

Министерство образования Республики Беларусь
Учреждение образования Белорусский государственный университет
информатики и радиоэлектроники

УДК 004.623

Бут-Гусаим
Роман Александрович

Распределённое программное средство
организации хранения и доступа к данным

АВТОРЕФЕРАТ

на соискание степени магистра технических наук
по специальности 1-40 80 05 Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Научный руководитель

Серебряная Лия Валентиновна

к.т.н., доцент

Минск 2017

КРАТКОЕ ВВЕДЕНИЕ

В эпоху стремительного развития интернета, облачных технологий и веб-сервисов широкую популярность приобретают системы для хранения пользовательских данных. Толчком для развития такого рода систем послужило резкий рост скорости доступа в интернет, увеличение вычислительных мощностей ЭВМ, а также совершенствование технологии производства жёстких дисков, что в свою очередь значительно снизило стоимость хранения единицы информации. Безусловные преимущества программных таких систем: отсутствие необходимости размещения данных на компьютере, организация быстрого доступа к необходимой информации с разных ЭВМ, удалённых друг от друга на тысячи километров, – способствуют стремительному развитию данного рынка программного обеспечения.

Построение систем такого типа автоматически предъявляет требования по значительной горизонтальной масштабируемости, т.к. в настоящее время вычислительные возможности одного ЭВМ ограничены характеристиками аппаратного обеспечения и недостаточны для удовлетворения нужд большого количества пользователей. Как следствие, такие решения автоматически становятся распределёнными с вытекающими из этого требованиями по обеспечению открытости, безопасности, устойчивости и т.п. Выполнение данных требований требует значительных усилий при проектировании и реализации системы как на аппаратном, так и на программном уровне.

Немаловажной особенностью такого рода систем является гарантия согласованности размещённых пользователем данных. Для её обеспечения можно использовать репликацию информации на нескольких серверах, а также непосредственное вычисление хеш-функции полученных данных. Как результат, практически полностью отпадает зависимость от одной версии файла и при повреждении данных на одном из серверов необходимая информация может быть получена со специально предназначенных для этих целей реплик. Однако распределённый характер изучаемой системы налагает дополнительные требования по обеспечению с одной стороны локальности к пользователю и удалённости для реплик, чтобы выход из строя группы близко расположенных узлов не привёл к полной потере доступа к желаемому ресурсу.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

Целью диссертационной работы является разработка распределённого программного средства организации хранения и доступа к данным.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ методов и способов построения распределённых программных средств.
2. Провести анализ методов и способов организации хранения и доступа к данным.
3. Предложить методы и алгоритмы построения распределённых программных средств хранения и доступа к данным.
4. Провести оценку эффективности реализации распределённого программного средства организации хранения и доступа к данным.

Объектом исследования являются распределённые системы.

Предметом исследования является распределённое программное средство организации хранения и доступа к данным.

Основной *гипотезой*, положенной в основу диссертационной работы, является возможность исследования и улучшения существующих практик и подходов по построению распределённых систем для хранения и передачи данных. Объёмы данных, с которыми оперирует человек в своей повседневной деятельности, испытывают существенный рост с каждым годом, поэтому поиск оптимальных решений в данной области уменьшит временные и финансовые затраты по работе с этой информацией.

Связь работы с приоритетными направлениями научных исследований и запросами реального сектора экономики

Работа выполнялась в соответствии с научно-техническим заданием и планом работ кафедры «Программное обеспечение информационных технологий» по теме «Разработка моделей, методов, алгоритмов, повышающих показатели проектирования, внедрения и эксплуатации программных средств для перспективных платформ обработки информации, решения интеллектуальных задач, работы с большими массивами данных и внедрение в современные обучающие комплексы» (ГБ № 16-2004, № ГР 20163588, научный руководитель НИР – Н. В. Лапицкая).

Личный вклад соискателя

Результаты, приведённые в диссертации, получены соискателем лично. Вклад научного руководителя Л.В. Серебряной заключается в формулировке целей и задач исследования.

Апробация результатов диссертации

Основные положения диссертационной работы докладывались и обсуждались на 52-й научно-практической конференции БГУИР «Использование акторной модели и очередей сообщений при построении распределённых систем» (Минск, Беларусь, 2016).

Опубликованность результатов диссертации

По теме диссертации опубликована 1 (одна) печатная работа в сборнике трудов и материалов международной научно-практической конференции молодых учёных «Научно-технический прогресс и современное общество» (г. Москва).

Структура и объём диссертации

Диссертация состоит из введения, общей характеристики работы, четырёх глав, заключения, списка использованных источников и приложения. В первой главе представлен анализ предметной области, выявлены основные существующие проблемы в рамках тематики исследования, проанализированы существующие модели и методы, позволяющие моделировать процессы тематики исследования. Вторая глава посвящена анализу и разработке моделей открытости, масштабируемости и хранения данных, призванных решать проблемы, возникающие при построении распределённых систем хранения данных. В третьей главе рассматриваются общие алгоритмы, используемые при построении программного средства, а также криптографические алгоритмы и алгоритмы масштабируемости. В рамках четвёртой главы рассмотрено построение архитектуры программного средства, его непосредственное моделирование и реализация компонент, а также проведён анализ полученных результатов.

Общий объём работы составляет 129 страниц, из которых основного текста – 75 страниц, 45 рисунков на 27 страницах, список использованных источников из 58 наименований на 5 страницах и 1 приложение на 22 страницах.

ОСНОВНОЕ СОДЕРЖАНИЕ

Во **введении** определена область и указаны основные направления использования разрабатываемого решения. Также показана актуальность темы для решения современных задач в области организации хранения и доступа к данным.

В **первой главе** проведён анализ распределённых систем и очерчен ряд сложностей и проблем, связанных с их построением. К ним относятся:

1. Гетерогенность компонентов.
2. Открытость.
3. Масштабируемость.
4. Прозрачность.
5. Отказоустойчивость.
6. Безопасность.

Помимо этого, в первой главе были рассмотрены основные современные архитектурные подходы, которые используются при проектировании и построении распределённых систем: классическая клиент-серверная, одноранговая и микросервисная, а также выделены преимущества и недостатки каждого из них.

В первой главе также рассматриваются основные подходы к организации хранения информации. Среди ключевых можно выделить:

1. Распределённая файловая система.
2. Реляционные базы данных.
3. Нереляционные базы данных.

В заключении в первой главе представлены задачи и цели диссертационной работы.

Во **второй главе** производится описание ряда моделей, необходимых для построения распределённого программного средства организации хранения и доступа к данным. Одной из них является модель открытости, призванная решить задачу организацию доступа к информации для пользователей системы. Рассмотренное решение является логическим развитием модели зрелости Ричардсона.

В рамках второй главы также рассмотрены основные недостатки использования потоков для организации масштабируемости в рамках системы. Для решения очерченных проблем предложено использование акторной модели, в основе которых лежат концепции параллельных вычислений и передачи сообщений.

Актор представляет собой базовую вычислительную единицу и включает в себя три важных понятия:

1. Обработка, т.к. он существует с целью выполнения определённой задачи.

2. Хранилище, т.к. ему необходимо хранить состояние.
3. Взаимодействие, т.к. акторы нет смысла рассматривать в единичном экземпляре – только в виде системы.

Для взаимодействия акторы применяют асинхронный механизм передачи сообщений (рисунок 1). В частности, акторы не используют каких-либо промежуточных сущностей, например, каналов. Вместо этого у актора есть почтовый ящик (mailbox), в который поступают приходящие сообщения.

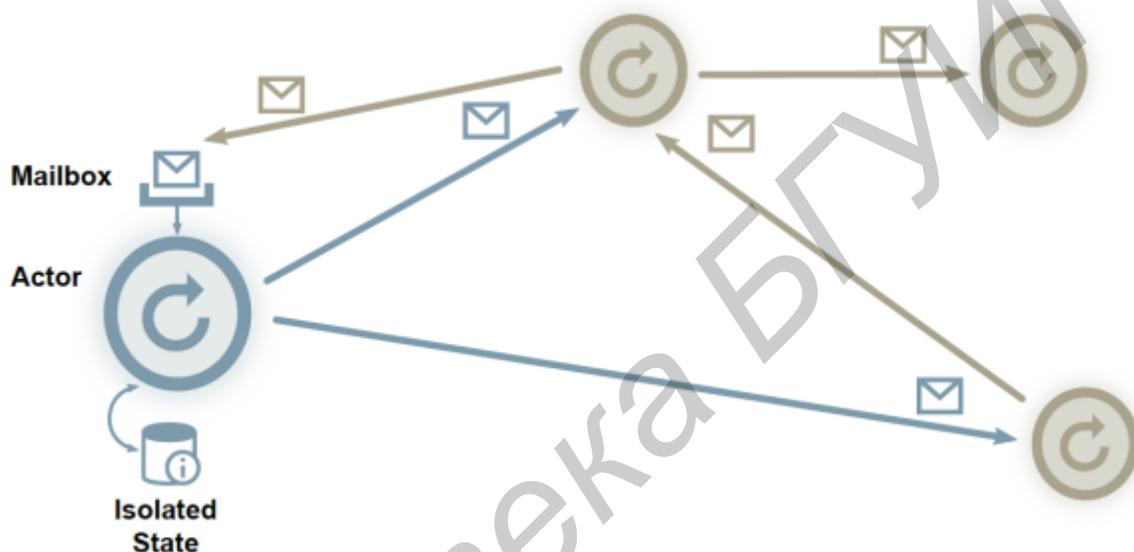


Рисунок 1 – Пример взаимодействия акторов в акторной модели

Недостатком акторной модели является то, что в её рамках доставка сообщений гарантируется максимум 1 раз, что может быть неудовлетворительно для ряда систем. Для решения этой проблемы могут быть использованы очереди сообщений, в основе которых лежат так называемые транзакционные персистентные очереди [1]. Очередь представляет собой транзакционный ресурс, где изменения происходят либо целиком, либо вообще не происходят (в случае прерывания транзакции). Обычно очередь может сохраняться на диске или любом другом долговременно хранилище. Таким образом использование очередей сообщений позволяет организовать транзакционность в рамках потока задач, основанного на акторах.

В рамках второй главы также была рассмотрена модель хранения данных (рисунок 2), в основе которой лежит классический метод организации доступа, не подразумевающий разделения ресурсов (shared-nothing approach).

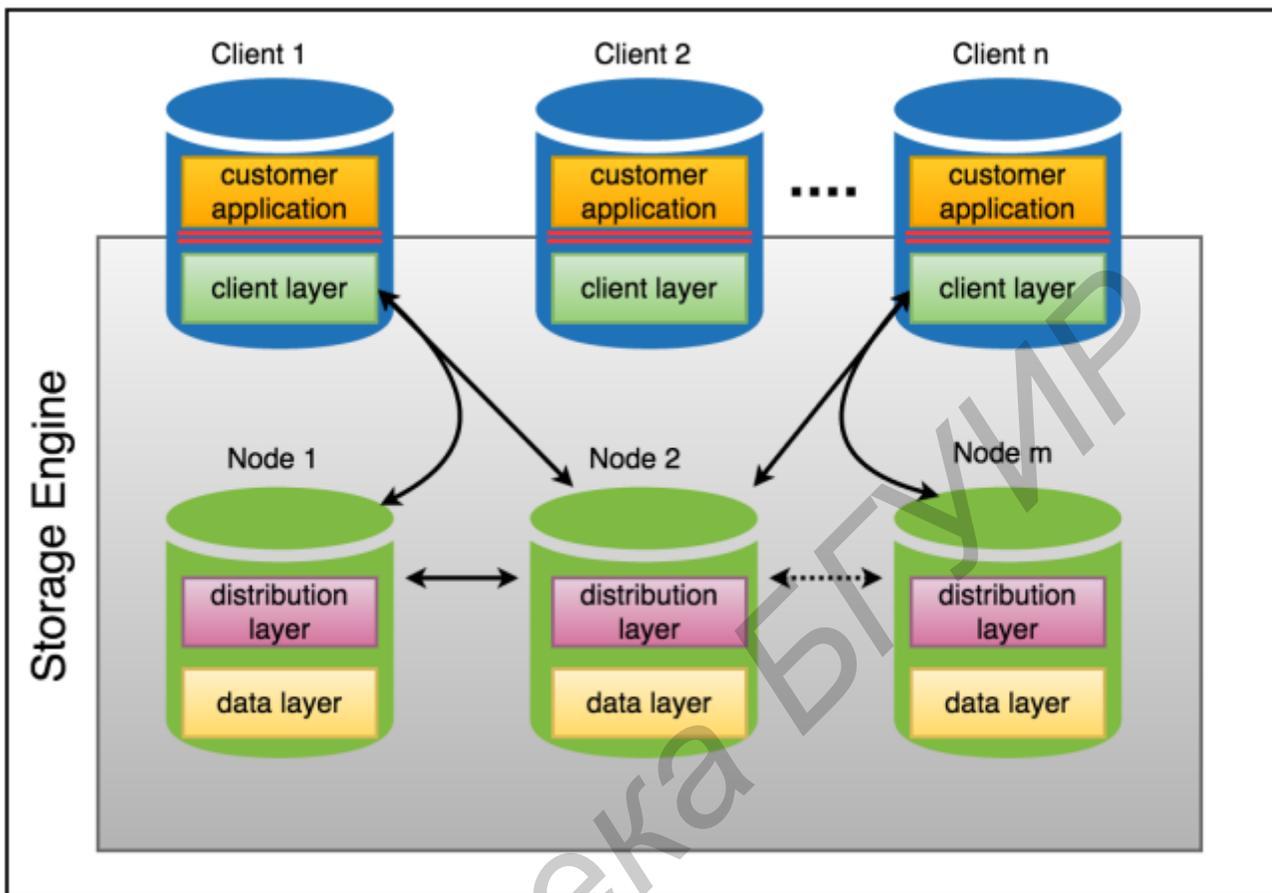


Рисунок 2 – Общая архитектура модели хранения данных

При данном подходе, как и в традиционных базах данных, существует разделение между клиентом и сервером. Клиент расположен на пользовательских машинах рядом с приложением и тесно с ним связан. Особенностью же подхода является использование балансировки нагрузки на стороне клиента, что приводит к значительному снижению времени на выполнение транзакций и позволяет линейно масштабировать систему.

В рамках **третьей главы** рассматриваются алгоритмы, используемые при построении моделей, обозначенных во второй главе. К таким алгоритмам относятся ГПСЧ «вихрь Мерсенна», алгоритм хеширования SHA-1, криптографические алгоритмы RSA и PBKDF, а также алгоритм согласованного хеширования и алгоритм консенсуса, используемые для организации масштабируемости и отказоустойчивости. В настоящее время наиболее подходящим для практической реализации является Raft - алгоритм консенсуса для управления реплицированными логами. В его основе лежит реплицированный конечный автомат.

Raft осуществляет консенсус за счёт того, что в первую очередь выбирает лидера, отвечающего за управление реплицированным логом. Лидер принимает запросы от клиентов, реплицирует их на серверах и сообщает серверам, когда можно безопасно выполнить операцию над записью в конечном автомате. В том случае, если лидер выпадает из сети, осуществляется процедура выбора нового лидера.

В **четвёртой главе** описывается архитектура распределённого программного средства организации хранения и доступа к данным (рисунок 3). Она объединяет в себе современные подходы, используемые при проектировании распределённых программных средств. К ним относятся:

1. API-шлюз для предоставления подходящего интерфейса для различных типов клиентов, а также позволяющим поддерживать гранулярность API микросервисов.
2. Микросервисная архитектура, применяемая для организации масштабируемости и гранулярности.
3. Очередь сообщений и акторы для организации передачи и обработки информации.

Структурно в распределённом программном средстве хранения и передачи данных можно выделить следующие подсистемы:

1. Клиентские приложения. В настоящее время в качестве клиентских приложений могут выступать веб-приложения, настольные приложения, мобильные приложения или боты.
2. API-шлюз, используемые клиентскими приложениями для выполнения функций по модификации и передаче данных.
3. Микросервисы, отвечающие за фактическое выполнение бизнес-задач, используемых в клиентских приложениях.
4. Очередь сообщений, представляющая собой промежуточный слой между микросервисами и хранилищем информации.
5. Система хранения пользовательских данных.
6. Подсистемы логирования, безопасности, кеширования, аналитики, используемые в подсистемах 1-5.

Следует отметить, что при данном архитектурном подходе построением клиентских приложений может заниматься любой желающий, имеющий доступ к API-шлюзу. Концентрация основной функциональности в рамках микросервисов распределённой системы призвано максимально упростить их разработку и использование.

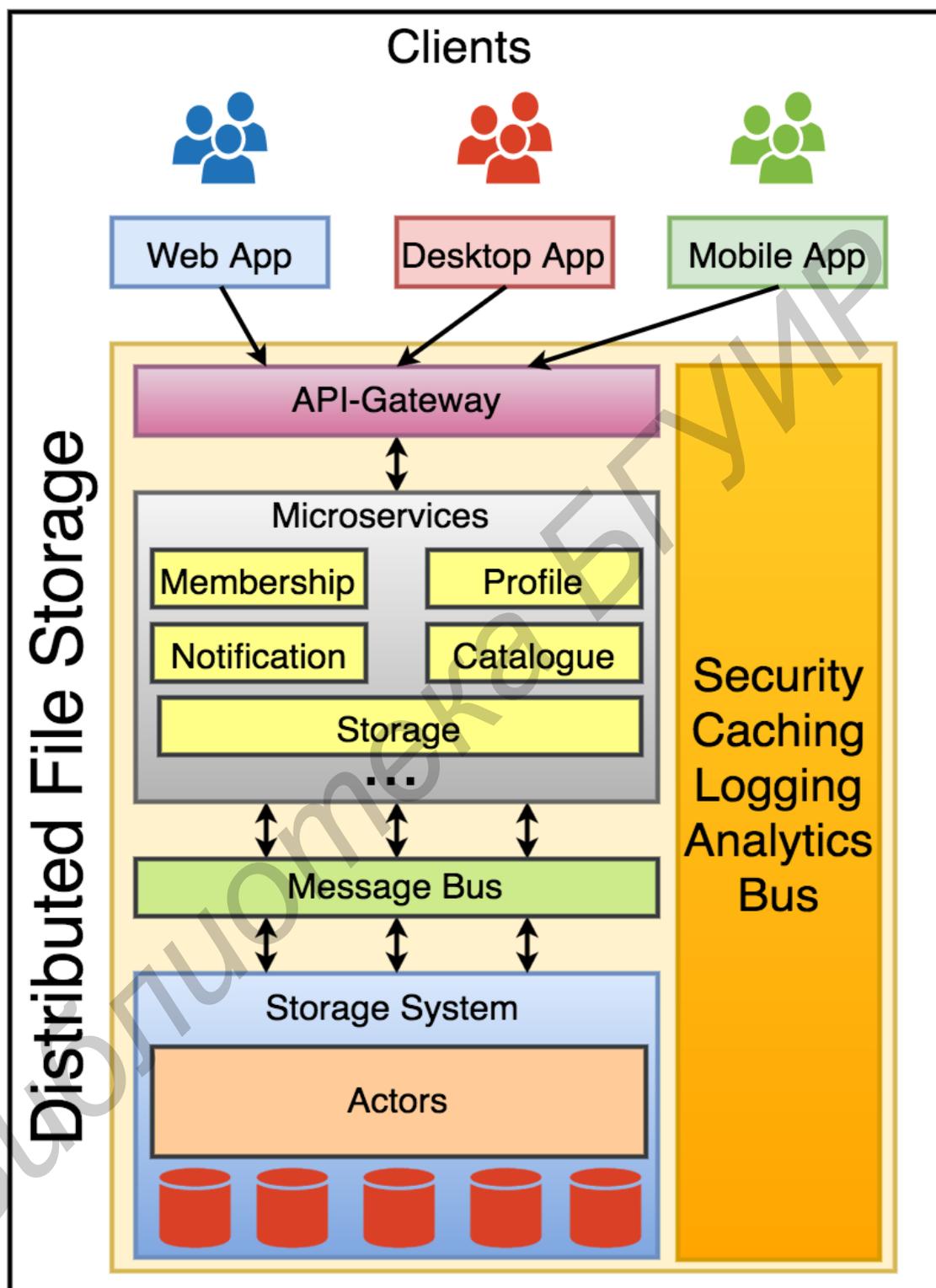


Рисунок 3 – Детализированная архитектура программного средства

Долговременное хранилище данных представляет собой NoSQL решение, модель которого представлена на рисунке 2. Отсутствие схемы и денормализация

позволяют ускорить операции чтения и записи в хранилище, однако приводит к согласованности в конечном счёте. Базовая структура хранения информации о пользовательских файлах представлена на рисунке 4.

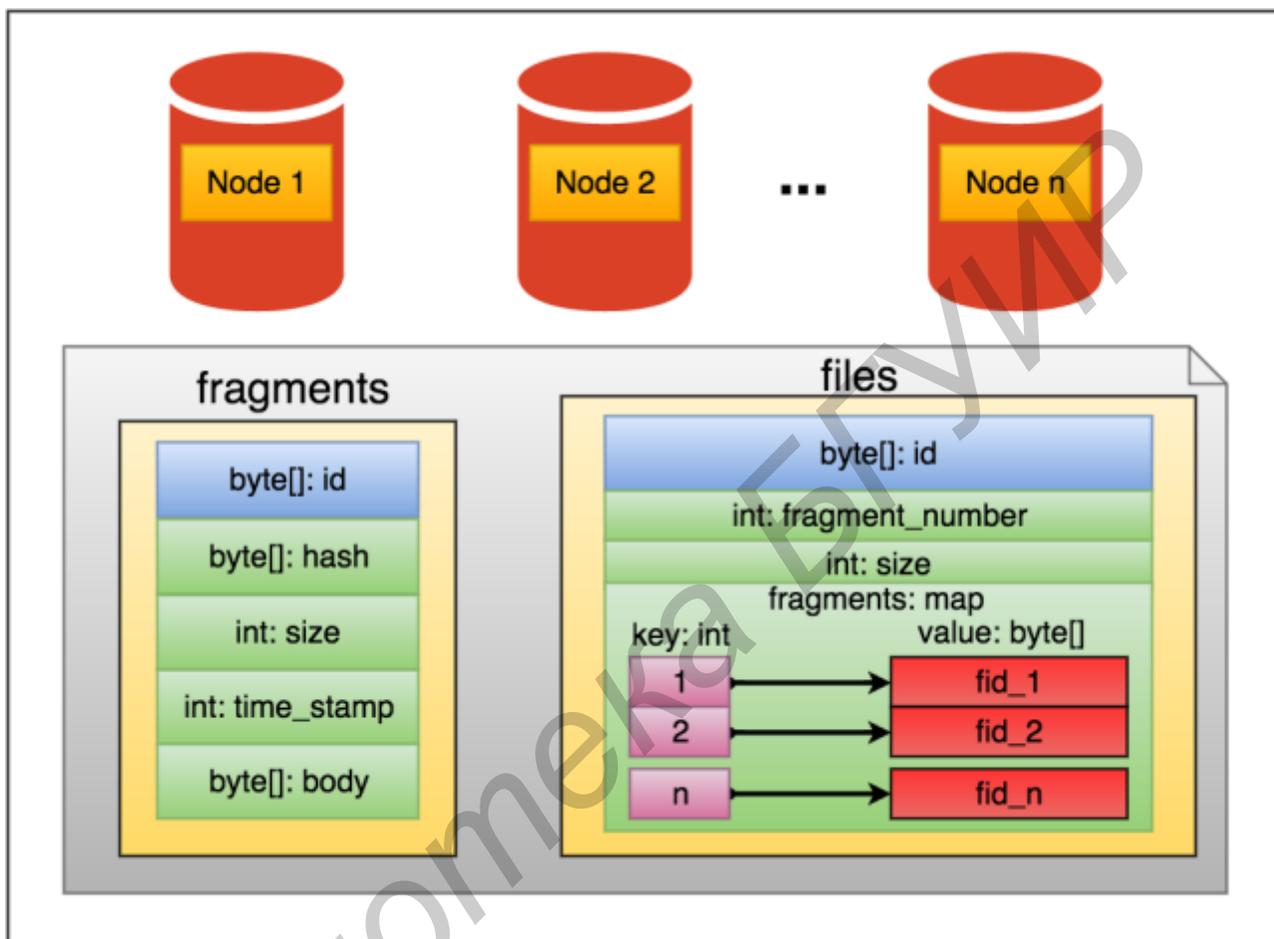


Рисунок 4 – Базовая структура долговременного хранилища

Пространство имён (namespace) состоит из двух ключевых множеств – множество (set) файлов и множество (set) фрагментов. Множество фрагментов отвечает за хранение частей файлов. Одно из ключевых предназначений данного множества – возможность осуществления операции дедупликации, снижающей общий объём размещаемой информации.

В **пятой главе** произведён анализ функционирования части компонент системы и произведена их оценка. Использование представленного подхода позволяет снизить объём хранимой информации. Величина уменьшения размера напрямую зависит от структуры данных и размера самого фрагмента.

Структура и набор тестовых данных представлены в таблице 1. Там же можно найти величину, на которую было произведено уменьшение объёма хранимой информации при заданном подходе (рисунок 5).

Таблица 1 – Результаты анализа данных на наличие дублирующих фрагментов

Категория файла	Общий объём, байт	Количество фрагментов	Уменьшение объёма, байт	Процент уменьшения, %
Музыка	389606949769	11891572	207677344	0.053
Видео	139435689106	4204317	1715544046	1.230
Изображения	11071211564	322613	598062324	5.402
Документы	50958288450	1462425	3974097841	7.799
Другое	70494966937	2112199	1407223115	1.996
Общее	661567105826	19993126	7902604670	

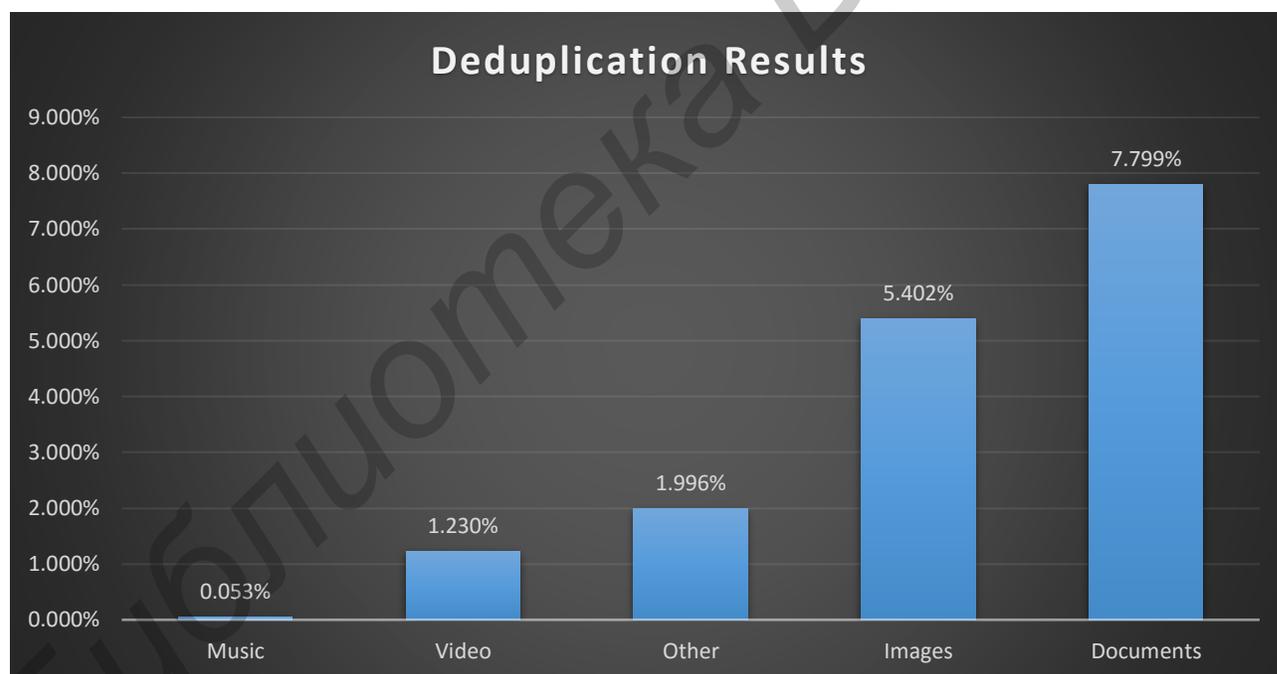


Рисунок 5 – Результаты дедупликации по категориям

На основании проведённых тестов можно сделать вывод о том, что добавление дополнительных проверок ожидаемо негативно сказывается на производительности. Однако разница находится в пределах 15%. Ввиду того, что разрабатываемое программное средство не предъявляет важных требований ко

времени выполнения операций, то данное снижение производительности не является критичным.

Обработкой и передачей информации в рамках системы занимается множество акторов. Физически их расположение является прозрачным для внешнего мира и определяется самим акторным модулем.

В качестве задач использовалось вычисления хеша передаваемого фрагмента при помощи алгоритма MurMurHash3 128. Общее количество акторов – 900. Результаты тестирования обработки запросов при добавлении узлов с акторами представлена на рисунке 6.

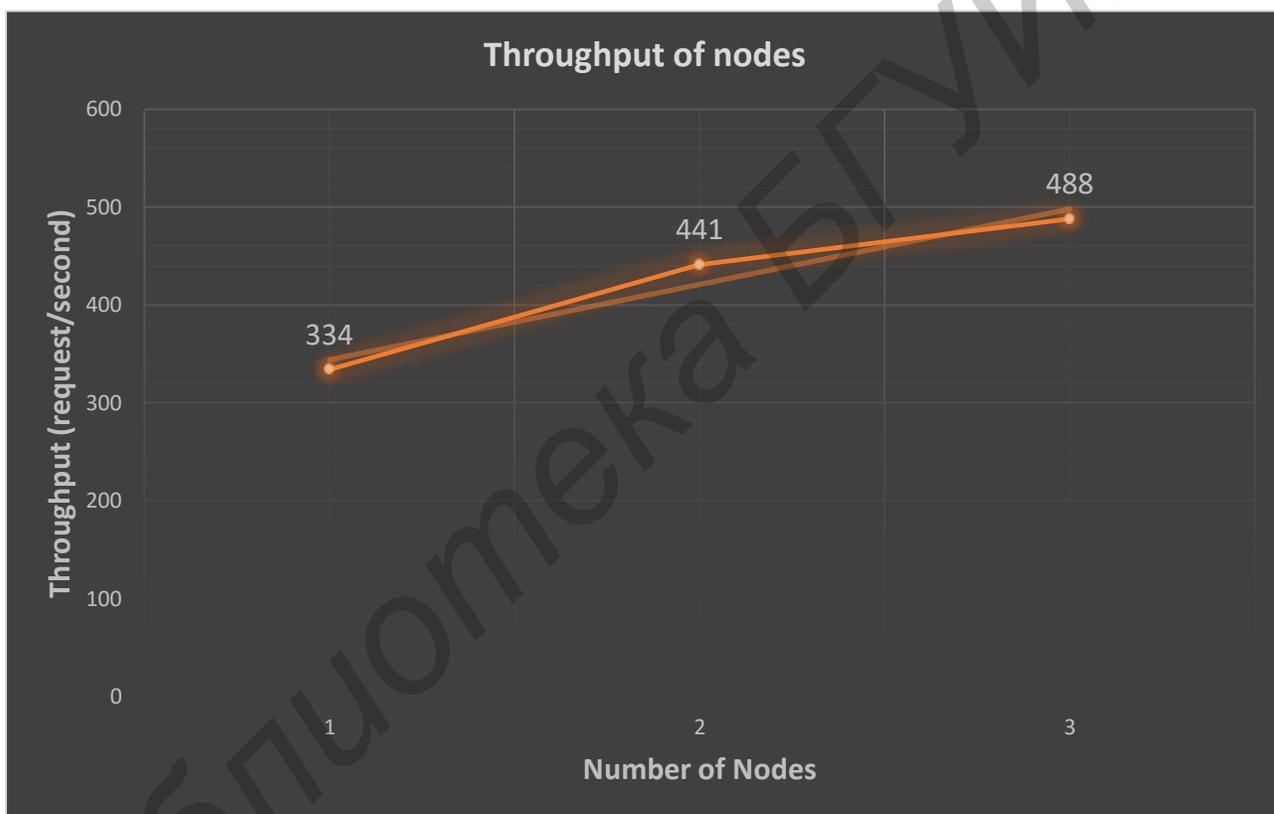


Рисунок 6 – Зависимость пропускной способности от количества узлов

Изоляция акторов между собой означает, что аргументы межакторного взаимодействия должны полностью копироваться между сущностями. Это добавляет дополнительные накладные расходы на время и память.

ЗАКЛЮЧЕНИЕ

1. Предложена модель хранения данных, обеспечивающая уменьшение объёма хранимых данных. Величина уменьшения зависит от структуры данных и размера фрагмента. В качестве недостатка выступает уменьшение количества операций чтения и записи в хранилище.

2. Предложена модель передачи, обработки данных и доступа к данным, в основе которой лежит акторная модель и транзакционные очереди. В качестве преимущества использования акторов можно назвать практически линейный рост числа выполняемых задач при добавлении вычислительных узлов. В качестве недостатка выступают дополнительные накладные расходы по времени и памяти ввиду полного копирования тела сообщений.

3. Предложена многокомпонентная архитектура решений, призванная решить задачи по организации хранения и доступа к данным, в основе которой лежат описанные модели, а также модель открытости.

Рекомендации по практическому использованию результатов

В общем и целом, модели, методы и программные компоненты, разработанные в рамках данной диссертационной работы, позволят облегчить построение распределённых программных средств и могут быть использованы в компаниях для организации хранения и доступа к данным.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

[1] Бут-Гусаим, Р.А. Использование акторной модели и очередей сообщений при построении распределённых систем / Р.А. Бут-Гусаим // Сборник научных трудов международной научно-практической конференции молодых учёных «Научно-технический прогресс и современное общество» - Москва, 2017