательного учреждения. Предлагаются архитектура и программный инструментарий распределенной обработки и хранения данных пользователей облачной платформы образовательного учреждения. Детально описываются основные диаграммы классов разработанной облачной платформы и принципы взаимодействия ее основных компонентов. Рассматриваются процесс функционирования и особенности построения сетевой инфраструктуры разработанной облачной платформы распределенного хранения данных. Предложенная в работе облачная платформа реализует модульный принцип взаимодействия, позволяет оперативно подключать и развертывать новые цифровые сервисы и услуги за счет использования широкого набора программных компонентов и открытых API-интерфейсов.*

Ключевые слова: *распределенные вычисления, облачные вычисления, хранилище данных, облачная платформа, программный компонент.*

DOI: 10.21667/1995-4565-2019-70-3-14

Введение

Современные российские информационные системы и платформы обработки больших массивов данных построены на основе облачных решений. Основными лидерами рынка облачных платформ являются компании Microsoft, Google, Amazon, Cisco, SAP, Oracle и Siemens. Среди российских облачных платформ выделяют продукты компаний Yandex, Mail.ru, КРОК и Софтлайн. В настоящий момент на рынке цифровых облачных услуг существует 3 основные модели организации облачных сервисов: программное обеспечение как услуга (Software-as-a-Service, SaaS), платформа как услуга (Platform-as-a-Service, PaaS) и инфраструктура как услуга (Infrastructure-as-a-Service, IaaS). По данным аналитического агентства Gartner, глобальный рынок облачных сервисов растет каждый год примерно на 30 %. Аналитическое агентство J'Son&Partners отмечает, что по итогам 2018 года больше половины заказчиков облачных сервисов IaaS/PaaS были из категорий торговли и оказания

услуг, а именно: 39 % пользователей облачных сервисов представляли торговые предприятия, 20 % – телекоммуникационные компании, 16 % занимались предоставлением различных услуг, 8 % работали в производственном секторе, 7 % работали в области государственного управления, а 4 % на финансовом рынке. Таким образом, четко просматривается тенденция широкой ориентации различных государственных компаний и коммерческих организаций на различные облачные услуги и сервисы. Однако предлагаемые решения от зарубежных производителей облачных платформ, несмотря на высокое качество, являются довольно дорогостоящими. Продукты российских компаний имеют приемлемую цену, но узко ориентированы на конкретную область и не поддерживают новые технологии и определенные типы сервисов. Российские компании при построении облачных платформ используют в основном готовые решения, т.е. практически не имеют возможности доработки программного обеспечения под требования конкретного заказчика. Стоит отметить, что наиболее перспективным и инновационным подходом к построению облачных платформ является подход, основанный на интеграции технологий программно-конфигурируемых сетей (ПКС) и облачных вычислений. Технология ПКС позволяет программно дорабатывать сетевые решения под конкретного заказчика, перевести сетевую инфраструктуру и ее элементы под контроль настраиваемого программного обеспечения, сделать процесс управления и обработки данных более интеллектуальными и простым [1, 2]. Интеграция данных технологий позволяет динамически масштабировать и настраивать сетевые и вычислительные ресурсы под текущую нагрузку, обеспечивая их оптимальное использование по принципу «оплата по мере использования». Централизация управления цифровой облачной платформой за счет применения технологии ПКС позволяет обеспечить высокую эластичность предоставления и перераспределения цифровых услуг, организовать повсеместный доступ к цифровым сервисам, автоматически контролировать и оптимизировать ресурсы по требованию. Таким образом, актуальной научной задачей является разработка новых программных решений для быстрого построения и развертывания распределенных облачных платформ обработки, передачи и хранения данных.

Теоретическая часть

В настоящее время растет количество информационных систем, технологий и методов, связанных с передачей и обработкой больших данных. Модернизация устройств для поддержки сетей нового поколения приводит к увеличению капитальных и эксплуатационных расходов. Чтобы избежать дополнительных затрат, предприятия используют единую сетевую инфраструктуру в виде центров обработки данных (ЦОД). В работах [3, 4] показаны варианты реализации сетей ЦОД, предложены математические модели облачных ЦОД, рассмотрены методы оптимизации виртуализированных ЦОД. Различные подходы к распределению ресурсов в ЦОД на основе платформы OpenStack описаны в работах [5, 6]. В работах [7, 8] предложены алгоритмы контроля трафика мультипровайдерных корпоративных сетей ЦОД. Задачи реконфигурирования параметров и структур распределенных ЦОД подробно рассматриваются в работах [9-13]. Вопросы построения многоуровневых облачных, кластерных и грид вычислительных систем и комплексов детально рассмотрены в работах [14-17]. Задачи практического построения ПКС-инфраструктур освящены в работах [18-27]. Эффективные подходы управления ресурсами облачных ПКС-инфраструктур рассмотрены в работах [28, 29].

Проведенный анализ известных работ показывает, что актуальным научным направлением является разработка новых программных компонентов построения и развертывания программно-конфигурируемых облачных платформ для распределенной обработки, передачи и хранения данных в различных сферах деятельности.

Для создания распределенного облачного хранилища данных был разработан фреймворк Pandora. Pandora – это php-фреймворк для быстрой и удобной разработки больших и модульных систем. Для быстрой установки фреймворка разработан менеджер пакетов Composer. В основу разработки Pandora были положены принципы минимализма, легкости повторного

использования кода, при этом не закрывались возможности для дальнейшего расширения функциональности.

Фреймворк Pandora состоит из следующих основных компонентов:

- Core – основные классы ядра фреймворка;
- Libs – подключаемые модули;
- Plugins – готовые блоки функциональности;
- Widgets – визуальные компоненты.

При инициализации класс приложения создает контейнер зависимостей $this->container. Переопределив метод dependencies в классе приложения, можно добавлять и переопределять зависимости. Диаграмма основных классов фреймворка Pandora для построения распределенного облачного хранилища данных приведена на рисунке 1.

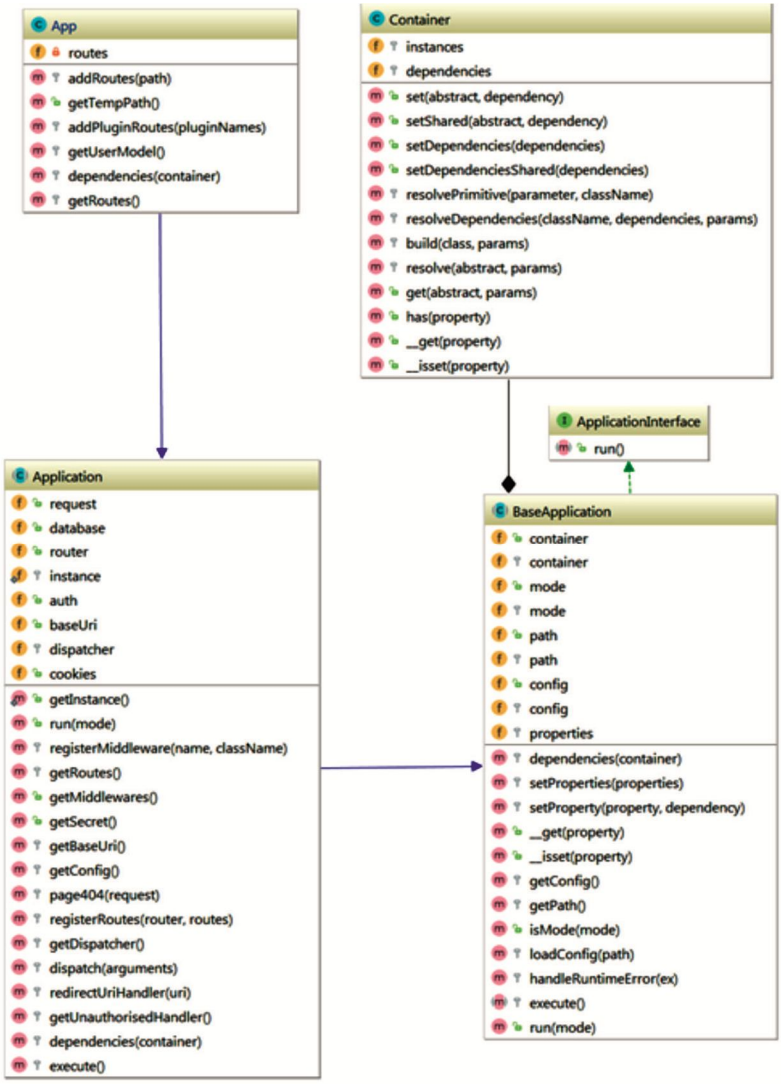

Рисунок 1 – Диаграмма основных классов облачного хранилища данных
Figure 1 – Diagram of the main classes of cloud data storage

Общепринятым подходом реализации web-приложения является использование архитектуры, при которой каждый запрос к web-приложению представлен классом *Request* и обработчиком приложения, возвращающим ответ – класс *Response*. Для выбора обработчика используется маршрутизация, которую осуществляет класс *Router*. Диаграммы основных классов для обработки данных основного приложения приведены на рисунке 2.

Класс *Request* предоставляет объектный интерфейс для работы с переменными запроса. Функции *get*(string $param), *post*(string $param) и *file*(string $param) служат для доступа к переменным.

Для возвращаемых ответов использует класс *Response*. В него оборачиваются данные ответа, при каждом обращении к URL, добавляя необходимую информацию HTTP ответа.

Класс *Router* реализует маршрутизацию. Правило маршрутизации реализуется с помощью метода *add*(string $routePath, $route). Простейшие маршруты принимают URI (путь) и функцию-замыкание. Маршруты определяются в файле маршрутов router.php.

Благодаря модульной архитектуре и использованию интерфейсов, многие части системы могут быть расширены, дополнены или заменены. Например, возможно подключение другого шаблонизатора или библиотеки для работы с базой данных.

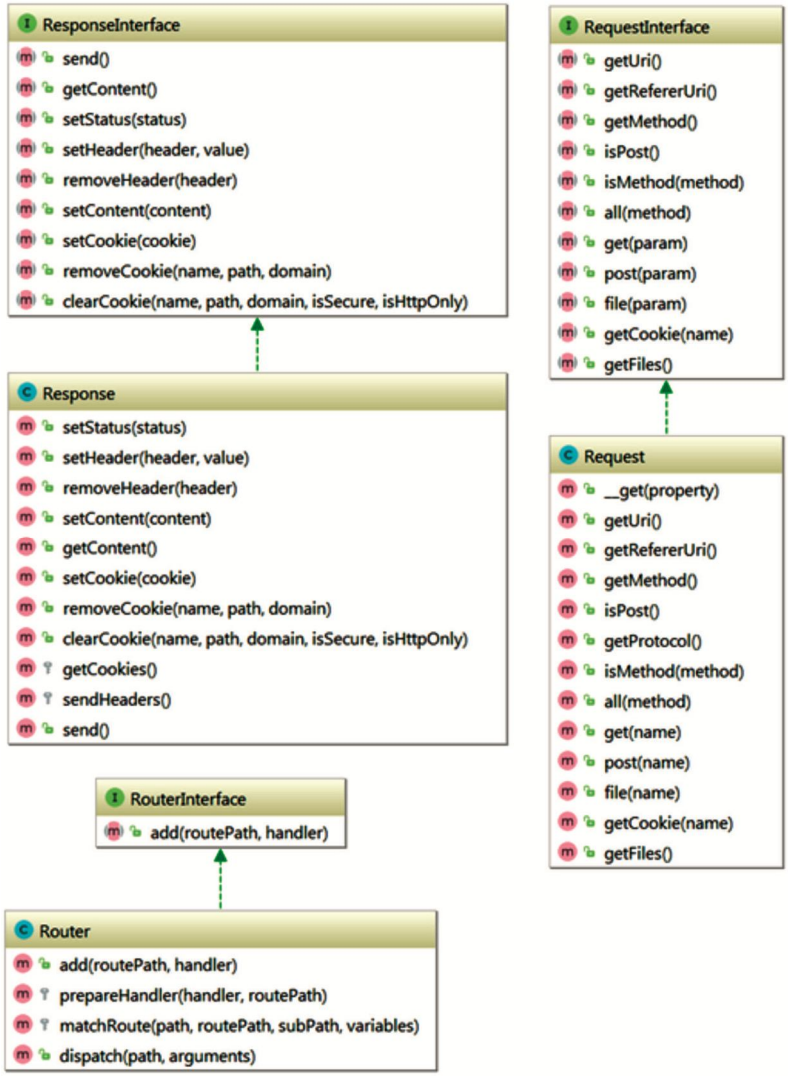

Рисунок 2 – Диаграмма классов Request, Response и Router
Figure 2 – Diagram of the Request, Response, and Router classes

Экспериментальные исследования

Точкой входа в приложение распределенного облачного хранилища данных являлся файл public/index.php:

```
<?php
    $loader = require(__DIR__.'/../vendor/autoload.php');
    $loader->setPsr4('App\\', __DIR__.'/../app');
        require __DIR__.'/../app/helpers.php';
        $mode = 'dev';
            (new App\App())->run($mode);
```

При наследовании класса приложения от Pandora3\Core\Application\Application, он получил следующие зависимости.

- Класс *Request* реализующий *RequestInterface*.
- Класс *Router* реализующий *RouterInterface*.
- Класс *DatabaseConnection*, реализующий *DatabaseConnectionInterface*.

Архитектура приложения облачной платформы построена с использованием принципа MVC. Структура каталогов приложения облачной платформы приведена в таблице.

Структура каталогов облачной платформы
Directory structure of cloud platform

Каталог	Описание
app	Каталог приложения проекта
config	Каталог с файлами конфигурации
app/Plugins	Каталог плагинов проекта
app/Models	Каталог моделей данных проекта
app/Views	Каталог представлений проекта
app/Widgets	Каталог виджетов проекта
public	Каталог статических файлов
storage	Каталог локального хранилища
vendor	Каталог пакетов composer

Модели предоставляют объектный способ работы с базой данных, реализуя подход ORM. В проекте использовалась ORM Eloquent. Каталог Model содержит модели сущностей приложения, а именно:

- пользователь системы;
- файл;
- каталог;
- группа;
- пользователь группы.

Диаграмма основных классов программного компонента построения распределенного облачного хранилища данных приведена на рисунке 3. Логика компонента разделена на блоки (каталог Plugin), каждый из которых состоит из контроллеров, классов форм и представлений. Представления находятся в каталоге Views. В качестве шаблонизатора был выбран Twig, так как он обладает такими преимуществами, как чёткое разделение текста и переменных в шаблоне, наличие циклов, наличие условных блоков и возможность создавать сложные условия, компилирующий режим при котором – последующая генерация шаблона занимает мало времени.

Правила маршрутизации находятся в файле app/routes.php:

```
<?php
return [
    '/*' => \App\Controllers\HomeController::class,
];
```

В качестве ключей указываются маршруты, а в качестве значений – имена классов контроллеров. В качестве значения вместо класса контроллера можно указать любой класс, который поддерживает интерфейс *RouteInterface* или функцию-замыкание.

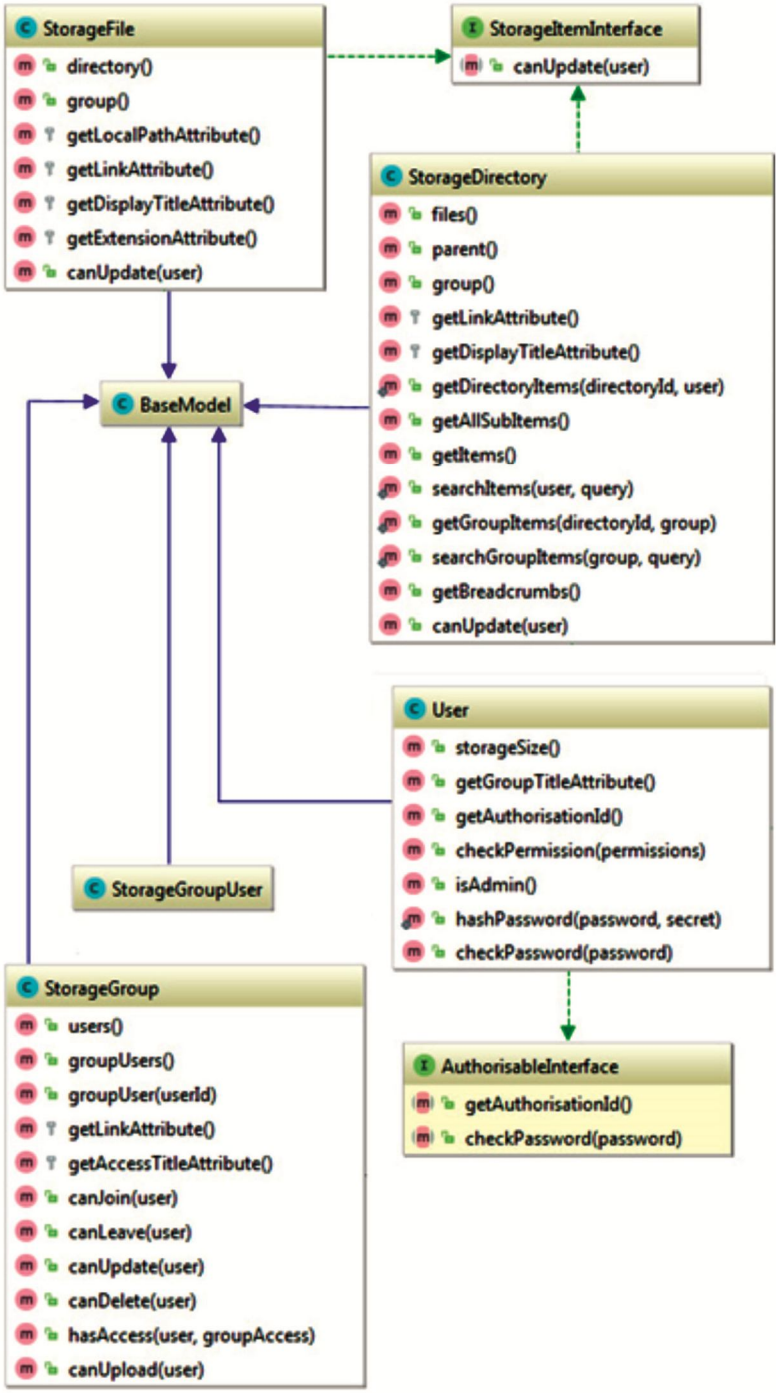

Рисунок 3 – Диаграмма основных классов программного компонента построения распределенного облачного хранилища данных
Figure 3 – Diagram of main classes of software component to build distributed cloud data storage

Общая схема функционирования программного компонента распределенного облачного хранилища данных приведена на рисунке 4.

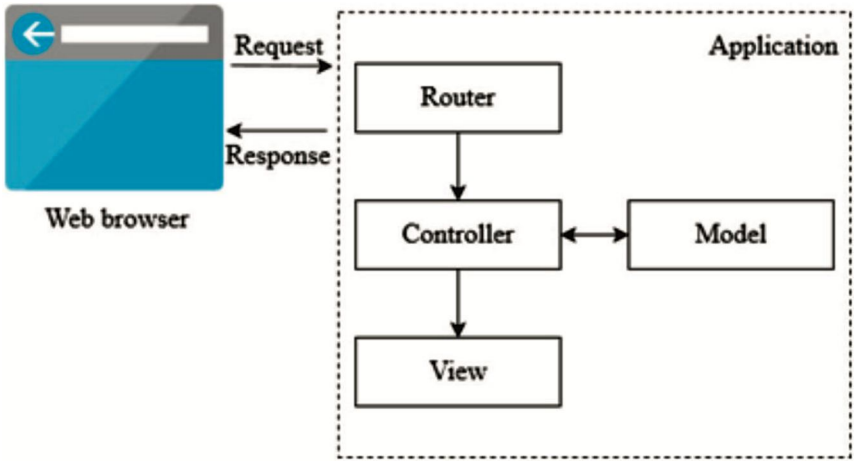

Рисунок 4 – Общая схема функционирования программного компонента распределенного облачного хранилища данных
Figure 4 – General scheme of distributed cloud data storage software component operation

На вход приложения приходит сообщение *Request*. В соответствии с правилами маршрутизации вызывается соответствующий *Controller*. *Controller* обрабатывает запросы web-приложения, загружает данные из модели и возвращает представление View. Фреймворк Pandora предоставляет абстракцию для работы с файловой системой и содержит простые в использовании драйверы для работы с локальными файловыми системами, такими как Amazon S3. Более того, можно очень просто переключаться между этими вариантами хранения файлов, поскольку используется одинаковый API интерфейс. Общая схема сетевой инфраструктуры распределенного облачного хранилища данных приведена на рисунке 5.

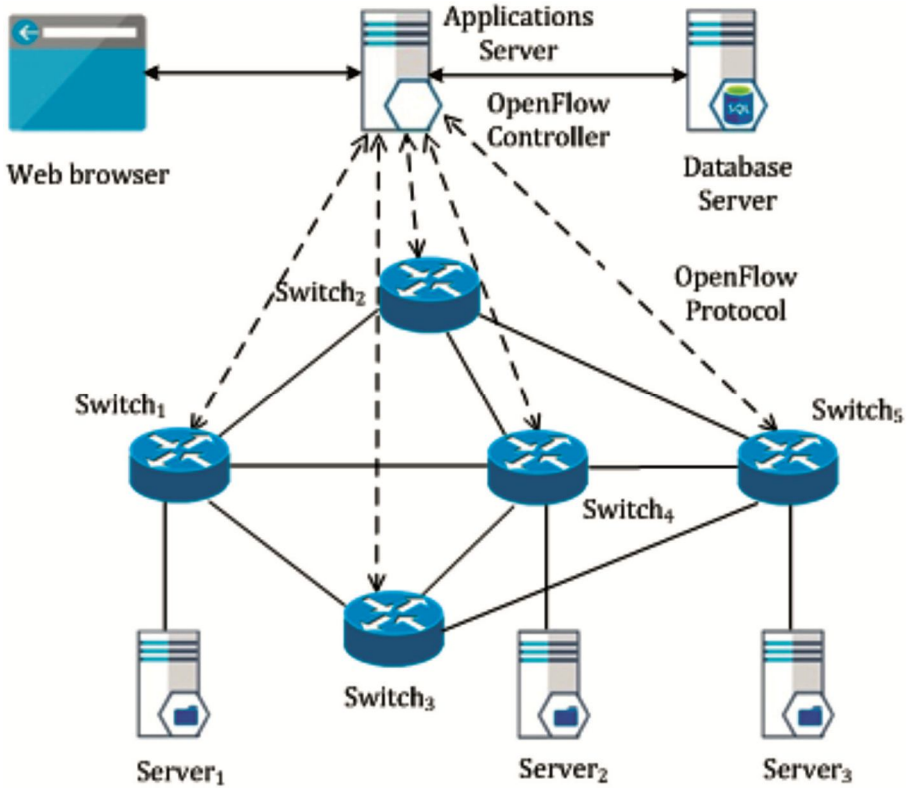

Рисунок 5 – Общая схема сетевой инфраструктуры распределенного облачного хранилища данных
Figure 5 – General scheme of distributed cloud data storage network infrastructure

На рисунке 5 сервер приложения взаимодействует с сервером базы данных и с файловыми серверами через программно-конфигурируемую компьютерную сеть (ПКС). Распределенная архитектура ПКС позволяет обеспечить хранение файлов на разных серверах.

Интерфейс главной страницы распределенного облачного хранилища данных образовательного учреждения приведен на рисунке 6. Интерфейс пользователя представлен на рисунке 7.

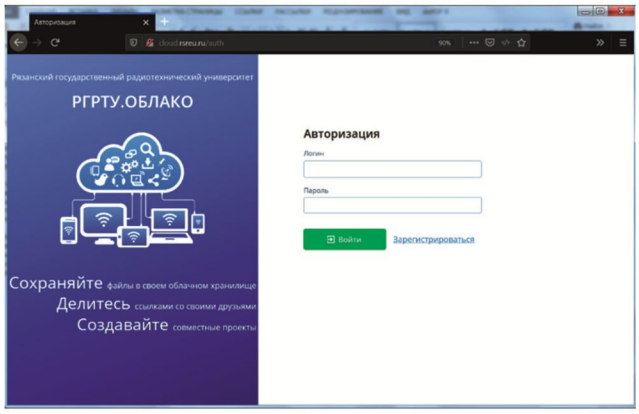

Рисунок 6 – Интерфейс главной страницы распределенного облачного хранилища данных
Figure 6 – Interface of distributed cloud data storage home page

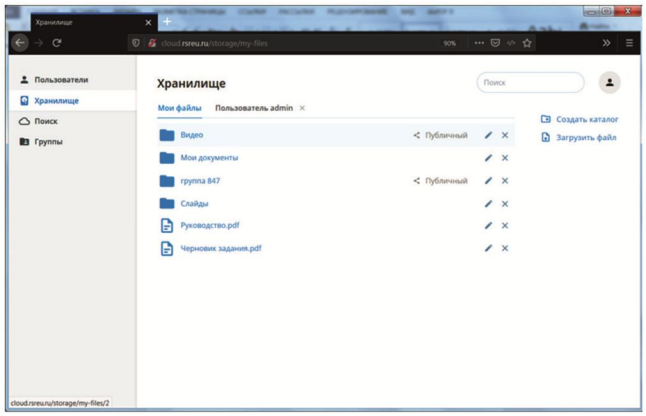

Рисунок 7 – Интерфейс пользователя распределенного облачного хранилища данных
Figure 7 – Distributed cloud data storage user interface

Система предоставляет возможность создавать каталоги и файлы различных типов, выполнять поиск данных в хранилище. При создании приватного каталога все вложенные файлы и каталоги будут иметь приватный доступ. Облачная платформа реализует несколько категорий пользователей с определенным уровнем доступа к данным: супер администратора, администратора и пользователя. Суперадминистратор курирует и управляет работой администраторов цифровой платформы, администраторы имеют возможность управлять пользователями и группами пользователей (рисунок 8). Также в облачной платформе реализована возможность создания своих групп, вступления и выходы из открытых групп других пользователей, поиск по группам. Интерфейс работы с группами пользователей приведен на рисунке 9.

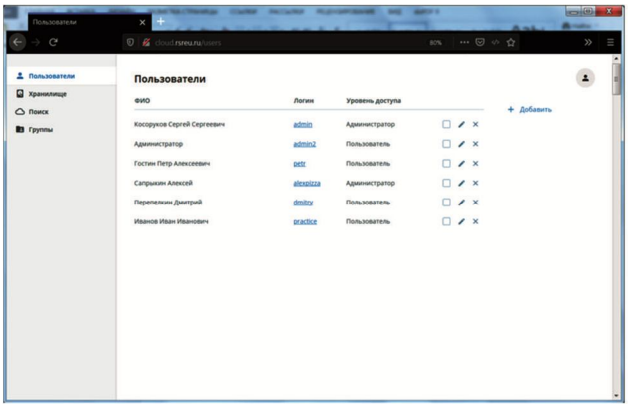

Рисунок 8 – Интерфейс основных пользователей распределенного облачного хранилища данных
Figure 8 – Interface of distributed cloud data storage main users

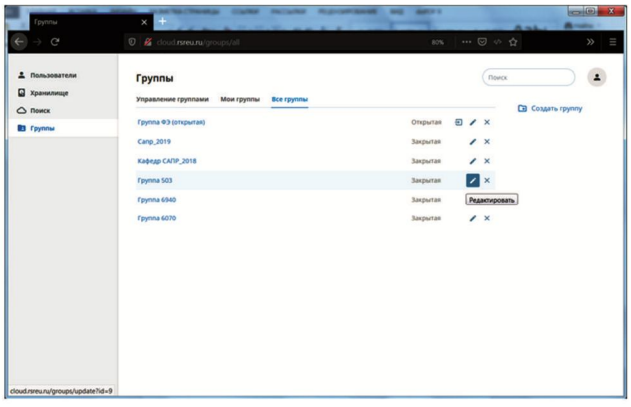

Рисунок 9 – Интерфейс работы с группами пользователей распределенного облачного хранилища данных
Figure 9 – Interface for working with user groups of distributed cloud data storage

Заключение

В статье разработан программный компонент построения распределенного облачного хранилища данных образовательного учреждения. Детально описана архитектура облачного хранилища и ее основные компоненты. Также в работе рассмотрены особенности построения и функционирования сетевой инфраструктуры разработанной облачной платформы. Предложенная в работе облачная платформа реализует модульный принцип взаимодействия, позволяет оперативно подключать и развертывать новые цифровые сервисы и услуги за счет использования широкого набора программных компонентов и открытых API-интерфейсов.

Работа выполнена при финансовой поддержке гранта Президента РФ МД-1826.2019.9.

Библиографический список

1. **McKeown N., Anderson T., Balakrishnan H. et al.,** «Openflow: enabling innovation in campus networks», ACM SIGCOMM Computer Communication Review, 2008, vol. 38, no. 2, pp. 69-74.

2. **Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijendam J., McKeown N.,** «Maturing of OpenFlow and Software-Defined Networking Through Deployments», Computer Networks, 2014, Vol. 61, pp. 151-175.

3. **Тарасов В. Н. и др.** Математические модели облачного вычислительного центра обработки данных с использованием OpenFlow // Вестник Оренбургского государственного университета. 2012. № 9 (145).

4. **Erickson D., Heller B. et al.,** «Optimizing a Virtualized Data Center», Proceedings ACM SIGCOMM, 2011, pp. 478-479.

5. **Zotov I., Kostenko V.,** «Resource allocation algorithm in data centers with a unified scheduler for different types of resources», Journal of Computer and Systems Sciences International, 2015, vol. 54, no. 1, pp. 59-68.

6. **Vdovin P., Kostenko V.,** «Algorithm for resource allocation in data centers with independent schedulers for different types of resources», Journal of Computer and Systems Sciences International, 2014, vol. 53, no. 6, pp. 854-866.

7. **Корячко В. П., Перепелкин Д. А., Иванчикова М. А.** Алгоритм адаптивной маршрутизации в корпоративных сетях нескольких провайдеров связи // Вестник Рязанского государственного радиотехнического университета. 2013. № 2(44). С. 52-56.

8. **Корячко В. П., Перепелкин Д. А., Иванчикова М. А.** Алгоритм парных переходов каналов связи при динамическом изменении нагрузки в корпоративных сетях нескольких провайдеров связи с различными зонами покрытия // Вестник Рязанского государственного радиотехнического университета. 2014. № 48. С. 68-76.

9. **Корячко В. П., Перепелкин Д. А., Иванчикова М. А.** Разработка и исследование алгоритма быстрой перемаршрутизации трафика между центрами обработки данных // Радиотехника. 2016. № 8. С. 133-139.

10. **Koryachko V., Perepelkin D., Ivanchikova M.** Adaptive rerouting of data flows in distributed data centers. Microprocessors and Microsystems, 2017, Vol. 52, no. 7, pp. 505-509. DOI: 10.1016/j.micpro. 2017.05.009.

11. **Perepelkin D., Ivanchikova M.** Improved adaptive routing algorithm in distributed data centers. Advances in electrical and electronic engineering, 2017, vol. 15, no. 3, pp. 502-506. DOI: 10.15598/aeee. v15i3.2185.

12. **Корячко В. П., Перепелкин Д. А., Иванчикова М. А.** Разработка программного инструментария реконфигурирования параметров мультипровайдерных сетей распределенных центров обработки данных // Вестник Рязанского государственного радиотехнического университета. 2019. № 68. С. 21-27.

13. **Иванчикова М. А.** Алгоритм масштабирования структур мультипровайдерных сетей распределенных центров обработки данных // Вестник Рязанского государственного радиотехнического университета. 2019. № 69. С. 74-85.

14. **Воеводин В. В.** Параллельные вычисления. Санкт-Петербург.: БХВ-Петербург, 2002. 608 с.

15. **Балашов Н. А., Баранов А. В., Кореньков В. В., Кутовский Н. А., Нечаевский А. В., Семенов Р. Н.** Облачный сервис ОИЯИ: статус и перспективы // Труды Института системного программирования РАН. 2015. Т. 27. № 6. С. 345-354.

16. **Балашов Н. А., Баранов А. В., Кадочников И. С., Кореньков В. В., Кутовский Н. А., Нечаевский А. В., Пелеванюк И. С.** Программный комплекс интеллектуального диспетчирования и адаптивной самоорганизации виртуальных вычислительных ресурсов на базе облачного центра ЛИТ ОИЯИ // Известия ЮФУ. Технические науки. 2016. № 12 (185). С. 92-103.

17. **Велихов В. Е., Климентов А. А., Машинистов Р. Ю., Пойда А. А., Рябинкин Е. А.** Интеграция гетерогенных вычислительных мощностей НИЦ «Курчатовский институт» для проведения масштабных научных вычислений // Известия ЮФУ. Технические науки. №11 (184). 2016.

18. **Choumas K., Makris N., Korakis T., Tassiulas L. and Ott M.** «Testbed Innovations for Experimenting with Wired and Wireless Software Defined Networks», IEEE 35th International Conference, 2015, pp. 87-94.

19. **Antonenko V., Smelyansky R., Baldin R., Izhvanov Yu., Gugel Y.** «Towards SDI-bases Infrastructure for supporting science in Russia», Proceedings SDN and NFV: Next Generation of Computational Infrastructure – 2014, International Science and Technology Conference MoNeTec 2014, 2014, pp. 1-7.

20. GENI: Exploring Network of the Future, http://www.geni.net.

21. **Егоров В. Б.** Некоторые вопросы практической реализации концепции SDN // Системы и средства информатики. 2016. Т. 26. № 1. С. 109-120.

22. **Гузев О. Ю., Чижов И. В.** Балансировка нагрузки в защищенных сетях с использованием технологии SDN // Системы и средства информатики. 2018. Т. 28. № 1. С. 123-138.

23. **Гузев О. Ю., Чижов И. В.** SDN-балансировка на криптографические маршрутизаторы при объединении центров обработки данных // Системы и средства информатики. 2018. Т. 28. № 1. С. 139-155.

24. **Бахарева Н. Ф., Полежаев П. Н., Ушаков Ю. А., Шухман А. Е.** Управление корпоративными программно-конфигурируемыми сетями // Вестник Оренбургского государственного университета. 2015. № 13 (188). С. 108-113.

25. **L. Tello-Oquendo, I. F. Akyildiz, S.-C. Lin and V. Pla (2018).** SDN-based architecture for providing reliable Internet of Things connectivity in 5G systems. Proceedings 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net). 2018. DOI:10.23919/medhocnet.2018.8407080/.

26. **Корячко В. П., Перепелкин Д. А., Иванчикова М. А., Бышов В. С., Цыганов И. Ю.** Программная инфраструктура и визуальная среда распределенной обработки потоков данных в программно-конфигурируемых сетях // Вестник Рязанского государственного радиотехнического университета. 2018. № 65. С. 44-54.

27. **Корячко В. П., Перепелкин Д. А., Иванчикова М. А., Бышов В. С.** Визуальная веб-ориентированная среда динамического управления потоками данных в кампусных программно-конфигурируемых сетях // Моделирование и анализ информационных систем. 2019. Т. 26. № 1. С. 63-74.

28. **Galán-Jiménez J., Polverini M. and Cianfrani V,** «Reducing the Reconfiguration Cost of Flow Tables in Energy-Efficient Software Defined Networks», Computer Communications, 2018, vol. 128, pp. 95-105. DOI: 10.1016/j.comcom.2018.07.022.

29. **Eramo V., Lavacca F. G., Catena T., Polverini M., Cianfrani A.** «Effectiveness of Segment Routing Technology in Reducing the Bandwidth and Cloud Resources Provisioning Times in Network Function Virtualization Architectures», Future Internet, 2019, vol. 11, no. 71. DOI: 10.3390/fi11030071.

UDC 004.72:004.42

DEVELOPMENT OF SOFTWARE FOR BUILDING A DISTRIBUTED CLOUD DATA STORAGE OF AN EDUCATIONAL INSTITUTION

D. A. Perepelkin, Dr. Sc. (Tech.), Associate Professor, Dean of Faculty of Computer Engineering, RSREU, Ryazan, Russia;
orcid.org/0000-0003-4775-5745, dmitryperepelkin@mail.ru
A. N. Saprykin, Ph.D (Tech.), associate professor of CAD department, RSREU, Ryazan, Russia;
orcid.org/0000-0002-3882-1301, e-mail: alexpizza@rambler.ru
M. A. Ivanchikova, assistant CAD department, RSREU, Ryazan, Russia;
orcid.org/0000-0002-9615-2898, ivanchikova.masha@yandex.ru
S. S. Kosorukov, engineer CAD department, RSREU, Ryazan, Russia;
orcid.org/0000-0002-9470-242X, e-mail: kosorukov.s.s@rsreu.ru

The increasing of number of devices connected to the Internet, the exponential growth of information volumes, the development of network and cloud technologies are the latest trends that are changing all areas of activity. ***The aim of the work*** *is to develop a software component for building a distributed cloud data storage of an educational institution. Architecture and software tools for distributed processing and storage of data owned by users of the cloud platform of educational institution are proposed. Main class diagrams of the developed cloud platform and the principles of interaction of its main components are described in detail. The functioning and the peculiarities of building a network infrastructure of the developed cloud platform for distributed data storage are considered. Cloud platform proposed in the work implements a modular principle of interaction, allowing us to quickly connect and deploy new digital services through the use of a wide range of software components and open APIs.*

Keywords: *distributed computing, cloud computing, data storage, cloud platform, software component.*

DOI: 10.21667/1995-4565-2019-70-3-14

References

1. **McKeown N., Anderson T., Balakrishnan H. et al.** Openflow: enabling innovation in campus networks, *ACM SIGCOMM Computer Communication Review*. 2008, vol. 38, no. 2, pp. 69-74.

2. **Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijendam J., McKeown N.** Maturing of OpenFlow and Software-Defined Networking Through Deployments. *Computer Networks*. 2014, vol. 61, pp. 151-175.

3. **Tarasov V. et al.,** Matematicheskie modeli oblachnogo vychislitel'nogo centra obrabotki dannyh s ispol'zovaniem OpenFlow, *Vestnik Orenburgskogo gosudarstvennogo universiteta*, 2012, no. 9 (145) (in Russian).

4. **Erickson D., Heller B. et al.** Optimizing a Virtualized Data Center. *Proceedings ACM SIGCOMM*. 2011, pp. 478-479.

5. **Zotov I., Kostenko V.** Resource allocation algorithm in data centers with a unified scheduler for different types of resources. *Journal of Computer and Systems Sciences International*. 2015, vol. 54, no. 1, pp. 59-68.

6. **Vdovin P., Kostenko V.** Algorithm for resource allocation in data centers with independent schedulers for different types of resources. *Journal of Computer and Systems Sciences International*. 2014, vol. 53, no. 6, pp. 854-866 (in Russian).

7. **Koryachko V., Perepelkin D., Ivanchikova M.** Algoritm adaptivnoj marshrutizacii v korporativnyh setyah neskol'kih provajderov svyazi. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2013, no. 2(44), pp. 52-56 (in Russian).

8. **Koryachko V., Perepelkin D., Ivanchikova M.** Algoritm parnyh perekhodov kanalov svyazi pri dinamicheskom izmenenii nagruzki v korporativnyh setyah neskol'kih provajderov svyazi s razlichnymi zonami pokrytiya. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2014, no. 48, pp. 68-76 (in Russian).

9. **Koryachko V., Perepelkin D., Ivanchikova M.** Razrabotka i issledovanie algoritma bystroj peremarshrutizacii trafika mezhdu centrami obrabotki dannyh. *Radiotekhnika*. 2016, no. 8, pp. 133-139 (in Russian).

10. **Koryachko V., Perepelkin D., Ivanchikova M.** Adaptive rerouting of data flows in distributed data centers. *Microprocessors and Microsystems*. 2017, vol. 52, no. 7, pp. 505-509. DOI: 10.1016/j.micpro. 2017.05.009.

11. **Perepelkin D., Ivanchikova M.** Improved adaptive routing algorithm in distributed data centers. *Advances in electrical and electronic engineering*. 2017, vol. 15, no. 3, pp. 502-506. DOI: 10.15598/aeee. v15i3.2185.

12. **Koryachko V., Perepelkin D., Ivanchikova M.** Razrabotka programmnogo instrumentariya rekonfigurirovaniya parametrov mul'tiprovajdernyh setej raspredelennyh centrov obrabotki dannyh. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2019, no. 68, pp. 21-27 (in Russian).

13. **Ivanchikova M.** Algoritm masshtabirovaniya struktur mul'tiprovajdernyh setej raspredelennyh centrov obrabotki dannyh. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2019, no. 69, pp. 74-85 (in Russian).

14. **Voevodin V.** *Parallel'nye vychisleniya*. Sankt-Peterburg.: BHV-Peterburg, 2002, 608 p. (in Russian).

15. **Balashov N., Baranov A., Koren'kov V., Kutovskij N., Nechaevskij A., Semenov R.** Oblachnyj servis OIYAI: status i perspektivy. *Trudy Instituta sistemnogo programmirovaniya RAN*. 2015, vol. 27, no. 6, pp. 345-354 (in Russian).

16. **Balashov N., Baranov A., Kadochnikov I., Koren'kov V., Kutovskij N., Nechaevskij A., Pelevanyuk I.** Programmnyj kompleks intellektual'nogo dispetchirovaniya i adaptivnoj samoorganizacii virtual'nyh vychislitel'nyh resursov na baze oblachnogo centra LIT OIYAI. *Izvestiya YUFU. Tekhnicheskie nauki*. 2016, no. 12 (185), pp. 92-103 (in Russian).

17. **Velihov V., Klimentov A., Mashinistov R., Pojda A., Ryabinkin E.** Integraciya geterogennyh vychislitel'nyh moshchnostej NIC «Kurchatovskij institut» dlya provedeniya masshtabnyh nauchnyh vychislenij. *Izvestiya YUFU. Tekhnicheskie nauki*. 2016, no. 11 (184), 2016 (in Russian).

18. **Choumas K., Makris N., Korakis T., Tassiulas L., and Ott M.** Testbed Innovations for Experimenting with Wired and Wireless Software Defined Networks. *IEEE 35th International Conference*. 2015, pp. 87-94.

19. **Antonenko V., Smelyansky R., Baldin R., Izhvanov Yu., Gugel Y.** Towards SDI-bases Infrastructure for supporting science in Russia, Proceedings SDN and NFV: Next Generation of Computational Infrastructure – 2014. *International Science and Technology Conference MoNeTec 2014*. 2014, pp. 1-7.

20. GENI: Exploring Network of the Future, http://www.geni.net.

21. **Egorov V.** Nekotorye voprosy prakticheskoj realizacii koncepcii SDN. *Sistemy i sredstva informatiki*. 2016, vol. 26, no. 1, pp. 109-120 (in Russian).

22. **Guzev O., Chizhov I.** Balansirovka nagruzki v zashchishchennyh setyah s ispol'zovaniem tekhnologii SDN. *Sistemy i sredstva informatiki*. 2018, vol. 28, no. 1, pp. 123-138 (in Russian).

23. **Guzev O., Chizhov I.** SDN-balansirovka na kriptograficheskie marshrutizatory pri ob"edinenii centrov obrabotki dannyh. *Sistemy i sredstva informatiki*. 2018, vol. 28, no. 1, pp. 139-155 (in Russian).

24. **Bahareva N., Polezhaev P., Ushakov Yu., Shuhman A.** Upravlenie korporativnymi programmno-konfiguriruemymi setyami. *Vestnik Orenburgskogo gosudarstvennogo universiteta*. 2015, no. 13 (188), pp. 108-113 (in Russian).

25. **L. Tello-Oquendo, I. F. Akyildiz, S.-C. Lin and V. Pla (2018).** SDN-based architecture for providing reliable Internet of Things connectivity in 5G systems. *Proceedings 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. 2018. DOI:10.23919/medhocnet.2018.8407080/.

26. **Koryachko V., Perepelkin D., Ivanchikova M., Byshov V., Cyganov I.** Programmnaya infrastruktura i vizual'naya sreda raspredelennoj obrabotki potokov dannyh v programmno-konfiguriruemyh setyah. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2018, no. 65, pp. 44-54 (in Russian).

27. **Koryachko V., Perepelkin D., Ivanchikova M., Byshov V,** Vizual'naya veb-orientirovannaya sreda dinamicheskogo upravleniya potokami dannyh v kampusnyh programmno-konfiguriruemyh setyah. *Modelirovanie i analiz informacionnyh system*. 2019, vol. 26, no. 1, pp. 63-74 (in Russian).

28. **Galán-Jiménez J., Polverini M. and Cianfrani V.** Reducing the Reconfiguration Cost of Flow Tables in Energy-Efficient Software Defined Networks. *Computer Communications*. 2018, vol. 128, pp. 95-105. DOI: 10.1016/j.comcom.2018.07.022.

29. **Eramo V., Lavacca F.G., Catena T., Polverini M., Cianfrani A.** Effectiveness of Segment Routing Technology in Reducing the Bandwidth and Cloud Resources Provisioning Times in Network Function Virtualization Architectures. *Future Internet*. 2019, vol. 11, no. 71. DOI: 10.3390/fi11030071.