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](#_Toc72241843)

[1. Описание проблемы 3](#_Toc72241844)

[2. Обзор аналогов 5](#_Toc72241845)

[3. Программная документация 6](#_Toc72241846)

[**a.** **Техническое задание на программное обеспечение** 7](#_Toc72241847)

[**b.** **Пояснительная записка к программному обеспечению** 8](#_Toc72241848)

[**c.** **Описание программы** 9](#_Toc72241849)

[**d.** **Программа и методика испытаний** 10](#_Toc72241850)

[4. Эксплуатационная документация на программный продукт 12](#_Toc72241851)

[5. Акт испытаний программного продукта 13](#_Toc72241852)

[6. Экономическое обоснование 14](#_Toc72241853)

[7. Заключение 15](#_Toc72241854)

[8. Список литературы 16](#_Toc72241855)

[9. Приложения 17](#_Toc72241856)

# Введение

С каждым годом, количество цифровых данных неумолимо растёт. Пользователи всё больше делают фотографий, записывают видео, создают документов, и т.д. Только в 2020 году во всём мире было создано и скопировано более 59 зетабайта или 59 триллионов гигабайтов данных. 44 из них являются неструктурированными.

Для решения проблемы организации и хранения такого большого количества неструктурированных данных появилась концепция объектного хранилища. В отличие от файлов в файловой системе, объекты хранятся в плоской структуре. Есть просто пул объектов: никаких папок, каталогов или иерархий. Для получения объекта необходимо просто сделать запрос к объектному хранилищу, предоставив идентификатор этого объекта.

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. Есть несколько вариантов решения этих проблем:

1. Самый безопасный, передать данные на флеш-накопителе или защищённом диске лично в руки получателю
2. Самый простой и удобный, разместить файлы в облачном хранилище, предлагаемом различными провайдерами

Быстроразвёртываемое объектное хранилище в свою очередь совмещает в себе безопасность физических накопителей и удобство использования облачных систем хранения.

Для того чтобы начать работу необходимо будет просто запустить приложение на локальном устройстве и указать путь до корневой директории, где будет развёрнуто объектное хранилище. Пользователь взаимодействует с объектным хранилищем через специальный интерфейс. Интерфейс взаимодействует с объектным хранилищем посредством API.

# Описание проблемы

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.

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.

В отличие от файлов в файловой системе, объекты хранятся в плоской структуре. Есть просто пул объектов: никаких папок, каталогов или иерархий. Для получения объекта необходимо просто сделать запрос к объектному хранилищу, предоставив идентификатор этого объекта.

Объекты могут быть как локальными, так и находиться на облачном сервере за тысячи километров, но, поскольку они находятся в плоском адресном пространстве, они извлекаются точно так же.

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

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.

Объектное хранилище обладает такими свойствами как:

**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.

# Обзор аналогов

На IT рынке уже существует немало облачных объектных хранилищ от крупных IT компаний. Например, Google cloud storage, Amazon s3, IBM object storage. Даже существуют 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.

Однако это не единственное следствие из облачного подхода к хранению данных. Важным недостатком является то, что клиент не знает, где и как хранятся его данные, когда компания, предоставляющая услугу, имеет доступ к конфиденциальной информации пользователей. И 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.

Многие люди, боясь за очередную утечку, хотят найти решение, которое позволит им сохранить свои данные в целости и безопасности как можно дальше от лишних глаз.

# Программная документация

* Текст

## **Техническое задание на программное обеспечение**

* Текст

## **Пояснительная записка к программному обеспечению**

* Текст

## **Описание программы**

* Текст

## **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}
2. POST /api/post -F file=”/path/to/file/”
3. PUT /api/put/{uid} -F file=”/path/to/file”
4. DELETE /api/delete/{uid}
5. GET /api/auth -H “email=email” -H “password=password”
6. POST /api/obj/{uid}/set-meta --data ‘{“key”:”value”}’
7. POST /api/obj/{uid}/add-meta --data ‘{“key”:”value”}’
8. GET /api/obj/{uid}/get-all-meta
9. GET /api/obj/{uid}/get-meta/key?={key}

|  |  |  |
| --- | --- | --- |
| 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 |

# Эксплуатационная документация на программный продукт

* Текст

# Акт испытаний программного продукта

* Текст

# Экономическое обоснование

* Текст

# Заключение

* Текст

# Список литературы

* Текст

# Приложения

* Текст