СИСТЕМА РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ ДАННЫХ ДЛЯ МАЛОГО БИЗНЕСА[1]

Большаков А.С.[2], Добряков А.С.[3], Туктаров Р. Р.[4]

Ключевые слова: *кибербезопасность, информационная безопасность, восстановление данных, доступность информации, отечественное программное обеспечение (ПО), программирование, алгоритмизация, критерии, архитектура, инфраструктура.*

Аннотация

Цель: разработать архитектуру системы распределенного хранения данных (СРХД) и основные алгоритмы сохранения и получения файла с удаленных серверов с поддержкой восстановления целостности файла при отключении одного из серверов.

Методы: экспертный анализ, метод разделения хранимых файлов на блоки с вычислением дополнительного блока чётности, позволяющий восстановить исходный файл в случае отключения одного из серверов, при этом не создавая высокую избыточность сохраняемых данных.

Результаты: реализован программный код приложения для сохранения, получения и восстановления данных в СРХД, а также алгоритм аутентификации пользователя на основе JWT (JSON Web Token); приведена схема рекомендуемой архитектуры для развертывания системы и описаны рекомендации по обеспечению информационной безопасности реализуемого ПО. Предлагаемая система позволяет создать отказоустойчивое хранилище данных, которое позволит обеспечить кибербезопасность, простоту масштабируемости, гибкость настройки, доступность, удобное администрирование и высокую производительность СРХД.

Практическая ценность: предложенное решение создания СРХД обеспечивает доступность и целостность файлов баз данных в информационных системах. Для реализации предлагаемой СРХД используется отечественное программное обеспечение с удобным для пользователя интерфейсом.

DOI: 10.21681/1994-1404-2024-2-61-71

Введение

Современные информационные системы должны соответствовать требованиям *отказоустойчивости* и *защищенности* [4], особенно когда речь идет о хранении внутренних данных бизнеса или данных его клиентов. В связи с этим рассмотрен возможный метод обеспечения дополнительной *доступности* данных в информационных системах с помощью применения *системы распределенного хранения данных* (СРХД). Особое внимание уделено построению СРХД для предприятий малого бизнеса, так как в настоящее время отсутствуют отечественные инструменты, пригодные для создания таких систем. Работа акцентирует внимание на обеспечении *информационной безопасности* [4] и предлагает решения для пользователя в части удобного и эффективного конфигурирования СРХД, использования неспециализированного оборудования для формирования хранилища данных предприятия малого бизнеса[5].

Данные являются одним из наиболее ценных активов для любого бизнеса, особенно для малых предприятий, которые полагаются на данные для принятия обоснованных решений, улучшения обслуживания клиентов и оптимизации своей деятельности. Однако хранение данных сопряжено с определенными обязанностями и рисками, такими как обеспечение их *без-*

[5] Малый бизнес — это локальная фирма с ограниченным списком видов деятельности, имеющая небольшую долю влияния на рынке. Его управление осуществляется собственником. Доход малого бизнеса составляет до 800 млн рублей в год, а среднесписочная численность персонала — до 100 сотрудников [5].

[1] Работа выполнена в рамках гранта МТУСИ для молодежных научных коллективов на основе приказа МТУСИ № 327-О от 19.12.2023.

[2] ***Большаков Александр Сергеевич,*** *кандидат технических наук, доцент, доцент кафедры информационной безопасности Московского технического университета связи и информатики, г. Москва, Российская Федерация.*

E-mail: alexbol57@mail.ru

[3] ***Добряков Александр Сергеевич,*** *студент Московского технического университета связи и информатики, г. Москва, Российская Федерация.*

E-mail: aleksandr@dobryakov.me

[4] ***Туктаров Рустам Русланович****, студент Московского технического университета связи и информатики, г. Москва, Российская Федерация.*

E-mail: lazzylust@yandex.ru

опасности, *конфиденциальности* и *доступности* [4]. В данной работе особое внимание уделено обеспечению *доступности* данных.

В отличие от СРХД для среднего и большого бизнеса, СРХД для малого не требует развертывания в кластере в связи с небольшой нагрузкой, а также хранит в себе меньшие объёмы информации и, соответственно, требует меньше вычислительных мощностей для поддержания работы [6]. Для поддержания работы СРХД в малом бизнесе не требуется также постоянное наличие специалиста в штате и высокая квалификация при установке и настройке.

Предприятие может собирать и хранить различные данные, как о своих пользователях, так и внутренние данные о работе бизнеса. Они могут включать в себя статистику, различную аналитику рынка, использование ресурсов и услуг бизнеса, итоги инвентаризации и др. [5]. Важно обеспечить непрерывную доступность к хранимым данным: например, в случае проверок или возникновения споров предприятию может потребоваться представить необходимые доказательства или документацию для поддержки своей позиции или защиты.

Хранение данных может повлечь за собой затраты и риски для малого бизнеса, такие, как стоимость хранения и управления данными, риск утечки или потери данных, риск несоблюдения требований и, вследствие этого, судебные разбирательства.

Следовательно, малому бизнесу требуется обеспечить надёжную *защиту* данных при небольшой стоимости оборудования и его обслуживания. Самое *надёжное* решение — аренда отказоустойчивого облачного хранилища, однако его стоимость может быть достаточно высока, а данные будут храниться у третьих лиц. Более *экономически выгодным* решением будет аренда облачного варианта управления данными, а хранение данных — на собственных серверах, однако в этом случае возникает задача обеспечения *отказоустойчивости* оборудования [12].

Существующие решения и критерии оценивания

Для начала следует выделить *критерии* оценивания систем распределенного хранения данных. Большую роль играет страна — разработчик системы. Здесь лучше ориентироваться на отечественные решения, предоставляющие гарантированно надёжную техническую поддержку, не содержащие зарубежного вредоносного *программного обеспечения* ПО, а также адаптированные под работу российских пользователей.

Наличие *веб-интерфейса* играет большую роль для использования системы малым бизнесом. Его наличие упростит работу с системой сотрудникам и позволит компании не разрабатывать собственный графический интерфейс.

Сложность установки и поддержания работоспособности системы является важным критерием. Чем проще установка и содержание системы, тем меньше затраты бизнеса, а соответственно, решение будет выгоднее.

Система должна поддерживать полное *восстановление* данных при отказе одного из серверов. Это позволит не только обеспечить отказоустойчивость в их доступности, но и поможет при переносе на новые сервера или расширении архитектуры.

Возможность установки системы на *NAS* (*Network Attached Storage* — сетевое хранилище данных) будет удобно при наличии существующей архитектуры из серверов, где уже развёрнуты какие-либо приложения. Это позволит согласовать работу внутренней сетевой инфраструктуры с СРХД.

Рассмотрим следующие системы распределенного хранения данных (табл. 1):

- Отечественная разработка компании «Рэйдикс» — ПО *“Raidix rain”* для создания блочной СРХД, работающей на серверных платформах «Эльбрус-4.4». Система имеет ряд преимуществ, одно из них — использование технологии *Raidix rain* для горизонтального масштабирования и расширения количества контроллеров ввода-вывода информации. Однако в концепции применения СРХД для малого бизнеса данная разработка имеет *недостатки*: высокую сложность установки и обслуживания — бизнесу придётся привлекать узконаправленного дорогостоящего специалиста, и ориентированность на отечественную аппаратную часть [4].
- *OpenMediaVault* с открытым программным кодом на базе дистрибутива *Debian* ориентирован на создание СРХД на отдельных серверах. Проект является зарубежной разработкой и в основном поддерживается разработчиками из-за рубежа. По сравнению с отечественной разработкой имеется более широкий функционал, возможности могут быть расширены плагинами, например, такими как *DAAP* медиа-сервер, *ISCSI*, *BitTorrent*-клиент и др. [2].

Разрабатываемая система учитывает основные критерии, сформулированные выше, и закрывает потребности малого бизнеса для хранения файлов и их восстановления. Техническая поддержка системы не является регулярным требованием и может быть осуществлена по запросу, так же как и во всех остальных рассматриваемых системах.

Система требует аппаратной и программной архитектуры, способной поддерживать работу контейнера *Docker* версии выше 0.13.0. При этом мастер-сервер должен иметь доступ в сеть с открытыми портами 80 и 443 для поддержания работы веб-интерфейса и пользовательского взаимодействия, и 3 000 для *API* (*application programming interface* — программный интерфейс приложения). Приложения баз данных (БД) должны иметь доступ к мастеру через локальную или глобальную сеть, открытый порт 300x, где *x* — номер приложения, начиная с 1.

Таблица 1

Рассматриваемые критерии СРХД

№ п/п	Критерии	Система РХД		
		OpenMediaVault	Рэйдикс	Разрабатываемая система
1	Страна-разработчик	США	Россия	Россия
2	Наличие графического веб-интерфейса	Да	Нет	Да
3	Наличие мастер- приложения	Да	Да	Да
4	Развертывание СРХД собственными силами	Нет	Нет	Да
5	Обеспечение восстановления данных	Да	Да	Да
6	Установка на NAS	Нет	Да	Нет
7	Стоимость лицензии ПО	Бесплатно	От 50 000 руб.	—
8	Тип лицензии	Стандартная общественная лицензия GNU (GPL)	Собственная лицензия	—

Техническая реализация

В случае использования СРХД на одном сервере, где находится некоторое количество баз данных (рис. 1), основным недостатком является зависимость состояния БД от стабильности работы сервера. При отключении сервера данные станут недоступны до восстановления его работоспособности. Данную проблему решает использование нескольких различных серверов, находящихся в разных сетях и независимых друг от друга, что и предлагается реализовать в разрабатываемой системе. То есть разрабатываемая система представляет собой комплекс мастер- и клиент-приложений БД, каждое из которых установлено на сервере под управлением операционных систем *Windows*, *Linux* или *Alt Linux*. Мастер-приложение необходимо в качестве конечной точки работы пользователя с системой. Оно принимает запросы на получение и сохранение файлов, регулирует эти процессы и восстанавливает файлы в случае отказа одного из серверов.

Мастер-приложение устанавливается в отказоустойчивое облако. Это необходимо для обеспечения непрерывной работы приложения, так как мастер обрабатывает все запросы, приходящие в СРХД, и в случае его отказа работа всей системы будет нарушена. Ведущие российские *IT* компании предоставляют решения с близким к нулю временем отказа и высокой производительностью. Мастер-приложение принимает запросы пользователей и отправляет их клиентам БД, на которых хранятся файлы.

Для обеспечения *отказоустойчивости* и *надёжности* данные на клиентах хранятся в виде частей, рассредоточенных по всей инфраструктуре (рис. 2). Одна из частей является блоком чётности, что позволяет восстановить недостающую часть файла в случае отказа одного из серверов.

Рис. 1. Топология сети для СРХД с одним сервером

В данной работе рассматривается реализация системы в инфраструктуре из 3 серверов: мастер-приложение и клиенты БД № 1, № 2 и № 3, на котором хранятся блоки чётности файлов.

Данная упрощённая инфраструктура приведена для демонстрации работы системы. В реальном случае использования системы мастер-приложение и клиенты должны быть установлены на разных серверах, находящихся в разных сетях. Важно учитывать, что систе-

Рис. 2. Пример распределения данных на серверах

ма поддерживает отказ любого из клиент-приложений, в том числе и с блоками чётности. Сейчас отказ клиента с блоками чётности может быть воспроизведён путём отключения приложения.

Ниже приведены алгоритмы работы приложения в данной инфраструктуре. Для получения доступа к чтению и записи файлов, пользователю необходимо пройти аутентификацию (рис. 3).

Исходной частью файла является его часть, полученная при сохранении. Части равны между собой по размеру. Блок файла — исходная часть или данные о чётности между двумя частями. Чётность вычисляется через функцию исключающего «ИЛИ» (рис. 4). Данная функция позволяет восстановить любую из недостающих частей файла при наличии второй. Объединение данных происходит конкатенацией частей **А** и **Б**, в результате чего получается запрашиваемый файл.

Для реализации записи файла в систему и чтения файла в системе разработаны соответствующие алгоритмы (рис. 5 и 6).

В случае отказа одного из серверов файлы могут быть прочитаны с использованием оставшихся данных. Так, при отказе сервера с блоками исходных данных недостающий блок будет вычислен с использованием блока чётности (рис. 7 и рис. 8). А при стабильной работе или потери связи с сервером блоков чётности данные будут получены без какого-либо восстановления (рис. 9 и рис. 10). Вычисление недостающей части с использованием блока чётности происходит в соответствии с формулами 1 и 2, где F_A — первая часть файла, F_B — вторая часть файла, P — блок чётности двух частей:

$$F_A = F_B \oplus P \quad (1)$$

$$F_B = F_A \oplus P \quad (2)$$

Приведём пример записи файла с содержимым *"Hello world!"*. В реализуемой системе файлы хранятся и обрабатываются в формате буфера (массива) из пар шестнадцатеричных чисел. В таком представлении файл будет иметь следующее содержимое:

«48 65 6c 6c 6f 20 77 6f 72 6c 64 21».

Следовательно, содержимое части **А** файла: «48 65 6c 6c 6f 20», а части Б: «77 6f 72 6c 64 21». Чётность (*P*) будет вычислена по формуле (3):

$$P = F_A \oplus F_B \quad (3)$$

и будет равна «3f 0a 1e 00 0b 01».

Рассмотрим подробнее вычисление чётности. Данные буфера хранятся в соответствующем формате, затем производится вычисление суммы по модулю 2 частей **А** и **Б** (рис. 11).

Полученные 3 блока данных передаются на серверы БД, где происходит их сохранение. При получении файла, в случае если серверы с частями файла **А** и **Б** в сети, производится конкатенация. Так, для указанных выше частей конкатенация будет иметь следующий вид (рис. 12).

Однако если один из серверов не в сети, будет произведено получение блока чётности с соответствующего сервера, и вычисление недостающей части с помощью формулы (1) в случае отсутствия части **А** (рис. 13), а в случае отсутствия части **Б** — с помощью формулы (2) (рис. 14).

После восстановлений недостающей части производится конкатенация частей **А** и **Б** (рис. 12). В результате получаем исходный файл, даже если один из серверов недоступен.

Рекомендуемая архитектура развёртывания приложения

Разработанную систему распределенного хранения данных рекомендуется разворачивать на серверах, находящихся в различных сетях и удалённых друг от друга, соблюдая следующие критерии:

- мастер-приложение должно быть размещено на отказоустойчивом сервере, например, в арендованном облаке;
- количество клиентов БД должно быть не менее 3;
- один клиент БД должен быть размещён внутри инфраструктуры бизнеса (при её наличии) и не хранить блоки чётности;
- при хранении персональных данных пользователей все сервера должны быть сертифицированы

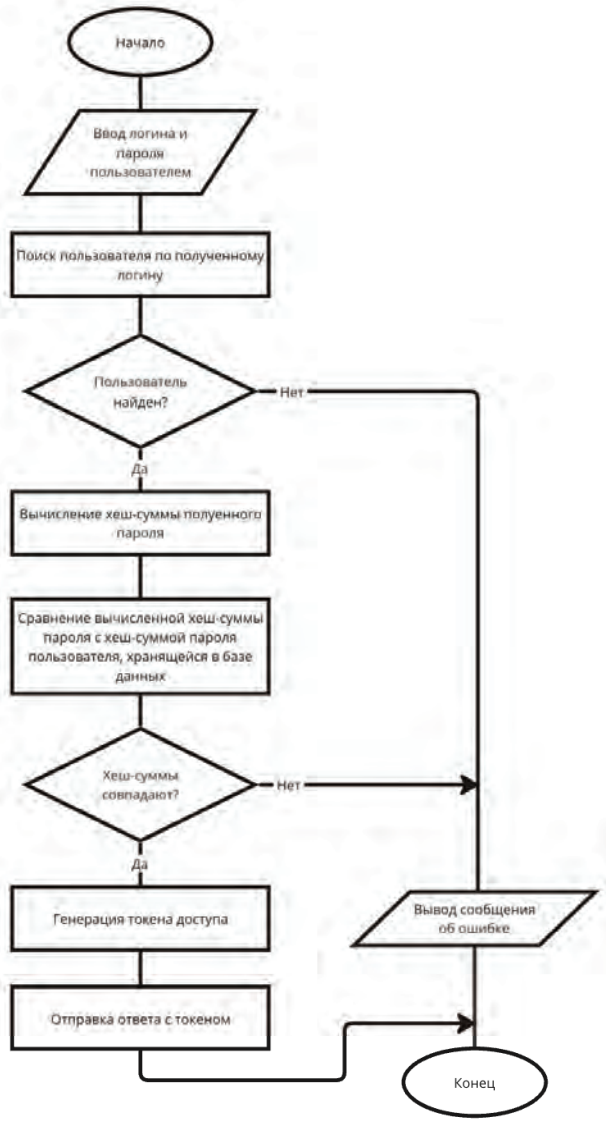

Рис. 3. Функциональная схема алгоритма аутентификации пользователя на мастере-приложении

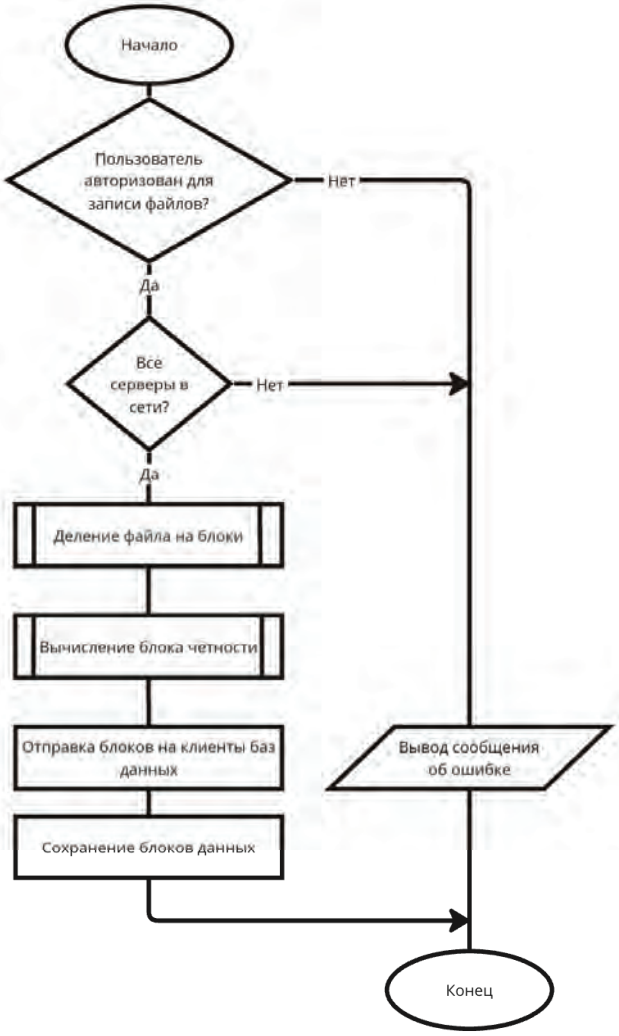

Рис. 5. Функциональная схема алгоритма записи файла в систему

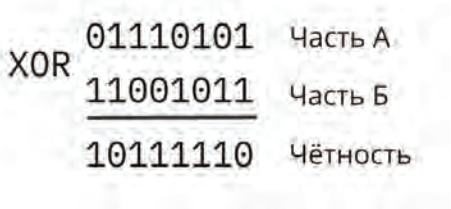

Рис. 4. Пример вычисления чётности через исключающее «ИЛИ»

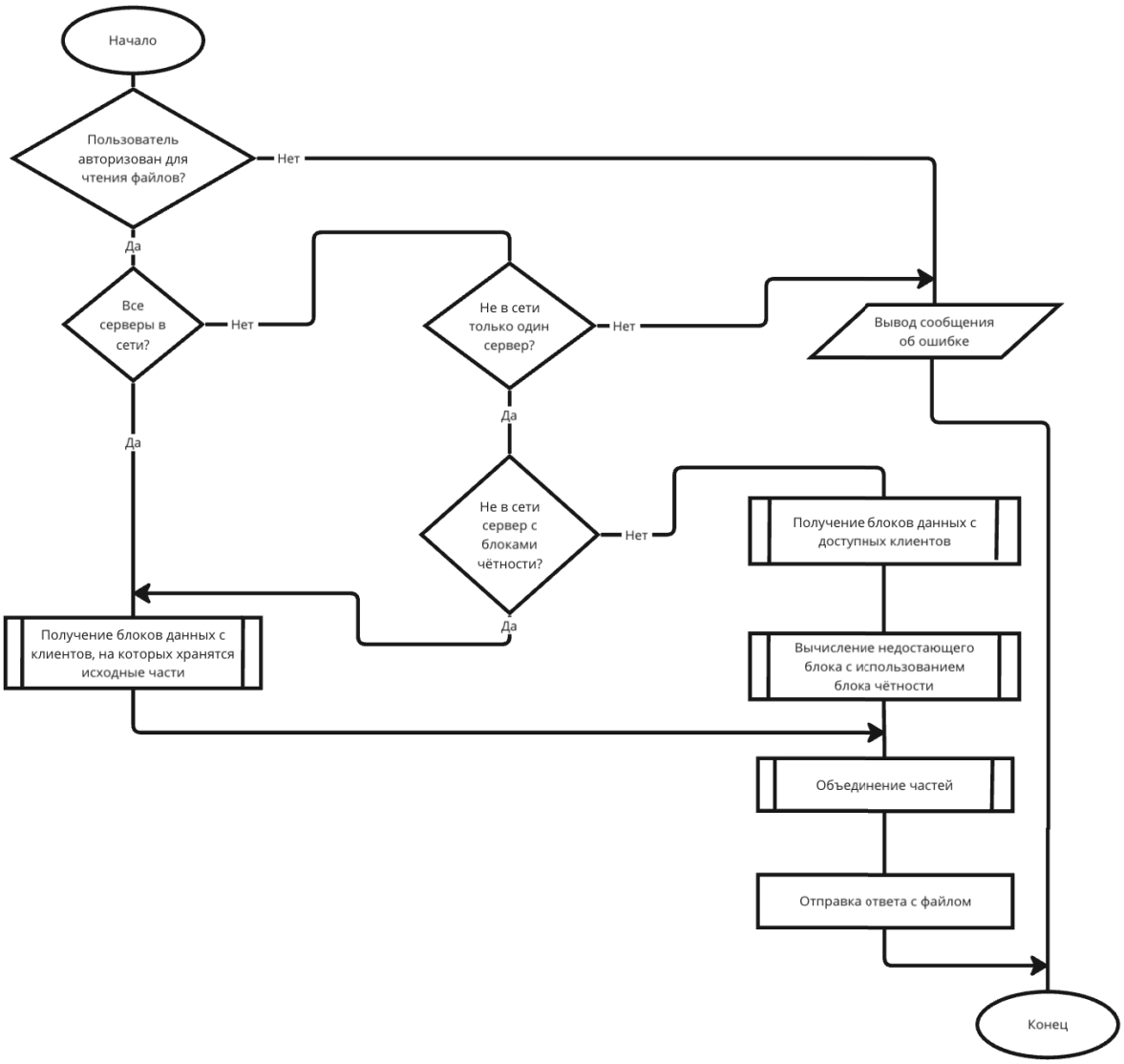

Рис. 6. Функциональная схема алгоритма чтения файла в системе

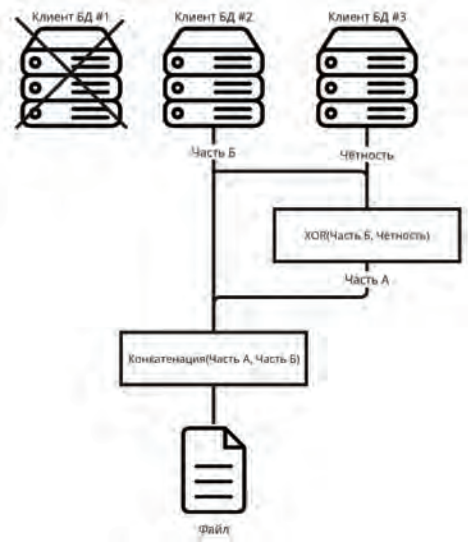

Рис. 7. Схема процесса восстановления файла при отключении сервера

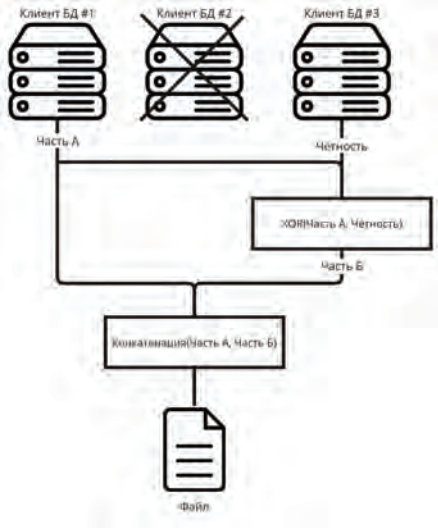

Рис. 8. Схема процесса восстановления файла при отключении сервера

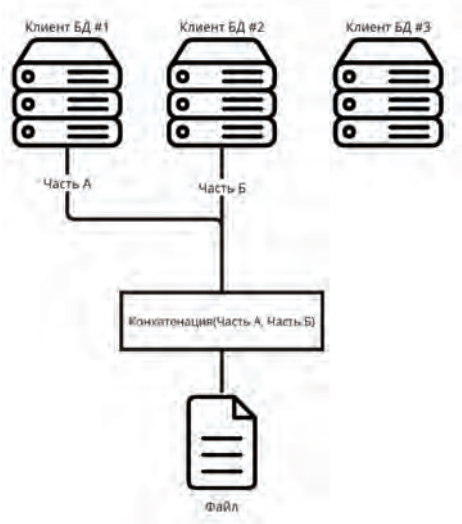

Рис. 9. Схема процесса получения файла при нормальной работе

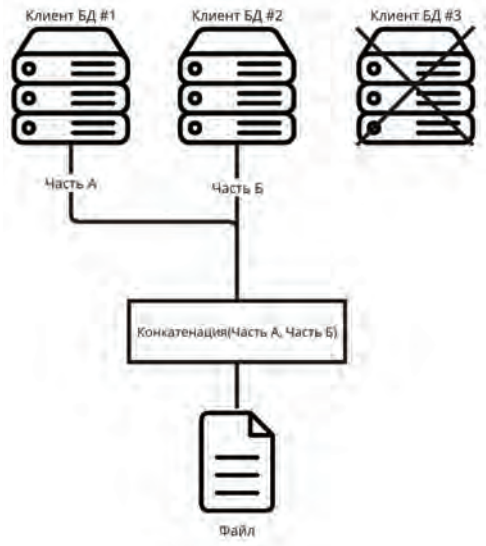

Рис. 10. Схема процесса получения файла при отказе сервера с блоками чётности

$$\begin{array}{l} \oplus \begin{array}{l} \text{48 65 6c 6c 6f 20} \\ \text{77 6f 72 6c 64 21} \end{array} \\ \hline \text{3f 0a 1e 00 0b 01} \end{array}$$

Рис. 11. Вычисление блока чётности

$$\text{48 65 6c 6c 6f 20} + \text{77 6f 72 6c 64 21} = \\ = \text{48 65 6c 6c 6f 20 77 6f 72 6c 64 21}$$

Рис. 12. Конкатенация файла из примера

$$\begin{array}{l} \oplus \begin{array}{l} \text{77 6f 72 6c 64 21} \\ \text{3f 0a 1e 00 0b 01} \end{array} \\ \hline \text{48 65 6c 6c 6f 20} \end{array}$$

Рис. 13. Вычисление недостающей части А с помощью чётности

$$\begin{array}{l} \oplus \begin{array}{l} \text{48 65 6c 6c 6f 20} \\ \text{3f 0a 1e 00 0b 01} \end{array} \\ \hline \text{77 6f 72 6c 64 21} \end{array}$$

Рис. 14. Вычисление недостающей части Б с помощью чётности

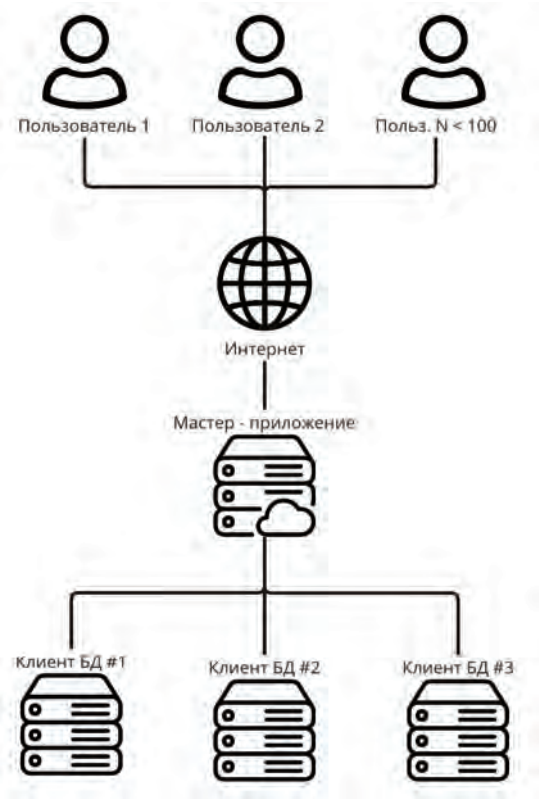

Рис. 15. Пример рекомендуемой архитектуры РСХД малого бизнеса

в соответствии с Федеральным законом № 152-ФЗ «О персональных данных».

С учётом всех критериев рекомендуемая архитектура СРХД для малого бизнеса изображена на рис. 15, где клиент БД #3 хранит блоки чётности файлов.

Аутентификация и авторизация пользователя

Аутентификация пользователя осуществляется с помощью *JWT* — стандарт для создания токенов доступа, основанный на формате *JSON*. Он широко используется для передачи данных аутентификации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения подлинности запросов от своей учётной записи. *JWT* авторизации состоит из трёх частей [3] (рис. 16):

- *заголовок* (*Header*) — содержит информацию о типе токена, алгоритме подписи данных и другие метаданные;
- *полезная нагрузка* (*Payload*) — содержит идентификационные данные пользователя, может нести в себе логин, роли, *id* и др. В разрабатываемом приложении содержит только *id* для обеспечения повышенной безопасности;
- *подпись* (*Signature*) — используется для проверки целостности и неизменности полезной нагрузки, создаётся с помощью секретного ключа (*Secret*) на сервере. Подпись вычисляется на основе заголовка и полезной нагрузки с использованием алгоритма, указанном в заголовке.

После того как все части токена созданы, они объединяются через символ « . » в строку и отправляются пользователю.

В разрабатываемом приложении для подписи *JWT* используется алгоритм хэширования *HMAC-SHA*256, который является стандартным алгоритмом для подписи токенов авторизации. Алгоритм основан на *SHA*-256, использует секретный ключ для создания хэша из предоставленного сообщения. Итоговая подпись имеет размер 256 *бит*, что достаточно для проверки *подлинности* и *целостности* данных аутентификации.

Для повышения *кибербезопасности* СРХД срок действия *JWT* ограничен и составляет 15 мин. Это означает, что через 15 мин после выполнения «входа» сервер перестанет принимать запросы от пользователя, предоставляющего просроченный токен, и ему нужно будет повторить «вход». Чтобы время действия сессии пользователя не зависело от срока действия *JWT*, применяется токен обновления доступа [7, 8].

Токен обновления доступа — это дополнительный токен, который используется для получения нового *JWT* после истечения срока действия предыдущего. В отличие от *JWT*, срок действия которого ограничен 15 мин, токен доступа бессрочен и перестаёт действовать только после завершения сессии пользователем. Применение токена обновления доступа позволяет пользователю не выполнять повторный «вход» в приложение и иметь постоянно активную сессию, сохранив при этом *безопасность* использования краткосрочных *JWT* [8]. Когда срок действия *JWT* истекает, клиент отправляет промежуточный запрос на обновление токена доступа, передав токен обновления доступа. Сервер выполняет необходимые проверки, генерирует новую пару токенов и отправляет их клиенту.

Генерация токена обновления доступа происходит по алгоритму *UUID v*4. Он представляет собой случайный идентификатор длительностью 128 бит. Токен обновления может иметь следующий вид:

“840d1728-5cf2-4297-b4e8-12cd0725fc45”.

В отличие от *JWT*, этот токен хранится в базе данных сервера и отвечает за действительность сессии пользователя. Данная структура позволяет обезопасить *достоверность* [4] запросов от клиента к серверу, а также отзывать токены в случае их компрометации.

Система поддерживает аутентификацию пользователей по логину и паролю. Для доступа к чтению и записи файлов пользователю необходимо выполнить «вход» в приложение, отправив логин и пароль в запросе аутентификации. Сервер осуществляет поиск пользователя по предоставленному логину, затем, если пользователь найден, генерирует хеш-сумму из полученного пароля и сравнивает её с той, что сохранена в записи пользователя. Если хеш-суммы совпадают, сервер генерирует токен с полезной нагрузкой в виде *id* пользователя и отправляет ответ с токеном. Если на каком-либо этапе произошла ошибка, то он отправляет ответ с указанием обнаруженной ошибки. Полученный токен пользователь должен предоставлять при подаче каждого запроса [1].

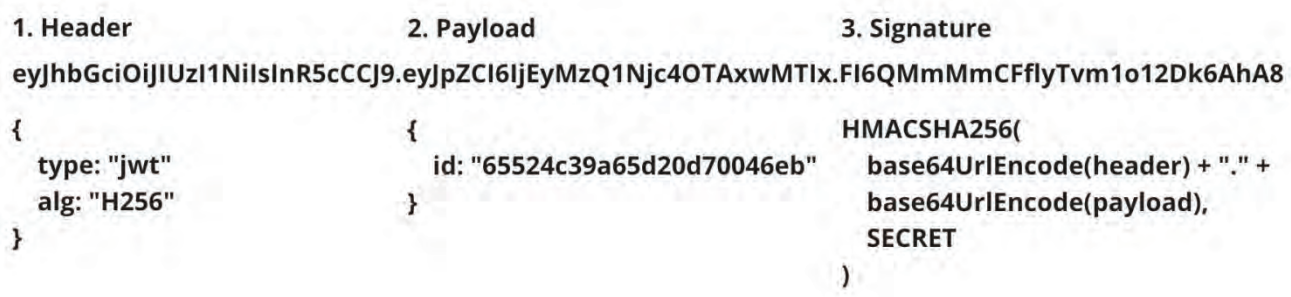

Рис. 16. Структура JWT

Тестирование реализованного ПО

Тестирование системы на скорость сохранения и получение файлов проводится на оборудовании различной конфигурации:

- сервер 1: ЦП *Apple M*1 *Pro*, 16 *Гб* ОЗУ *DDR*4, *SSD*;
- сервер 2: ЦП *Intel Xeon E*5 2620 *V*3 6x3.5 *ГГц*, 16 *Гб* ОЗУ DDR3, *HDD* 7200 *RPM*;
- сервер 3: ЦП *Intel Xeon E*5-2689 8x2.2 ГГц, 8 *Гб* ОЗУ *DDR*4, *SSD*.

По итогу тестирования получены положительные результаты обработки, сохранения и извлечения информации программой [13]. Данные показатели можно улучшить, реализовав алгоритм на более продвинутых языках программирования, таких, например, как *C++* или *Java* [10, 11].

№ п/п	Тестируемая характеристика	Сервер 1, *Мб/с*	Сервер 2, *Мб/с*	Сервер 3, *Мб/с*
1	Сохранение файла при нормальной работе	5.2	4.8	5.6
2	Получение файла при нормальной работе	7.1	6.1	8.2
3	Получение файла в режиме восстановления при отказе одного сервера	3.7	4.5	5.2

Заключение

Таким образом, разработанная система распределённого хранения файлов для малого бизнеса отвечает обоснованным критериям и требованиям. Предложена схема архитектуры для развёртывания системы с учётом специфики её работы и мер защиты информации, а также алгоритмы работы программы. Данная разработка имеет потенциал реализации и коммерциализации, комплексно решая задачу обеспечения доступности и целостности информации внутри инфраструктуры малого бизнеса.

Рецензент: ***Ловцов Дмитрий Анатольевич,*** *доктор технических наук, профессор, заслуженный деятель науки Российской Федерации, заведующий кафедрой информационного права, информатики и математики Российского государственного университета правосудия, г. Москва, Российская Федерация.*
E-mail: dal-1206@mail.ru

Литература

1. Абдулаева У.Б. Программная реализация серверной части регистрации/аутентификации в веб-приложении // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2023. № 4 (42). С. 5—20. EDN: ASKUHN.
2. Зарубин А.А. Анализ характеристик открытых систем хранения данных / А.А. Зарубин, А.А. Савельева, А.А. Швидкий // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022) : XI Международная научно-техническая и научно-методическая конференция, Санкт-Петербург, 15—16 февраля 2022 г. Том 1. СПб. : Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2022. С. 473—478. EDN: XDNXLH.
3. Колесников А.О. Идентификация пользователей клиент-серверных приложений с помощью JWT-токена // EurasiaScience. Труды XXXVI Междунар. науч.-прак. конф., 31 марта 2021 г. М. : Актуальность.РФ, 2021. С. 42—43. EDN: HCBTDB.
4. Ловцов Д.А. Теория защищенности информации в эргасистемах : монография. М. : РГУП, 2021. 276 с. ISBN 978-5-93916-896-0.
5. Лопастейская Л.Г., Хасянова Ю.Р. Малый бизнес: общая характеристика и критерии отнесения организаций к малому бизнесу // Труды Междунар. науч.-прак. конф. «Научные революции: сущность и роль в развитии

науки и техники», 18 января 2019 г. Часть 1. Магнитогорск : ООО «Агентство международных исследований», 2019. С. 140—142. EDN: VRRJUA.
6. Иванов И.П., Гантимуров А.П., Виниченко А.Д., Босов А.В. Методы оптимизации информационных потоков в бизнес-процессах с распределенной системой хранения данных // Перспективы науки. 2019. № 9 (120). С. 35—39. EDN: FDWNCH.
7. Мишин А.Б., Марковин А.Ю. Варианты систем аутентификации в веб-приложениях // Телекоммуникации. 2022. № 5. С. 27—32. DOI: 10.31044/1684-2588-2022-0-5-27-32 .
8. О некоторых особенностях JWT-аутентификации в веб-приложениях / А.Б. Бетелин, И.Б. Егорычев, А.А. Прилипко и др. // Труды НИИ системных исследований РАН. 2021. Т. 11. № 1. С. 4—10. DOI: 10.25682/NIISI.2021.1.0001 .
9. Петренко С.А. Модель программно-определяемого хранилища данных на основе инвариантов подобия и размерностей / С.А. Петренко, А.А. Петренко // Дистанционные образовательные технологии : сборник трудов V Международной научно-практической конференции, Ялта, 22—25 сентября 2020 г. / Ответственный редактор В.Н. Таран. Симферополь : ООО «Издательство типография «Ариал», 2020. С. 388—394. EDN: QDRBWU.
10. Садовский Б.С. Эффективность виртуальных функций языка С++ на примере алгоритма сортировки простыми обменами // Наука и бизнес: пути развития. 2021. № 3 (117). С. 59—62. EDN: IPWHMZ.
11. Сайфуллин И.И., Айзатуллова А.Ф., Шмакова А.Ф. Исследование эффективности применения Java Streams в параллельной обработке больших данных // Изв. Тульского ГУ. Технические науки. 2023. № 9. С. 261—263. DOI: 10.24412/2071-6168-2023-9-261-262 .
12. Сираева Э.Ф. Опасности для облачной безопасности и способы защиты данных в облачных хранилищах // Труды VI Всеросс. молодеж. науч.-прак. конф. с междунар. участием «Информационные технологии обеспечения комплексной безопасности в цифровом обществе», 19—20 мая 2023 г. Уфа: Уфимский ун-т науки и технологий, 2023. С. 165—168.
13. Швидкий А.А. Анализ методов определения характеристик систем хранения данных при различных типах нагрузки // Труды XI Междунар. науч.-прак. и науч.-метод. конф. «Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022)», 15—16 февраля 2022 г. Т. 1. СПб.: СПбГУТ им. проф. М.А. Бонч-Бруевича, 2022. С. 788—793. EDN: CFGQNS.

A DISTRIBUTED DATA STORAGE SYSTEM FOR SMALL BUSINESS

Aleksandr Bol'shakov, *Ph.D. (Technology), Associate Professor at the Information Security Department of the Moscow Technical University of Communication and Informatics, Moscow, Russian Federation.*
E-mail: alexbol57@mail.ru

Aleksandr Dobriakov, *student at the Moscow Technical University of Communication and Informatics, Moscow, Russian Federation.*
E-mail: aleksandr@dobryakov.me

Rustam Tuktarov, *student at the Moscow Technical University of Communication and Informatics, Moscow, Russian Federation.*
E-mail: lazzylust@yandex.ru

Keywords: *cyber security, information security, data recovery, information accessibility, domestic software, programming, algorithmisation, criteria, architecture, infrastructure.*

Abstract

Purpose of the work: developing the architecture for a distributed data storage system (DDSS) and the basic algorithms for saving and retrieving a file from remote servers with the support for restoring the file integrity when one of the servers is disconnected.

Methods used in the study: expert analysis, a method for dividing files into blocks and calculating an additional parity block which allows to restore the original file when a server is disconnected without creating a high redundancy of the saved data.

Study findings: an application programme for saving, receiving and restoring data in the DDSS as well as the user authentication algorithm based on JSON Web Token (JWT) is implemented. A recommended architecture diagram for deploying the system and recommendations for ensuring information security of the implemented software are given. The proposed system allows to create a fault-tolerant data warehouse allowing to ensure cyber security, easy scalability, customisation flexibility, accessibility, convenient administering and high performance of the DDSS.

Practical value: the proposed solution for setting up a DDSS ensures accessibility and integrity of database files in information systems. Domestic software with a friendly user interface is used for implementing the proposed DDSS.

References

1. Abdulaeva U.B. Programmnaia realizatsiia servernoi chasti registratsii/autentifikatsii v veb-prilozhenii. Informatsionno-komp'iuternye tekhnologii v ekonomike, obrazovanii i sotsial'noi sfere, 2023, No. 4 (42), pp. 5–20. EDN: ASKUHN.
2. Zarubin A.A. Analiz kharakteristik otkrytykh sistem khraneniia dannykh. A.A. Zarubin, A.A. Savel'eva, A.A. Shvidkii. Aktual'nye problemy infotelekommunikatsii v nauke i obrazovanii (APINO 2022) : XI Mezhdunarodnaia nauchno-tekhnicheskaia i nauchno-metodicheskaia konferentsiia, Sankt-Peterburg, 15–16 fevralia 2022 g. Tom 1. SPb. : Sankt-Peterburgskii gosudarstvennyi universitet telekommunikatsii im. prof. M.A. Bonch-Bruevicha, 2022, pp. 473–478. EDN: XDNXLH.
3. Kolesnikov A.O. Identifikatsiia pol'zovatelei klient-servernykh prilozhenii s pomoshch'iu JWT-tokena. EurasiaScience. Trudy XXXVI Mezhdunar. nauch.-prak. konf., 31 marta 2021 g. M. : Aktual'nost'.RF, 2021, pp. 42–43. EDN: HCBTDB.
4. Lovtsov D.A. Teoriia zashchishchennosti informatsii v ergasistemakh : monografiia. M. : RGUP, 2021. 276 pp. ISBN 978-5-93916-896-0.
5. Lopasteiskaia L.G., Khasianova Iu.R. Malyi biznes: obshchaia kharakteristika i kriterii otneseniia organizatsii k malomu biznesu. Trudy Mezhdunar. nauch.-prak. konf. "Nauchnye revoliutsii: sushchnost' i rol' v razvitii nauki i tekhniki", 18 ianvaria 2019 g. Chast' 1. Magnitogorsk : OOO "Agentstvo mezhdunarodnykh issledovanii", 2019, pp. 140–142. EDN: VRRJUA.
6. Ivanov I.P., Gantimurov A.P., Vinichenko A.D., Bosov A.V. Metody optimizatsii informatsionnykh potokov v biznes-protsessakh s raspredelennoi sistemoi khraneniia dannykh. Perspektivy nauki, 2019, No. 9 (120), pp. 35–39. EDN: FDWNCH.
7. Mishin A.B., Markovin A.Iu. Varianty sistem autentifikatsii v veb-prilozheniiakh. Telekommunikatsii, 2022, No. 5, pp. 27–32. DOI: 10.31044/1684-2588-2022-0-5-27-32 .
8. O nekotorykh osobennostiakh JWT-autentifikatsii v veb-prilozheniiakh. A.B. Betelin, I.B. Egorychev, A.A. Prilipko i dr. Trudy NII sistemnykh issledovanii RAN, 2021, t. 11, No. 1, pp. 4–10. DOI: 10.25682/NIISI.2021.1.0001 .
9. Petrenko S.A. Model' programmno-opredeliaemogo khranilishcha dannykh na osnove invariantov podobiia i razmernostei. S.A. Petrenko, A.A. Petrenko. Distantsionnye obrazovatel'nye tekhnologii : sbornik trudov V Mezhdunarodnoi nauchno-prakticheskoi konferentsii, Ialta, 22–25 sentiabria 2020 g. Otvetstvennyi redaktor V.N. Taran. Simferopol' : OOO "Izdatel'stvo tipografiia "Arial", 2020, pp. 388–394. EDN: QDRBWU.
10. Sadovskii B.S. Effektivnost' virtual'nykh funktsii iazyka C++ na primere algoritma sortirovki prostymi obmenami. Nauka i biznes: puti razvitiia, 2021, No. 3 (117), pp. 59–62. EDN: IPWHMZ.
11. Saifullin I.I., Aizatullova A.F., Shmakova A.F. Issledovanie effektivnosti primeneniia Java Streams v parallel'noi obrabotke bol'shikh dannykh. Izv. Tul'skogo GU. Tekhnicheskie nauki, 2023, No. 9, pp. 261–263. DOI: 10.24412/2071-6168-2023-9-261-262 .
12. Siraeva E.F. Opasnosti dlia oblachnoi bezopasnosti i sposoby zashchity dannykh v oblachnykh khranilishchakh. Trudy VI Vseross. molodezh. nauch.-prak. konf. s mezhdunar. uchastiem "Informatsionnye tekhnologii obespecheniia kompleksnoi bezopasnosti v tsifrovom obshchestve", 19–20 maia 2023 g. Ufa: Ufimskii un-t nauki i tekhnologii, 2023, pp. 165–168.
13. Shvidkii A.A. Analiz metodov opredeleniia kharakteristik sistem khraneniia dannykh pri razlichnykh tipakh nagruzki. Trudy XI Mezhdunar. nauch.-prak. i nauch.-metod. konf. "Aktual'nye problemy infotelekommunikatsii v nauke i obrazovanii (APINO 2022)", 15–16 fevralia 2022 g, t. 1. SPb.: SPbGUT im. prof. M.A. Bonch-Bruevicha, 2022, pp. 788–793. EDN: CFGQNS.