



УДК 004.75

## ИНФРАСТРУКТУРА ВЫСОКОПРОИЗВОДИТЕЛЬНОЙ КОМПЬЮТЕРНОЙ СИСТЕМЫ ДЛЯ РЕАЛИЗАЦИИ ОБЛАЧНЫХ СЕРВИСОВ ХРАНЕНИЯ И АНАЛИЗА ДАННЫХ ПЕРСОНАЛЬНОЙ МЕДИЦИНЫ

## HIGH-PERFORMANCE COMPUTER SYSTEM INFRASTRUCTURE FOR CLOUD SERVICE IMPLEMENTATION FOR STORING AND ANALYZING PERSONAL MEDICAL DATA

Д.С. Батищев, В.М. Михелев  
D.S. Batishchev, V.M. Mikhelev

*Белгородский государственный национальный исследовательский университет,  
Россия, 308015, Белгород, ул. Победы, 85*

*Belgorod State National Research University, 85 Pobeda St, Belgorod, 308015, Russia*

*e-mail: batishchev@bsu.edu.ru, mikhelev@bsu.edu.ru*

*Аннотация.* В данной статье рассматривается способ создания инфраструктуры облачной системы для хранения и обработки данных персональной медицины. Рассматриваются возможные варианты решения этой проблемы. Приводится описание разработанной архитектуры развернутого облачного окружения и прототип пользовательского интерфейса.

*Resume.* This article discusses how to create a cloud infrastructure systems for storing and processing personal medical data. Possible solutions to this problem. The description of the designed architecture deployed cloud environments and prototype user interface.

*Ключевые слова:* облачные вычисления, распределенные файловые системы, OpenStack, Ceph, данные персональной медицины

*Keywords:* cloud computing, distributed file systems, OpenStack, Ceph, personal medical data

### Введение

В настоящее время создание инфраструктуры компьютерной системы для реализации облачных сервисов хранения и обработки медицинских данных является весьма актуальной задачей, она обусловлена следующими факторами:

- одним из основных источников информации о состоянии здоровья пациента для врача является использование персональных медицинских данных, поэтому желательно иметь возможность длительного хранения и быстрого доступа к этой информации в любой момент времени;
- широкое распространение современного медицинского оборудования для диагностики состояния здоровья пациентов позволяет получать персональные медицинские данные, которые зачастую не хранятся в цифровом виде либо хранятся только на том компьютере, где проводилось медицинское исследование без доступа к этой информации из Интернета;
- активное развитие методов компьютерного зрения и машинного обучения позволяет проводить интеллектуальный анализ медицинских изображений, реализованный в виде облачного сервиса, а также создать эффективную систему принятия решения для диагностики состояния здоровья пациента;
- повсеместное проникновение быстрого подключения к Интернету и предоставление персональных услуг в виде облачных сервисов позволяет при разработке соответствующих программных продуктов получить возможность быстрого доступа в любой момент времени к неограниченному объему персональных медицинских данных конкретного пациента.

Все перечисленные факторы делают важным перспективную разработку инфраструктуры частного облака [1, 2], в котором будут накапливаться персональные медицинские данные пациентов. Врач может производить анализ этих данных с использованием интеллектуальных методов их обработки. Это избавит пациентов от постоянного личного хранения этих данных и их утери, так как медицинские данные любого объема будут храниться в облаке и в любой момент времени в любом месте эти данные могут быть доступны через сеть Интернет [3].



Так же становится возможным на основе персональных медицинских данных хранящихся в облаке выполнить анализ [4, 5] этих данных с использованием специальных методов, алгоритмов и программного обеспечения для извлечения знаний и интеграции этих полученных результатов в архитектуру облачной системы хранения данных. Все это позволит создать эффективную интеллектуальную систему диагностики и прогнозирования состояния здоровья пациентов в виде облачного сервиса.

### Архитектура хранилища данных

Традиционно, все медицинские изображения хранятся только определенный промежуток времени в управляющем компьютере рентгеновского аппарата, томографа. Данный способ прост и доступен всем, но не лишен ряда существенных недостатков – низкая масштабируемость и отказоустойчивость, кроме этого существенны риски потерять информацию. Однако есть и достоинство – консолидация (близость данных) и простота доступа.

Для сохранения персональных медицинских данных в облаке необходимо создать соответствующую инфраструктуру вычислительной компьютерной системы, а для эффективной реализации облачных сервисов обработки данных эта система должна быть кроме того и высокопроизводительной, то есть обладать возможностью выполнять на ней параллельные вычисления.

В настоящее время существуют следующие платформы для построения облачных компьютерных систем OpenStack, CloudStack и OpenNebula, Eucalyptus, OpenShift, oVirt [2].

Эти платформы позволяют строить облачные окружения различных типов от «инфраструктура как сервис» (infrastructure as a service) до «база данных как сервис» (database as a service). Такое гранулирование позволяет при желании запускать и использовать готовую инфраструктуру без установки и конфигурировании гипервизоров и гостевых операционных систем.

Нами предлагается для построения инфраструктуры высокопроизводительной компьютерной системы для реализации облачных сервисов хранения и анализа данных персональной медицины использовать OpenStack [6]. На сегодня это наиболее динамично развивающийся программный комплекс с поддержкой крупными компаниями. Кроме того, он реализует все \*aaS [2] типы предоставления окружения.

Облачная среда нуждается в хранилищах [2], которые могут масштабироваться и при этом иметь низкую стоимость, а также они могут быть легко интегрированы с другими компонентами структуры данной облачной среды. Существует ряд традиционных производителей систем хранения, которые утверждают, что обеспечивают интеграцию в рамках облаков, однако сегодня нам нужны дополнительные функции помимо простой поддержки интеграции. Эти традиционные решения для хранения данных, возможно, казались успешными несколько лет назад, но в настоящее время они не являются хорошим кандидатом на унифицированное решение хранения для облака. Кроме того, традиционные системы хранения стоят слишком дорого для развертывания и поддержки в долгосрочной перспективе, а масштабирование вверх и наружу является для них серой зоной.

Сегодня необходимо найти решение для хранения данных, которое было бы полностью ориентировано на выполнение текущих и будущих потребностей системы. Это решение должно быть создано на основе программного обеспечения с открытым исходным кодом и техническими средствами, которые могут обеспечить требуемую масштабируемость экономически эффективным способом.

Наиболее удобным подходом, с нашей точки зрения, является использование распределенной файловой системы в облачном окружении. основополагающие факторы, определяющие ключевую роль в выборе этой технологии для решения поставленной задачи – это масштабируемость, возможность балансировки нагрузки и надежность, что позволяет системе эффективно добавлять новые вычислительные узлы в случае увеличения нагрузки.

Существует несколько программных продуктов для реализации распределенных файловых систем. Нами предлагается использовать распределенную систему хранения Ceph [7]. Ceph – свободная к использованию распределенная файловая система, которая может использоваться на системах, состоящих как из нескольких машин, так и из тысяч узлов. Общий объем хранилища данных может измеряться петабайтами, встроенные механизмы продублированной репликации данных обеспечивают чрезвычайно высокую живучесть системы.

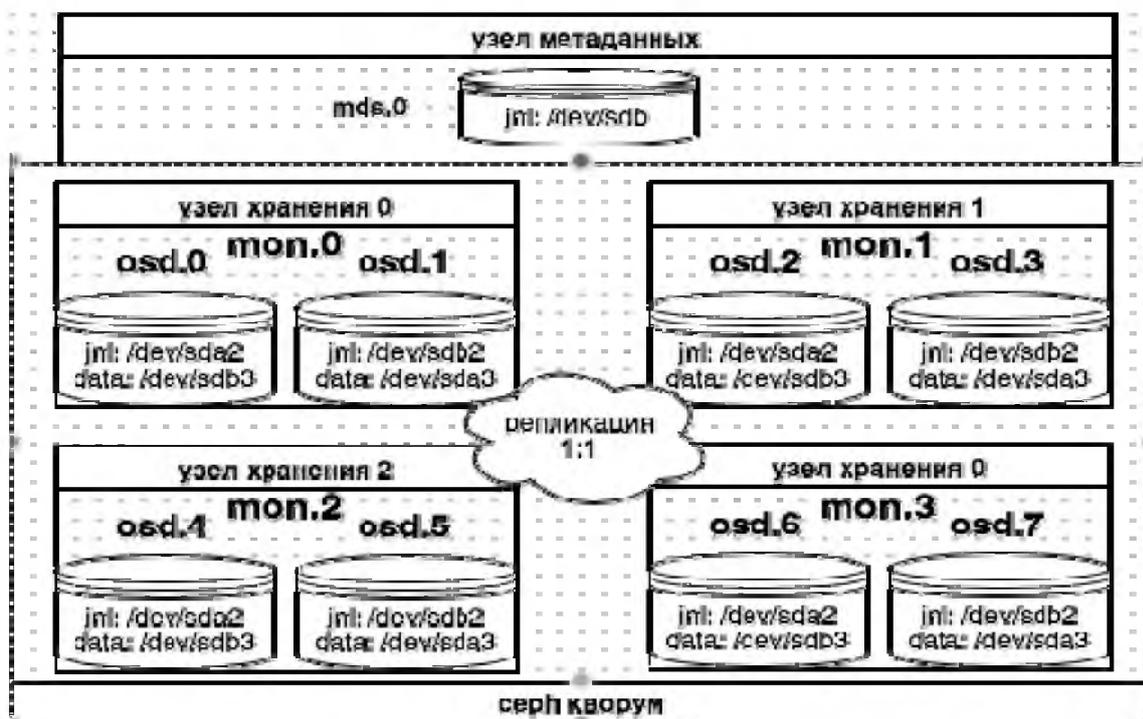
Основных типов сервисов в Ceph два – мониторы (MON, Ceph monitor) и сервисы хранения (OSD, Object Storage Device). MON – элемент инфраструктуры Ceph, который обеспечивает адресацию данных внутри кластера и хранит информацию о топологии, состоянии и распределении данных внутри хранилища. OSD – сущность, которая отвечает за хранение данных, основной строительный элемент кластера Ceph. На одном физическом сервере может размещаться несколько OSD, каждая из которых имеет под собой отдельное физическое хранилище данных.

### Прототип компьютерной системы для реализации облачных сервисов

Нами создан прототип высокопроизводительной компьютерной системы для реализации облачных сервисов [8], который развернут на базе четырех физических узлов и одной виртуальной машины. Каждый физический узел оснащен двумя жесткими дисками.

Было принято решение разметить каждый диск в двух логических томах. Первые тома объединить в RAID1 (для надежности туда устанавливалась операционная система узла). Вторые же тома выступали в роли физических томов для LVM (Logical Volume Manager — менеджер логических томов). В процессе работы Ceph доступ к метаданным осуществляется достаточно часто, поэтому расположение их на одном диске с самими данными может негативно сказаться на производительности. Исходя из-за этого, для OSD на каждом узле были созданы четыре логических LVM тома по два на один OSD так, что том для метаданных и для данных были разделены по независимым жестким дискам.

Разработанная инфраструктура компьютерной системы для реализации облачных сервисов хранения данных приведена на рисунке.



Инфраструктура компьютерной системы хранения данных  
Storage service infrastructure

Получившийся кворум в сумме обладает восемью OSD. Возможная и удобная схема дублирования – 1+1, как RAID1, кроме того, данная конфигурация обладает высокой устойчивостью – при выпадении одного/двух дисков есть вероятность восстановления данных, так же не станет проблемой и выпадение одного из мониторов – мониторы обладают механизмом перевыбора мастера из оставшихся в строю.

В разработанную нами архитектуру был добавлен слой абстракции для доступа к данным. В основе Ceph лежит блочное хранение объектов, где каждый объект идентифицируется уникальным именем в пределах сконфигурированного пула. Из доступных инструментов для работы в Ceph имеется RADOS (Reliable Autonomic Distributed Object Store) Gateway, являющийся абстракцией объектного хранилища, но требующий дополнительного конфигурирования и слежения за исправностью (RADOS, Reliable Autonomic Distributed Object Store) его работы.

Однако мы решили, что наиболее подходящим вариантом является разработка собственного REST (Representational state transfer) - интерфейса с использованием прямого доступа к мониторам. Положительными сторонами такого подхода являются, кроме отсутствия лишнего администрируемого слоя, возможности самостоятельного выбора пула, а также организации логики выбора этого пула. Например, оперативные данные можно хранить в пуле без реплики, что обеспечит прирост в скорости, а важные данные – уже в другом пуле, в котором описана crush-карта. Разработанный интерфейс обеспечивает все необходимые CRUD-действия (create-read-update-delete).



## Включение Ceph в состав OpenStack

Общедоступные и частные облака получили широкое распространение благодаря реализации проекта OpenStack[3]. OpenStack – облачное решение полного цикла. Он имеет свои внутренние компоненты хранения с названием Swift для хранилища облака и Nova-Volume, который поддерживает тома блочного хранилища для виртуальных машин.

В отличие от Swift, который ограничивается только хранением объектов, Ceph является универсальным решением для хранения блочных данных, файлов, а также объектов, и, таким образом приносит пользу OpenStack, предоставляя различные типы носителей в одном кластере хранения. Таким образом, можно легко и эффективно управлять хранением облака OpenStack. Сообщества OpenStack и Ceph сотрудничают в течение многих лет в разработке полностью поддерживаемой облаком OpenStack системы хранения данных Ceph.

Ceph полностью интегрирован с шестой основной редакцией OpenStack, что способствует появлению новых функций, а также корректировке ошибок. OpenStack использует одно из самых требовательных свойств Ceph, блочное устройство RADOS (*RBD, RADOS block device*) с применением своих компонентов cinder и glance.

По нашему мнению, облачные платформы с Ceph в качестве механизма хранения данных обеспечивают гибкость для поставщиков услуг при построении решений системы хранения – как – службы и инфраструктуры – как – службы (IaaS), которые они не способны получить от других традиционных решений корпоративного уровня, поскольку они не предназначены для удовлетворения потребностей облаков. Используя Ceph в качестве механизма для платформ облаков, можно предложить недорогие облачные службы для своих потребителей.

Dell, SUSE и Canonical предлагают и предоставляют инструменты поддержки настройки и управления для автоматизированного и легкого развертывания систем хранения Ceph для корпоративных облачных решений OpenStack. Каждый из этих инструментов имеет открытый код, а также готовые модули Ceph, которые могут быть легко использованы для развертывания Ceph. В распределенной среде, такой как облачная, каждый компонент должен масштабироваться. Эти инструменты управления настройкой важны для быстрого наращивания масштабов инфраструктуры.

Созданный нами прототип высокопроизводительной компьютерной системы для реализации облачных сервисов имеет следующие технические характеристики:

- пять узлов, один из которых – с конфигуратором и MDS-демоном для поддержки cephfs и rados gateway, представляет из себя виртуальную машину, имеющую 4 процессорных ядра, 4 ГБ ОЗУ и 30 ГБ жесткого диска;

- в системе развернуты веб-сервер и сервер СУБД для пользовательского интерфейса с возможностью использования контроллера облачного окружения.

- узлы хранения (гипервизоры) представляют собой 4 идентичные аппаратные платформы, имеющие 8 процессорных ядер, 4 ГБ ОЗУ, 2 винчестера по 0,5 ТБ памяти частично в программном рейде 1, 2 ГБ сетевых интерфейса. Они используются в роли OSD-сервисов Ceph и Nova OpenStack.

## Интерфейс пользовательского доступа

Для обеспечения доступа к данным нами разработано фронтенд-приложение. В качестве языка программирования выбран объектно-ориентированный язык Java, который давно и устойчиво зарекомендовал себя как один из лучших вариантов выбора для кроссплатформенных enterprise-приложений.

В качестве фреймворка для разработки фронтенда был выбран Play Framework[9] – каркас разработки с открытым кодом, написанный на Scala и Java, использует паттерн проектирования Model-View-Controller (MVC).

В качестве хранилища данных о пользователях, системной информации, выбрана СУБД PostgreSQL – свободная объектно-реляционная система управления базами данных. PostgreSQL является наиболее эффективным решением для высоконагруженных, распределенных сервисов. Выбор технологий Java и Play Framework позволяет использовать JPA (Java Persistence API) – API, предоставляющий возможность сохранять в удобном виде Java-объекты в базе данных. Наиболее распространенной реализацией JPA-интерфейса является Hibernate [10] – библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения данных.

Для того чтобы организовать хранение данных в БД, необходимо описать, с использованием Java-аннотаций, сущности и их связи. Среди сущностей можно выделить основные:

User – описывает клиента как единичного пользователя системы. Связана с User Information связью 1:1 и User Data связью 1:N;

User Information – содержит основную информацию о клиенте как пациенте;



User Data – содержит уникальное поле для однозначной идентификации в пуле распределенного хранилища. Связана с User Data Type связью 1:1 и User Data Meta связью 1:N;

User Data Type – описывает типы медицинских данных;

User Data Meta – описывает всю необходимую дополнительную, а также системную информацию о данных.

Разработанное клиент-серверное приложение позволяет при обращении пользователя к веб-интерфейсу, после завершения процедуры регистрации и заполнения сводной информации, получить возможность загружать, хранить, скачивать и удалять персональную медицинскую информацию, в том числе – изображения с рентгеновских аппаратов и томографов. Кроме этого, разработанное приложение позволяет в дальнейшем добавлять в систему облачные сервисы, выполняющие анализ данных персональной медицины с использованием интеллектуальных методов их обработки. При этом кроме надежного хранения информации в распределенном хранилище, информация доступна из любой точки планеты через Интернет.

### Заключение

В данной статье рассмотрен способ построения инфраструктуры высокопроизводительной компьютерной системы для реализации облачных сервисов хранения и анализа данных персональной медицины. Благодаря наличию хранилища в частном облаке имеется возможность накапливать там персональные медицинские данные и по желанию врача или клиента производить их анализ с использованием интеллектуальных методов. Это избавит пользователя от постоянного личного хранения этих данных: они будут храниться в облаке и поэтому в любой момент времени и в любом месте эти данные могут быть доступны через Интернет. Так же становится возможным на основе персональных медицинских данных, хранящихся в облаке, выполнить их анализ с использованием специальных методов для извлечения знаний и интеграции полученных результатов в архитектуру облачной системы хранения с возможностью реализации интеллектуальных систем прогнозирования и диагностики в виде облачного сервиса.

**Работа выполнена при финансовой поддержке РФФИ (код проекта 16-07-00435 А)**

### Список литературы References

1. Михелев М.В., Кузнецов К.В., Михелев В.М. 2012. Построение информационной инфраструктуры высокопроизводительной компьютерной системы для реализации облачных вычислений. Журнал Вопросы радиоэлектроники Серия Электронная вычислительная техника. 4 (1): 12–20.  
Mihelev M.V., Kuznecov K.V., Mihelev V.M. 2012. Postroenie informacionnoj infrastruktury vysokoproizvoditel'noj komp'yuternoj sistemy dlja realizacii oblachnyh vychislenij. Zhurnal Voprosy radiojelektroniki Serija Jelektronnaja vychislitel'naja tehnika. 4 (1): 12–20.
2. Рииз Дж. 2011. Облачные вычисления: Пер. с англ. СПб. БХВ-Петербург: 288.  
Riiz Dzh. 2011. Oblachnye vychislenija: Per. s angl. SPb. BHV-Peterburg: 288.
3. Foster, Ian; Carl Kesselman. 2004. The Grid: Blueprint for a New Computing Infrastructure Second Edition. San Francisco. Morgan Kaufmann Publishers. 748.
4. Торопчин Д.А., Батищев Д.С., Медведев А.С., Михелев В.М. 2014. Архитектура программно-аппаратной поддержки распознавания нечетких дубликатов изображений. Журнал Вопросы радиоэлектроники. Серия Электронная вычислительная техника, 4 (2): 78–88.  
Toropchin D.A., Batishhev D.S., Medvedev A.S., Mihelev V.M. 2014. Arhitektura programmno-apparatnoj podderzhki raspoznavanija nechetkih dublikatov izobrazhenij. Zhurnal Voprosy radiojelektroniki. Serija Jelektronnaja vychislitel'naja tehnika, 4 (2): 78–88.
5. Кузнецов К.В., Михелев В.М., Торопчин Д.А., Петров Д.В., Липницкий А.Г. 2013. Архитектура программно-аппаратной поддержки высокопроизводительного поиска изображений по визуальному подобию. Журнал Вопросы радиоэлектроники Серия Электронная вычислительная техника. 4 (1): 10–17.  
Kuznecov K.V., Mihelev V.M., Toropchin D.A., Petrov D.V., Lipnickij A.G. 2013. Arhitektura programmno-apparatnoj podderzhki vysokoproizvoditel'nogo poiska izobrazhenij po vizual'nomu podobiju. Zhurnal Voprosy radiojelektroniki Serija Jelektronnaja vychislitel'naja tehnika. 4 (1): 10–17.
6. OpenStack Operations Guide. Available at: <http://docs.openstack.org/openstack-ops/content/>
7. Welcome to Ceph – Ceph Documentation. Available at: <http://docs.ceph.com/docs/master/>
8. Батищев Д.С., Михелев В.М. 2014. Инфраструктура облачной системы для обработки медицинских данных. В сборнике: Наукоемкие технологии и инновации Белгородский государственный технологический университет им. В.Г. Шухова. Белгород: 8–12.  
Batishhev D.S., Mihelev V.M. 2014. Infrastruktura oblachnoj sistemy dlja obrabotki medicinskih dannyh. V sbornike: Naukoemkie tehnologii i innovacii Belgorodskij gosudarstvennyj tehnologicheskij universitet im. V.G. Shuhova. Belgorod: 8–12.
9. Documentation - Play Framework. Available at: <https://www.playframework.com/documentation/1.4.x/home>
10. Hibernate ORM documentation (5.0) - Hibernate ORM. Available at: <http://hibernate.org/orm/documentation/5.0/>