МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИЙ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Рыбинский государственный авиационный технический университет   
имени «П.А. СОЛОВЬЕВА»

Факультет радиоэлектроники и информатики

Кафедра математического и программного обеспечения электронных вычислительных средств

Выпускная квалификационная работа

На тему

«Разработка быстроразвёртвываемого объектного хранилища»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Студент группы ИПБ-17 Ащеулов М.Р.

Руководитель Задорина Н.А.

Рыбинск 2021

**Содержание**

[Введение 2](#_Toc72934653)

[1. Описание проблемы 5](#_Toc72934654)

[2. Обзор аналогов 8](#_Toc72934655)

[3. Программная документация 10](#_Toc72934656)

[**a.** **Техническое задание на программное обеспечение** 11](#_Toc72934657)

[**b.** **Пояснительная записка к программному обеспечению** 14](#_Toc72934658)

[**c.** **Описание программы** 15](#_Toc72934659)

[**d.** **Программа и методика испытаний** 17](#_Toc72934660)

[4. Эксплуатационная документация на программный продукт 19](#_Toc72934661)

[5. Акт испытаний программного продукта 20](#_Toc72934662)

[6. Экономическое обоснование 21](#_Toc72934663)

[7. Заключение 22](#_Toc72934664)

[8. Список литературы 23](#_Toc72934665)

[9. Приложения 24](#_Toc72934666)

# Введение

С каждым годом, количество цифровых данных неумолимо растёт. Фотографии, видео, документы пользователи предпочитают хранить в электронном формате в виде файлов. В 2020 году во всём мире было создано и скопировано 59 зетабайта или 59 триллионов гигабайтов данных, из которых большую часть, а именно 44 зетабайт, составляют неструктурированные данные (тексты, изображения, видео, аудио и пр.), то есть уже сейчас 75% всей цифровой информации – это неструктурированные данные.

Неструктурированные данные – это данные, которые либо не имеет заранее определенной структуры, либо не организованы в установленном порядке. Неструктурированные данные представлены, как правило, в виде текстовых документов, электронных писем и презентаций. Примерами не текстовых неструктурированных данных являются видео, изображения и аудиофайлы.

На сегодняшний день существует механизм хранения данных, который используется с момента появления компьютеров, − файловая система. Однако существует ряд ограничений, связанных с работой с неструктурированными данными. Во-первых, используемая для хранения данных система должна хорошо масштабироваться. Масштабировать файловую систему - сложная задача: требуется не только управлять ненужными метаданными и иерархией, которые навязывают файловые системы, но также необходимо учитывать такие аспекты обслуживания, как управление резервными копиями. Во-вторых, недостаточно просто собрать и хранить неструктурированные данные, необходимо также применить некоторый уровень организации, чтобы эффективно с ними работать. Такие методы, как анализ текста, автоматическая категоризация и автоматическая пометка, имеют решающее значение для получения бизнес-смысла для неструктурированных данных. Этого сложно добиться с файловыми системами, потому что они имеют фиксированные слои (или расположение). Для решения этих проблем появилась концепция объектного хранилища.

Важным аспектом любых систем хранения является обработка метаданных. Объектное хранилище обеспечивает большую гибкость, поскольку метаданные объекта изначально не определены. Метаданные не ограничиваются тем, что система хранения считает важным. Имеется возможность добавить любой тип или количество метаданных. Объектное хранилище позволяет назначить тип приложения, с которым связан объект; важность для приложения; уровень защиты данных, который требуется конкретному объекту.

Объектное хранилище обладает следующими свойствами:

* Масштабируемость. Объектное хранилище может содержать практически любое количество данных без необходимости в разбиении набора данных на разделы.
* Эффективность. Отсутствие иерархии означает отсутствие узких мест, возникающих вследствие использования сложных систем каталогов.
* Доступность. Объектные системы хранения имеют механизмы для сохранения целостности данных, обеспечивают репликацию данных, последовательные обновления и отсутствие простоев.

На IT рынке уже существует немало облачных объектных хранилищ от крупных IT компаний. Например, Google cloud storage, Amazon s3, IBM object storage, отечественные решения от Mail.ru и Yandex. Их объединяет одна важная деталь: размещение данных в облаке собственных серверах компании. Вследствие этого все они предоставляют услуги по использованию места на их серверах за определённую плату. Плата взимается за каждый запрос, а так же за само хранение данных. Еще одним важным недостатком является то, что клиент не знает, где и как хранятся его данные. Кроме того, компания, предоставляющая услугу, имеет доступ к конфиденциальной информации пользователей. Не смотря на обещания крупных IT компаний в безопасности на 99,9999999%, по данным InfoWatch только за прошлый год в сеть утекло более 14 миллиардов конфиденциальных записей. Поэтому многие потенциальные клиенты, боясь за очередную утечку, хотят найти решение, которое позволит им сохранить свои данные в целости и безопасности. Так же на некоторых предприятиях требования информационной безопасности или внутреннего регламента обязывает сотрудников или само предприятие использовать отечественные разработки и решения, которые могу функционировать без доступа в сеть интернет.

В современном мире каждый человек где-то хранит персональные данные: в облаке, на флеш-накопителе или жестком диске компьютера. Сейчас, в эпоху развития компьютерных технологий, практически в любом виде деятельности необходимо передавать данные другим людям или машинам и хранить эти данные. Системы хранения и передачи информации должны быть удобны как для пользователя, так и для обслуживающего персонала или машины.

Возможны два варианта решения этой задачи:

* Самый безопасный: передать данные лично на USB-флеш-накопителе.
* Самый простой: разместить файлы в облачном хранилище, предлагаемом IT гигантами

Однако существует альтернативное решение - это быстроразвёртываемое объектное хранилище, которое запускается на любом компьютере и обеспечивает удобство облачных систем хранения и безопасность физических накопителей. Для работы в хранилищем необходимо просто запустить исполняемый файл и указать путь до места, где необходимо развернуть объектное хранилище. Пользовательское взаимодействие с объектным хранилищем выполняется через специальный интерфейс. Интерфейс же в свою очередь взаимодействует с объектным хранилищем посредством API.

Таким образом, быстроразвёртывыаемое объектное хранилище является перспективным механизмом организации хранения неструктурированных данных. Предлагаемый механизм обеспечивает удобство работы с неструктурированными данными, а также безопасность и надежность их хранения.

# Описание проблемы

Данные существуют во множестве различных форм и размеров, но большинство из них могут быть представлены в виде структурированных и неструктурированных данных.

|  |  |  |
| --- | --- | --- |
|  | Структурированные данные | Неструктурированные данные |
| Организация | Структурированные данные представляют собой высокоорганизованную, фактическую (важную) и точную информацию | Неструктурированные данные не имеют заранее определённой структуры и представлены во всём многообразии форм |
| Характеристика данных | Количественная | Качественная |
| Место размещения | Хранилища данных  Реляционные базы данных | Пул данных  Нереляционные базы данных |
| Формат и модель | Несколько предопределённых форматов и моделей данных | Формат и модель данных заранее не определены |

Таблица 1.1 - Сравнение структурированных и неструктурированных данных

Структурированные данные – это хорошо организованные и точно отформатированные данные. Эти данные существуют в формате реляционных баз данных, то есть информация хранится в таблицах со связанными строками и столбцами. Таким образом структурированные данные аккуратно упорядочиваются и записываются, поэтому их можно легко найти и обработать. Пока данные вписываются в структуру базы данных, мы можем легко искать конкретную информацию и выделять отношения между ее частями. Такие данные можно использовать только по прямому назначению. Кроме того, для структурированных данных обычно не требуется много места для хранения.

В аналитических целях можно использовать хранилища данных . Data Warehouse – это центральные хранилища данных, используемые компаниями для анализа данных и составления отчетов.

Неструктурированные данные – данные, которые не соответствуют заранее определённой модели данных и не имеют легко идентифицируемой структуры. Вследствие этого их нелегко использовать в компьютерных программах. Неструктурированные данные не организованны заранее определённым образом или не имеют заранее определённой модели данных, что делает их неудобными для размещения в реляционной базе данных. Сегодня крупнейшие и не только предприятия осознают ценность и важность данных на всех этапах деятельности и жизненного цикла. В различных областях – в маркетинге, в медицине и здравоохранении, в медиа-сервисах и т.д. – неструктурированные данные крайне ценны тем, что находятся новые способы их анализа, изменения и форматирования. К примеру:

1. Медицинские учреждения применяют различные методы анализа архивных данных для прогнозирования хода болезни
2. Автоконцерны исследуют особые данные с датчиков автомобилей, чтобы предугадать вероятные неисправности
3. Телекомпании повторно используют передачи, фильмы, сериалы для показа. Чтобы их повторно не обрабатывать под требуемый формат их необходимо хранить в надёжном месте

Но ценность такие данные имеют только при условии экономного хранения, удобного доступа и безопасности от «утечек».

Уже существует механизм хранения, который люди используют с появления современных компьютеров − файловая система. Но когда речь идёт о неструктурированных данных, важно понимать, что система, используемая для хранения данных, должна очень хорошо масштабироваться. Однако масштабировать файловую систему это сложная задача. Вам требуется не только управлять ненужными метаданными и иерархией, которые навязывают файловые системы, но также необходимо учитывать такие аспекты обслуживания, как управление резервными копиями.

Недостаточно просто собрать и хранить неструктурированные данные. Также необходимо применить некоторый уровень организации, чтобы разобраться в них. Такие методы, как анализ текста, автоматическая категоризация и автоматическая пометка, имеют решающее значение для получения бизнес-смысла для всех неструктурированных данных, которые вы собираете. Этого сложно добиться с файловыми системами, потому что они имеют фиксированные слои или расположение.

Для решения этой проблемы появились объектные хранилища.

Объектное хранилище представляет собой стратегию управления и манипуляции хранилищем данных как отдельными единицами, объектами. Эти объекты хранятся в одном хранилище и не интегрируются в файлы, находящиеся в других папках. Вместо этого хранилище объектов объединяет фрагменты данных, из которых состоит файл, добавляет в него все соответствующие метаданные и прикрепляет пользовательский идентификатор.

Объектное хранилище добавляет в файл полные метаданные, устраняя многоуровневую структуру файлового хранилища, и помещает все в плоское адресное пространство — пул хранилища. Именно метаданные являются ключом к успеху объектного хранилища: они обеспечивают глубокий анализ использования и функций данных в пуле.

Очень важно, чтобы данные были доступны через HTTP(S), чтобы обеспечить лёгкий доступ к файлу. Затем его можно подвергнуть анализу или другим методам. Объектное хранилище справляется с этим хорошо. Почти все объектные хранилища имеют REST API, которое позволяет крайне просто получить доступ к файлам через HTTP(S).

REST API полезно не только для получения доступа к объектам, оно позволяет авторизоваться и получать токен сессии, получать свойства файла, управлять разрешениями и т.д., в общем всё, что необходимо будет делать вручную в файловой системе.

Изображение

Рисунок 1.2 - Объект

Важным аспектом любых систем хранения является обработка метаданных. Объектное хранилище обеспечивает большую гибкость, поскольку метаданные объекта изначально не определены. Метаданные не ограничиваются тем, что система хранения считает важным. Имеется возможность добавить любой тип или количество метаданных. Объектное хранилище позволяет назначить тип приложения, с которым связан объект; важность для приложения; уровень защиты данных, который требуется конкретному объекту. И так далее, возможности безграничны.

# Обзор аналогов

Несмотря на то, что реализаций объектных хранилищ существует огромное множество, почти все они являются облачным решением и хранят персональные данные пользователей на собственных серверах. Быстроразвёртываемое объектное хранилище обладает уникальной особенностью – запуск и работа на локальной машине или устройстве, что позволяет вам настроить его в локальной сети без доступа к интернету и ограничить возможный несанкционированный доступ извне. Так же важным преимуществом является цена, а точнее ценообразование. Если при использовании облачных решений вы платите за обслуживание, пользование и хранение данных, то в случае с локальным быстроразвёртываемым хранилищем вы платите исключительно за «железо» или выбранному Вами провайдеру по утверждённому тарифу. Для наглядности я приведу таблицу сравнения цены пользования облачным решением от некоторых компаний за год и стоимость системы с накопителями того же объёма.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Amason s3 | MinIO | Быстроразвёртываемое объектное хранилище |
| Объём накопителей | 8ТБ | 8ТБ | 8ТБ |
| Количество GET запросов | 10000 | Не ограничено | Не ограничено |
| Количество PUT, COPE, POST запросов | 100 | Не ограничено | Не ограничено |
| Обслуживание | Включено в стоимость | Ручное (1 человек) | Ручное (1 человек) |
| Размещение | В облаке | В облаке или локально | В облаке и/или локально |
| Цена | 2329.8 USD/год | 960 USD/год за облачное решение и 500 USD за комплект оборудования | 500 USD за комплект оборудования  Или облачное размещение по цене тарифа провайдера, у которого будет размещено |

Таблица 2.1 - Сравнение с аналогами

Из таблицы видно, что Быстроразвёртываемое объектное хранилище выигрывает в цене в долгосрочной перспективе и возможностях конфигурации системы: всё зависит от вашего бюджета и требований. А за счёт простоты настройки и использования с обслуживанием справится обычный пользователь.

Для некоторых пользователей (клиентов) страна разработки является важным фактором при выборе программного продукта. Так как существуют ограничения (постановление Правительства Российской Федерации от 16 ноября 2015 г. №1236) на использование иностранного программного обеспечения в определённых структурах и предприятиях. Так же для обеспечения дополнительного уровня безопасности от возможного вмешательства или утечки требуется использовать программное обеспечение, которое не требует и может функционировать без доступа в интернет.

# Программная документация

Текст

## **Техническое задание на программное обеспечение**

1. Общие сведения
   1. Наименование программы  
      Быстроразвёртываемое объектное хранилище
   2. Наименование разработчика и заказчика системы  
      Разработчик: Ащеулов М. Р.  
      Заказчик: Ащеулов М. Р.
   3. Основание для проведения работ

Документ, на основании которого создается подсистема: приказ № 550-04, утвержденный ректором РГАТУ им. П.А. Соловьева от 30.11.2020.

* 1. Область применения программы

Программа предназначена для персонального и корпоративного использования с целью организации, хранения, а так же получения и использования неструктурированных данных.

1. Назначение разработки

Быстроразвёртывыаемое объектное хранилище является перспективным механизмом организации хранения неструктурированных данных. Предлагаемый механизм обеспечивает удобство работы с неструктурированными данными, а также безопасность и надежность их хранения.

1. Требования к программному продукту или программному обеспечению
   1. Требования к функциональным характеристикам

Программный продукт должен обеспечивать:

* Загрузку одного объекта в объектное хранилище;
* Загрузку нескольких объектов последовательно;
* Получение объекта из объектного хранилища;
* Добавление параметра к метаданным заданного объекта;
* Удаление параметра из метаданных заданного объекта;
* Установление новых метаданных для заданного объекта;
* Удаление всех метаданных для заданного объекта;
* Получение всех метаданных объекта;
* Получение одного параметра метаданных объекта по ключу;
* Аутентификацию существующего пользователя;
* Регистрацию (добавление) нового пользователя;
* Авторизацию пользователей.
  1. Требования к надёжности

Надежность должна обеспечиваться за счет использования сертифицированных технических средств системного и базового ПО, комплектующих.

С целью повышения отказоустойчивости системы может быть использован источник бесперебойного питания с возможностью автономной работы системы и активного сетевого оборудования.

Для стабильной работы системы и обеспечения быстрого доступа к программному обеспечению необходимо устойчивое сетевое соединение либо в локальной сети, либо с глобальной сетью.

* 1. Требования к составу и параметрам технических средств
     1. Сервер или ПК который используется для запуска приложения

Требования к аппаратной части:

* Процессор с совокупным количеством ядер не менее 2
* Оперативная память: не менее 2 гб
* Подсистема хранения данных системы, ПО, конфигурационных файлов: не менее 100 гб
* Сетевой адаптер с максимальной пропускной скоростью не менее 10 Мбит/сек

Требования к каналам связи:

* Гарантированная пропускная способность не менее 10 Мбит/сек
* Доступ к сетевому адресу сервера из сети интернет для работы за пределами локальной сети

Требования к программному обеспечению:

* Операционная система: Ubuntu 20.04 и выше, Fedora 36 и выше, MacOS 11.0 и выше, Windows 10 1709 и выше.
  + 1. Клиентская часть

Требования к аппаратной части:

* Процессор: 2 ядра, x86-64 с частотой 1.3 Ггц и выше
* Оперативная память: не менее 2 гб
* Свободное дисковое пространство: не менее 100 мб

Требования к каналам связи между клиентским устройством и сервером:

* Скорость передачи данных: не менее 512 Кбит/сек

Требования к программному обеспечению

* Операционная система: Ubuntu 20.04 и выше, Fedora 36 и выше, MacOS 11.0 и выше, Windows 10 1709 и выше.
* Браузер на основе движка Chromium (Google Chrome, Яндекс Браузер, Edge 89.0.774.50 и т.д.)
  1. Специальные требования

Специальные требования к программному продукту не предъявляются

1. Требования к программной документации

* Пояснительная записка к программному обеспечению
* Описание программы
* Программа и методика испытаний

## **Пояснительная записка к программному обеспечению**

Текст

## **Описание программы**

Быстроразвёртываемое объектное хранилище является программным продуктом, который предоставляет пользователю (клиенту) запустить локальное объектное хранилище на собственном компьютере или сервере. Взаимодействие с быстроразвёртываемым объектным хранилищем осуществляется посредством API.

Для взаимодействия с разрабатываемым программным обеспечением может быть разработан и использован графический интерфейс, который обращается к серверу и предоставляет пользователю возможность работать с программным продуктом через браузер или десктоп-клиент.

Для начала работы с Быстроразвёртываемым объектным хранилищем необходимо запустить исполняемый файл через консоль или двойным кликом. Далее появляется диалоговое окно, которое предлагает запустить уже существующее объектное хранилище или инициализировать новое.

При инициализации нового система просит ввести данные учётной записи для администратора объектного хранилища. Введённые данные проверяются на корректность и соответствие установленным требованиям безопасности (длина пароля, наличие специальных символов, разные регистры символов, наличие цифр и т.д.). После успешной регистрации пользователя-администратора открывается диалог выбора корневой директории объектного хранилища или место, где будет оно инициализировано. При выборе файла вместо папки корневая директория будет местом расположения файла и файл автоматически проиндексируется и добавится как объект. При выборе папки с файлами или другими папками, объектное хранилище инициализируется и индексирует всё как отдельные объекты. Т.е. и папки, и файлы станут отдельными объектами с идентификатором соответствующим имени. После успешного завершения этапа инициализации сервер запускается, и пользователь может начать работу с программным обеспечением.

При выборе пункта «Запустить уже существующее объектное хранилище» открывается диалог выбора расположения. Если пользователь выбирает директорию, в которой уже инициализировано объектное хранилище, то программа запрашивает у пользователя логин и пароль для учётной записи администратора этого объектного хранилища. После успешной верификации пользователя запускается сервер объектного хранилища, и пользователь может начать работу с программным обеспечением.

Важным аспектом взаимодействия с программой является то, как организована работа с метаданными. Изначально метаданные объекта при инициализации не определены. По умолчанию доступ к получению, модификации, удалению и прочим операциям над объектом разрешены только пользователю-администратору после получения токена сессии (session). Чтобы разрешить публичный или общий доступ к объекту необходимо добавить параметр “access”:”all” к метаданным объекта. Ключ “access” обозначает то, кто может иметь доступ к объекту. Если необходимо разрешить доступ конкретным пользователям, то необходимо в значении ключа ввести через пробел логины требуемых пользователей. Если необходимо разрешить доступ всем, то нужно указать значение “all”. Чтобы получить доступ к объекту с включённым значением “access”:”all” не требуется никаких дополнительных действий. Достаточно просто обратиться к объектному хранилищу через команду предоставив уникальный идентификатор объекта. Если объект имеет не публичный уровень доступа, то необходимо так же передать в заголовке запроса токен сессии, который может быть получен с помощью обращения к серверу объектного хранилища через /api/auth с заголовками “email” и “password”.

## **Программа и методика испытаний**

Для тестирования API будет использовано специализированное программное обеспечение Postman.

Программа должна корректно обрабатывать и отвечать на следующие запросы:

1. GET /api/get/{uid}  
   Получение объекта по уникальному идентификатору  
   uid - уникальный идентификатор объекта
2. POST /api/post -F file=”/path/to/file/”   
   Отправка(загрузка) объекта в объектное хранилище  
   -F file=”” файл(-ы), которые требуется загрузить
3. PUT /api/put/{uid} -F file=”/path/to/file”  
   Обновление объекта в объектном хранилище  
   uid - уникальный идентификатор объекта  
   -F file=”” файл, который будет загружен вместо предыдущего
4. DELETE /api/delete/{uid}   
   Удаление объекта из объектного хранилища  
   uid - уникальный идентификатор объекта
5. GET /api/auth -H “email=email” -H “password=password”  
   Аутентификация или получение токена  
   Передаются логин и пароль с помощью заголовков запроса
6. POST /api/obj/{uid}/set-meta --data ‘{“key”:”value”}’  
   Установление всех метаданных.  
   uid - уникальный идентификатор объекта  
   --data - тело POST запроса, которое содержит JSON с метаданными
7. POST /api/obj/{uid}/add-meta --data ‘{“key”:”value”}’  
   Добавление записи (параметра) к метаданным
8. GET /api/obj/{uid}/get-all-meta  
   Получить все метаданные объекта по уникальному идентификатору  
   uid - уникальный идентификатор объекта
9. GET /api/obj/{uid}/get-meta/key?={key}   
   Получение значения метаданных по ключу  
   uid - уникальный идентификатор объекта  
   key - ключ
10. POST /api/new-user -H email -H password  
    Регистрация нового пользователя

|  |  |  |
| --- | --- | --- |
| Номер запроса | Описание запроса | Ожидаемый результат |
| 1 | Получение объекта по уникальному идентификатору | Объект соответствующий уникальному идентификатору |
| 2 | Загрузка в объектное хранилище нового объекта | Объект загружен и получен ответ об успешном завершении операции |
| 3 | Обновление объекта в объектном хранилище | Старый объект с соответствующим идентификатором удалён, новый загружен |
| 4 | Удаление объекта в объектном хранилище | Требуемый объект удалён |
| 5 | Аутентификация и получение токена доступа | Токен доступа |
| 6 | Установление метаданных для объекта | Для объекта установлены метаданные |
| 7 | Добавление параметра к метаданным | Добавлен параметр в конец списка для соответствующего объекта |
| 8 | Получение всех метаданных | Все метаданные |
| 9 | Получение значения метаданных по ключу | Значение соответствующее определённому ключу |
| 10 | Регистрация нового пользователя | Пользователь зарегистрирован и получен ответ об успешном завершении операции |

# Эксплуатационная документация на программный продукт

Текст

# Акт испытаний программного продукта

Текст

# Экономическое обоснование

Текст

# Заключение

Текст

# Список литературы

Текст

# Приложения

Текст