

На правах рукописи



СОТНИКОВ Дмитрий Владимирович

**УПРАВЛЕНИЕ БОЛЬШИМИ ДАННЫМИ ОБЛАЧНЫХ СЕРВИСОВ
НА ОСНОВЕ МНОГОСТАДИЙНЫХ АЛГОРИТМОВ И СРЕДСТВ ИХ
ДИНАМИЧЕСКОГО ПЕРЕРАСПРЕДЕЛЕНИЯ**

Специальность: 2.3.5. Математическое и программное обеспечение
вычислительных систем, комплексов и
компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Воронеж – 2026

Работа выполнена в ФГБОУ ВО «Воронежский государственный технический университет».

Научный руководитель: **Кравец Олег Яковлевич**, доктор технических наук, профессор

Официальные оппоненты: **Ковалев Игорь Владимирович**, доктор технических наук, профессор, ФГАОУ ВО «Сибирский федеральный университет» (г. Красноярск), профессор кафедры программной инженерии

Перепелкин Дмитрий Александрович, доктор технических наук, профессор, ФГБОУ ВО "Рязанский государственный радиотехнический университет им. В.Ф.Уткина", декан факультета вычислительной техники

Ведущая организация: **АО «Научно-исследовательский институт вычислительных комплексов им. М. А. Карцева» (г. Москва)**

Защита состоится «15» мая 2026 года в 14 часов в конференц-зале на заседании диссертационного совета 24.2.286.04, созданного на базе ФГБОУ ВО «Воронежский государственный технический университет», по адресу: г. Воронеж, Московский просп., д. 14, ауд. 216.

С диссертацией можно ознакомиться в научно-технической библиотеке ФГБОУ ВО «Воронежский государственный технический университет» и на сайте <https://cchgeu.ru>.

Автореферат разослан «30» марта 2026 г.

Ученый секретарь
диссертационного совета



Гусев Константин Юрьевич

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. За последние годы большие данные стали новой парадигмой для обработки и анализа огромных объемов данных. Обработка больших данных была объединена с сервисными и облачными вычислениями, что привело к появлению нового класса сервисов, получившего название “Большие сервисы”. Для удовлетворения сложных и разнородных потребностей пользователей в эпоху больших данных повторное использование сервисов является естественным и эффективным средством, которое помогает организовать их работу для предоставления больших сервисов по требованию клиентов. Актуальна задача управления большими данными облачных сервисов, интерес представляет и задача их компоновки. Большой вклад в разработку методов и средств управления большими данными облачных сервисов внесли Кузнецов С.О., Arndt H., Chiheb F., Gai K., Hossain M.S., Liu X., Mezni H., Sellami M.

Одной из актуальных предметных областей задач управления большими данными является компоновка реентерабельных больших сервисов. Интерес представляет и динамическое распределение больших данных по сервисам. С точки зрения интуитивного понимания, чем меньше объем знаний, тем больший резерв производительности облачных систем будет создан. Важной является и интеллектуальная кластеризация гетерогенных данных, качество которой отражает способность облачных систем к эффективной обработке данных.

Таким образом, актуальность темы диссертационного исследования продиктована необходимостью разработки специальных средств управления большими данными облачных сервисов на основе реализации многостадийных алгоритмов и процедур динамического их перераспределения.

Тематика диссертационной работы соответствует научному направлению ФГБОУ ВО «Воронежский государственный технический университет» «Вычислительные комплексы и проблемно-ориентированные системы управления».

Целью работы является разработка методов и средств управления большими данными облачных сервисов на основе многостадийных алгоритмов и динамического перераспределения данных.

Задачи исследования. Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ проблем управления большими данными облачных сервисов на основе многостадийных алгоритмов и динамического перераспределения данных.
2. Разработать алгоритм расширения хранилища больших сервисов в различных облачных зонах, обеспечивающий оценку близости формальных концепций, которые объединяют эти сервисы и источники данных.
3. Создать алгоритм компоновки больших сервисов, обеспечивающий отбор кандидатов, их комбинацию и оптимальных выбор больших сервисов, отвечающий требованиям QoS, качества данных и безопасности.
4. Предложить архитектуру динамической системы распределения дан-

ных, обеспечивающую регулирование распределения данных по каждому узлу хранения в режиме реального времени.

5. Разработать графическую модель интеграции принятия решений в большие данные, обеспечивающую выделение трех уровней больших данных, которые необходимо учитывать при разработке их проекта: данных, анализа и принятия решений.

6. Разработать архитектуру программной системы оптимизации больших данных от датчиков в Интернете вещей, реализующую уменьшение доли дубликатов и несоответствий в данных.

Объект исследования: процессы управления большими данными облачных сервисов на основе многостадийных алгоритмов и динамического перераспределения данных.

Предмет исследования: структура математического и программного обеспечения процессов управления большими данными облачных сервисов на основе многостадийных алгоритмов и динамического перераспределения данных.

Методы исследования. При решении поставленных в диссертации задач использовались методы теории вероятностей, теории принятия решений, а также методы объектно-ориентированного программирования.

Тематика работы соответствует следующим пунктам паспорта специальности 2.3.5 «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»: п. 4. «Интеллектуальные системы машинного обучения, управления базами данных и знаний, инструментальные средства разработки цифровых продуктов»; п. 9. «Модели, методы, алгоритмы, облачные технологии и программная инфраструктура организации глобально распределенной обработки данных».

Научная новизна работы. В диссертации получены следующие результаты, характеризующиеся научной новизной:

- алгоритм расширения хранилища больших сервисов в различных облачных зонах, отличающийся представлением в виде семейства решеток и использованием сходства по Жакарду экземпляров и источников данных и обеспечивающий оценку близости формальных концепций, которые объединяют эти сервисы и источники данных;

- алгоритм компоновки больших сервисов, отличающийся учетом качества данных (QoD) и определением набора формальных понятий, которые объединяют запрашиваемые сервисы и обеспечивающий отбор кандидатов, их комбинацию и оптимальных выбор больших сервисов, отвечающий требованиям QoS, QoD и безопасности и улучшающий качество итогового большого сервиса в среднем на 3.4%;

- архитектура динамической системы распределения данных, отличающаяся использованием «жадного» алгоритма сокращения миграции данных с динамическим выбором точки данных в перегруженном узле хранения с максимальной нагрузкой и обеспечивающая регулирование распределения данных по каждому узлу хранения в режиме реального времени;

- графическая модель интеграции принятия решений в большие дан-

ные, отличающаяся использованием наборов данных с новыми характеристиками, жизненного цикла анализа данных, технологий, аналитических методов, понимания и принятия решений и обеспечивающая выделение трех уровней больших данных, которые необходимо учитывать при разработке их проекта: данных, анализа и принятия решений.

- архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей, отличающаяся итерационным распределением больших данных на основе упорядочивания объектов и ссылок и реализующая уменьшение доли дубликатов и несоответствий в наборе данных в среднем на 12%.

Теоретическая и практическая значимость исследования заключается в разработке моделей и алгоритмов управления большими данными облачных сервисов на основе многостадийных алгоритмов и динамического перераспределения данных.

Теоретические результаты работы могут быть использованы в проектных и научно-исследовательских организациях, занимающихся проектированием программных систем с облачными базами больших данных.

Положения, выносимые на защиту

1. Алгоритм расширения хранилища больших сервисов в различных облачных зонах обеспечивает оценку близости формальных концепций, которые объединяют эти сервисы и источники данных.

2. Алгоритм компоновки больших сервисов обеспечивает отбор кандидатов, их комбинацию и оптимальных выбор больших сервисов, отвечающий требованиям QoS, качества данных и безопасности и улучшает качество итогового большого сервиса в среднем на 3.4%.

3. Архитектура динамической системы распределения данных обеспечивает регулирование распределения данных по каждому узлу хранения в режиме реального времени.

4. Графическая модель интеграции принятия решений в большие данные обеспечивает выделение трех уровней больших данных, которые необходимо учитывать при разработке их проекта: данных, анализа и принятия решений.

5. Архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей реализует уменьшение доли дубликатов и несоответствий в наборе данных в среднем на 12%.

Результаты внедрения. Основные результаты внедрены в ООО М-Сервис (г. Воронеж) при проектировании систем управления гетерогенными программными системами, в учебный процесс Воронежского государственного технического университета в рамках дисциплин: «Вычислительные машины, системы и сети», «Информационные сети и телекоммуникационные технологии», а также в рамках курсового и дипломного проектирования.

Апробация работы. Основные положения диссертационной работы докладывались и обсуждались на следующих конференциях: XXVIII-th - XXXI-th International Open Science Conference «Modern informatization problems» (Yelm, WA, USA, 2023-2026); XII Всероссийской научно-практической

конференции «Решение» (Пермь, 2023); II Всероссийской научной конференции «Достижения науки и технологий-ДНиТ-II-2023» (Красноярск, 2023), Международной молодежной научной школе «Оптимизация и моделирование в автоматизированных системах» (Воронеж, 2023); Международной научно-практической конференции, «Интеллектуальные информационные системы» (Воронеж, 2024); VI Всероссийской научно-практической конференции «Информационные технологии в экономике и управлении» (Махачкала, 2024), а также на научных семинарах кафедры автоматизированных и вычислительных систем ВГТУ (2023-2026 гг.).

Достоверность результатов обусловлена корректным использованием теоретических методов исследования и подтверждена результатами сравнительного анализа данных вычислительных и натуральных экспериментов.

Публикации. По результатам диссертационного исследования опубликовано 19 научных работ, в том числе 7 – в изданиях, рекомендованных ВАК РФ (из них 1 – в издании, индексируемом в WoS и одно свидетельство о регистрации программы для ЭВМ). В работах, опубликованных в соавторстве и приведенных в конце автореферата, лично автором получены следующие результаты: [7, 8] - алгоритм расширения хранилища больших сервисов в различных облачных зонах, отличающийся представлением в виде семейства решеток и использованием сходства по Жакарду экземпляров сервисов и источников данных; [2, 6, 19] - алгоритм компоновки больших сервисов, отличающийся учетом качества данных и определением набора формальных понятий, которые объединяют запрашиваемые сервисы; [5, 9, 12, 14] - архитектура динамической системы распределения данных, отличающаяся использованием «жадного» алгоритма сокращения миграции данных с динамическим выбором точки данных в перегруженном узле хранения с максимальной нагрузкой; [13, 16, 17, 18] - графическая модель интеграции принятия решений в большие данные, отличающаяся использованием наборов данных с новыми характеристиками, жизненного цикла анализа данных, технологий, аналитических методов, понимания и принятия решений; [1, 3, 4, 12] - архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей, отличающаяся итерационным распределением больших данных на основе упорядочивания объектов и ссылок.

Структура и объем работы. Диссертационная работа состоит из введения, пяти глав, заключения, списка литературы из 185 наименований. Работа изложена на 195 страницах.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность исследования, сформулированы его цель и задачи, научная новизна и практическая значимость полученных результатов, приведены сведения об апробации и внедрении работы.

В первой главе исследуются проблемы управления большими данными облачных сервисов на основе многостадийных алгоритмов и динамического перераспределения данных. Отмечено, что повысить эффективность такого управления можно путем применения алгоритма расширения храни-

лица больших сервисов в различных облачных зонах, алгоритм компоновки больших сервисов, отличающийся учетом качества данных и определением набора формальных понятий, выбора архитектуры динамической системы распределения данных, интеграции принятия решений в большие данные, архитектуры программной системы оптимизации больших данных от датчиков в Интернете вещей. Результат анализа потребовал формализации данных задач, а также алгоритмизации их решения с учетом особенностей, отраженных на рис. 1. Сформулирована цель и задачи исследования.

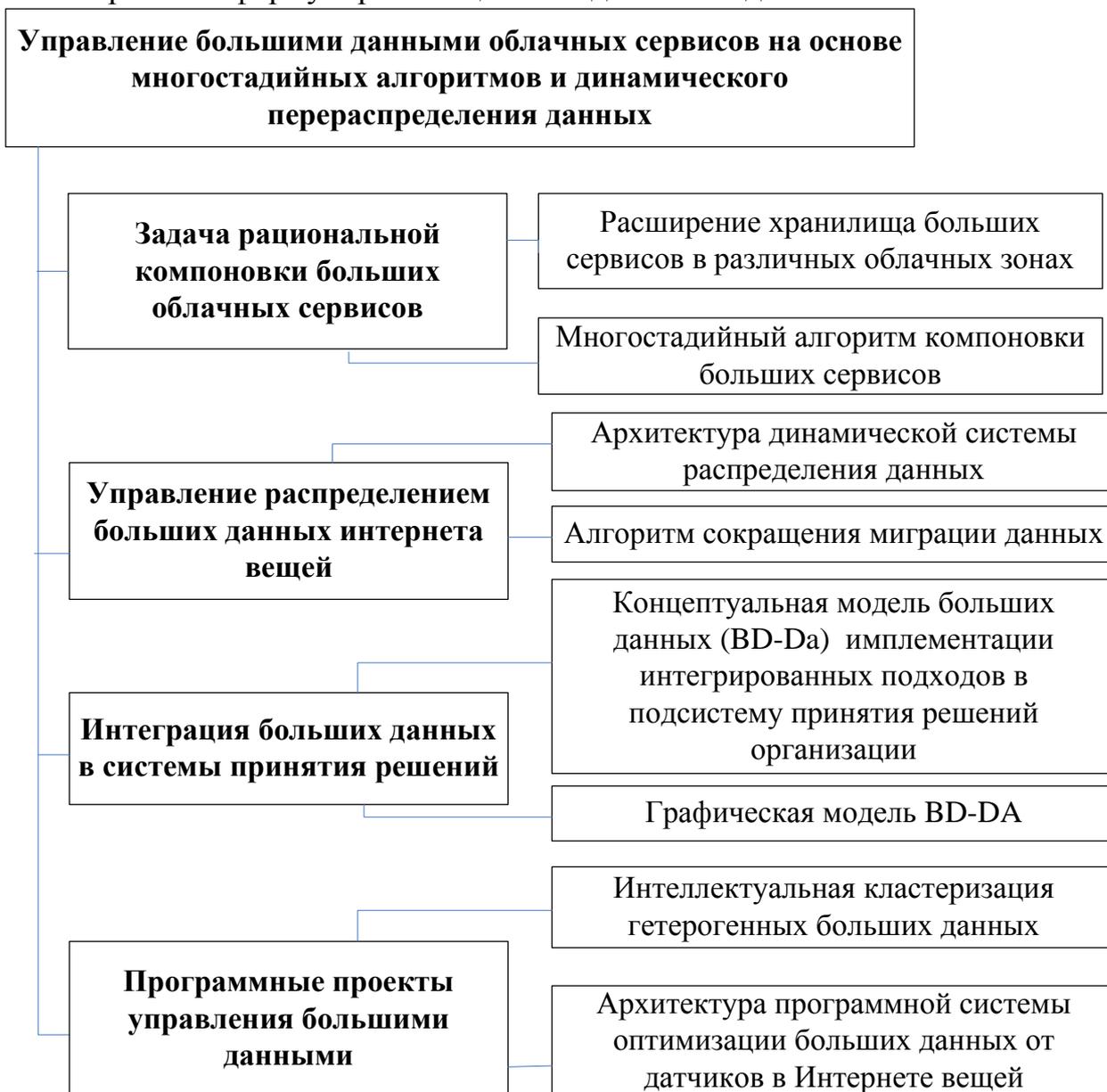


Рис. 1. Дизайн исследования

Необходимо провести экспериментальные исследования для доказательства эффективности выбранного подхода к составлению большого сервиса с точки зрения состава с учетом QoD, масштабируемости и устранения нарушений безопасности.

Вторая глава посвящена задаче рациональной компоновки больших облачных сервисов. В качестве модели абстрагирования больших данных и

сокрытия их сложности в больших сервисах (BS) рассматриваются как управляемая интеграция массивной, сложной серии разнородных сервисов, ориентированных на большие данные. Такая сложная и масштабная сервисная экосистема способна обрабатывать огромные объемы данных и предлагать их в качестве сервисов по требованию клиентов.

Будем считать, что качество BS зависит не только от традиционных параметров QoS (например, надежности, доступности, стоимости, безопасности и т.д.), но и от качества источников данных (QoD), используемых компонентами BS. Фактически, оценка полноты, точности и своевременности этих источников данных является важным шагом для принятия решения о способности сервиса-кандидата участвовать в больших облачных сервисах (BSCo), даже если он отличается высоким качеством обслуживания.

Определение 2.1. Семейство реляционных контекстов (RCF) - это пара (K, R) , где K - набор объектно-атрибутивных контекстов $K_i=(O_i, A_i, I_i)$; R - набор объектно-объектных контекстов $R_j=(O_k, O_l, I_j)$; (O_k, O_l) - наборы объектов формальных контекстов $(K_k, K_l) \in K$; $I_j \subseteq O_k \times O_l$; K_k - исходный контекст или контекст предметной области; K_l - целевой контекст/контекст диапазона.

Определение 2.2. Семейство решеток (LF) - набор решеток, которые выводятся из набора формальных контекстов после обогащения их реляционными атрибутами. Каждая решетка представляет собой набор кластеров, называемых формальными понятиями. Формальное понятие - это пара (E, I) , где E - это набор объектов, называемый экстендом, I - это набор атрибутов, называемых намерением, и все объекты в E находятся в отношении R со всеми атрибутами в I .

Определение 2.3. Большой сервис $S = \{s_1, s_2, \dots, s_n\}$ - управляемая интеграция массивной, сложной серии сервисов, ориентированных на большие данные. Каждый сервис S_i представляет собой объединение множества m задач (или абстрактных сервисов) $S_i = \{t_1, \dots, t_m\}$. Каждая задача t_s может быть реализована одним сервисом из определенного набора сервисов-кандидатов k_i $C_i = \{s^i_1, \dots, s^i_{k_i}\}$. Каждая служба-кандидат s^i использует блок данных ch_t в наборе конфиденциальных источников данных $CH_z = \{ch_1, \dots, ch_t\}$. Каждая служба s^i размещается у одного или нескольких поставщиков облачных сервисов $CP = \{cp_1, cp_2, \dots, cp_t\}$. Сервисы во множестве C_i функционально схожи, но могут отличаться уровнем качества обслуживания.

Служба S_i имеет набор атрибутов QoS $Q^s = \{q^s_1, q^s_2, \dots, q^s_t\}$ (например, надежность, доступность, время выполнения и т.д.) и набор атрибутов QoD $Q^d = \{q^d_1, q^d_2, \dots, q^d_t\}$ (например, полнота, точность, своевременность и т.д.), обозначающий уровни качества каждого используемого источника данных службой S_i . Исходя из чувствительности источников данных, каждая служба имеет уровень параметра L-Severity, который отражает серьезность утечек данных при использовании их фрагментов.

Определение 2.4. Пусть $ST(a_1, \dots, a_n)$ - исходная таблица, $L(b_1, \dots, b_m)$ - «утекшая» таблица с $\{b_1, \dots, b_m\} \subseteq \{a_1, \dots, a_n\}$, $S = \{s_1, \dots, s_m\} \subseteq A$ - множество из чувствительных атрибутов в модели данных (b_1, \dots, b_m) и

$DM=(T,I,HR,IR,SL,PL)$. Задана запись $r \hat{=} RD^{L(b_1, \dots, a_m)}$ в $L(b_1, \dots, a_m)$ и $DF_r^{ST(a_1, \dots, a_n)}$, чувствительность записи r равна:

$$RSENS_r = DF_r^{ST(a_1, \dots, a_n)} \mathring{a}_{s_i \hat{=} S} NS(s_i[r])$$

где NS - чувствительность узла в модели данных, которая соответствует значению $s_i[r]$ чувствительного атрибута s_i .

Определение 2.5. Пусть $ST(a_1, \dots, a_n)$ - исходная таблица, а $L(b_1, \dots, b_m)$ - «утерянная» таблица с $\{b_1, \dots, b_m\} \subseteq \{a_1, \dots, a_n\}$. Учитывая чувствительность записи $RSENS_r$ для каждой записи $r \hat{=} R^{L(a_1, \dots, a_n)}$, параметр L-Severity из (b_1, \dots, b_m) вычисляется следующим образом:

$$L\text{-Severity}_L = \mathring{a}_{r \hat{=} R^{L(b_1, \dots, b_m)}} RSENS_r$$

Определение 2.6. Пусть $DS(s_1, \dots, s_n)$ - служба данных, осуществляющая доступ к источникам данных $D(a_1, \dots, a_n)$ с помощью $s_1, \dots, s_n \subseteq (a_1, \dots, a_n)$ и $L(b_1, \dots, b_m)$ - «утечки» записей при доступе к источникам данных с помощью $\{b_1, \dots, b_m\} \subseteq \{s_1, \dots, s_n\}$. Учитывая чувствительность $RSENS_r$ для каждой записи $r \hat{=} R^{L(s_1, \dots, s_n)}$, параметр L-Severity набора $BS=(ds_1, ds_2, \dots, ds_n)$ определяется следующим образом:

$$L\text{-Severity}(BS) = \mathring{a}_{ds_i \hat{=} BS} L\text{-Severity}(ds_i)$$

Правильное представление сущностей среды BS (сервисов, источников данных, поставщиков и т.д.) и их свойств является первым шагом на пути к эффективной $BSCo$. Этот процесс состоит из четырех основных этапов, как показано на рис. 2.1.

1. Фаззификация облака заключается в моделировании облачной среды с использованием набора нечетких формальных контекстов, которые помогают точно представлять отношения между объектами (сервисами, источниками данных и т.д.).

2. Генерация облачной решетки позволяет иерархически моделировать взаимосвязи между источниками данных, сервисами и их описаниями (QoS, QoD и т.д.).

3. Расширение структуры сервисов заключается в обогащении структуры сервисов сервисами данных и их описаниями, что позволяет группировать сервисы и их обычно используемые источники данных в небольшие кластеры, называемые формальными понятиями.

4. Сокращение хранилища BS направлено на устранение избыточных элементов и бесполезных описаний BS . В результате хранилище BS будет содержать всю необходимую и уместную информацию (сервисы, информацию о QoS, источники данных, QoD и серьезность данных) для процесса $BSCo$.

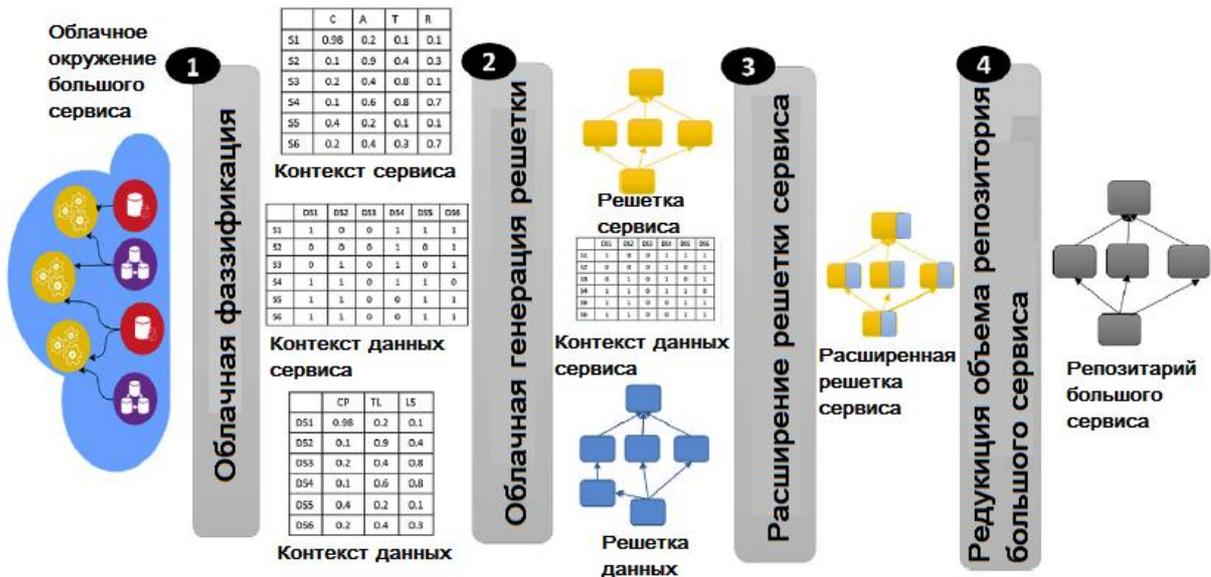


Рис. 2.1. Создание хранилища BS

Определение 2.7. Нечеткий контекст сервисов - это триплет $K^S=(S,Q,R)$, где S - набор объектов, представляющих сервисы, а Q - набор атрибутов QoS. Соотношение $R=S \in Q$ указывает на степень удовлетворенности конкретного атрибута QoS доступным сервисом.

Определение 2.8. Контекст нечетких данных представляет собой триплет $K^D=(D,Q,R)$, где D - набор объектов, представляющих источники данных, а Q - набор атрибутов, обозначающих показатели качества данных. Соотношение $R=D \in Q$ указывает, удовлетворяет ли источник данных определенному уровню качества для определенного атрибута QoS.

Определение 2.9. Формальный контекст привязки данных к сервису - это триплет $K^B=(S,D,R)$, где S - набор объектов, представляющих сервисы, а D - набор атрибутов, представляющих источники данных. Бинарное отношение $R=S \in D$ указывает, использует ли служба конкретный источник данных.

Увеличение размера решетки и сокращение объема хранилища состоит в обогащении каждой решетки служб в семействе решеток полезной информацией о потребляемых источниках данных (например, QoS, L-Severity). Результатом этого шага является набор расширенных решеток, представляющих среду BS. Представлен Алгоритм 2.1 расширения решетки (рис. 2.2).

Результатом работы алгоритма 2.1 является новая расширенная решетка, которая содержит только доступные сервисы высокого качества с наилучшими используемыми источниками данных.

Таким образом, представлен алгоритм расширения хранилища больших сервисов в различных облачных зонах, отличающийся представлением в виде семейства решеток и использованием сходства по Жакарду экземпляров сервисов и источников данных и обеспечивающий оценку близости формальных концепций, которые объединяют эти сервисы и источники данных.

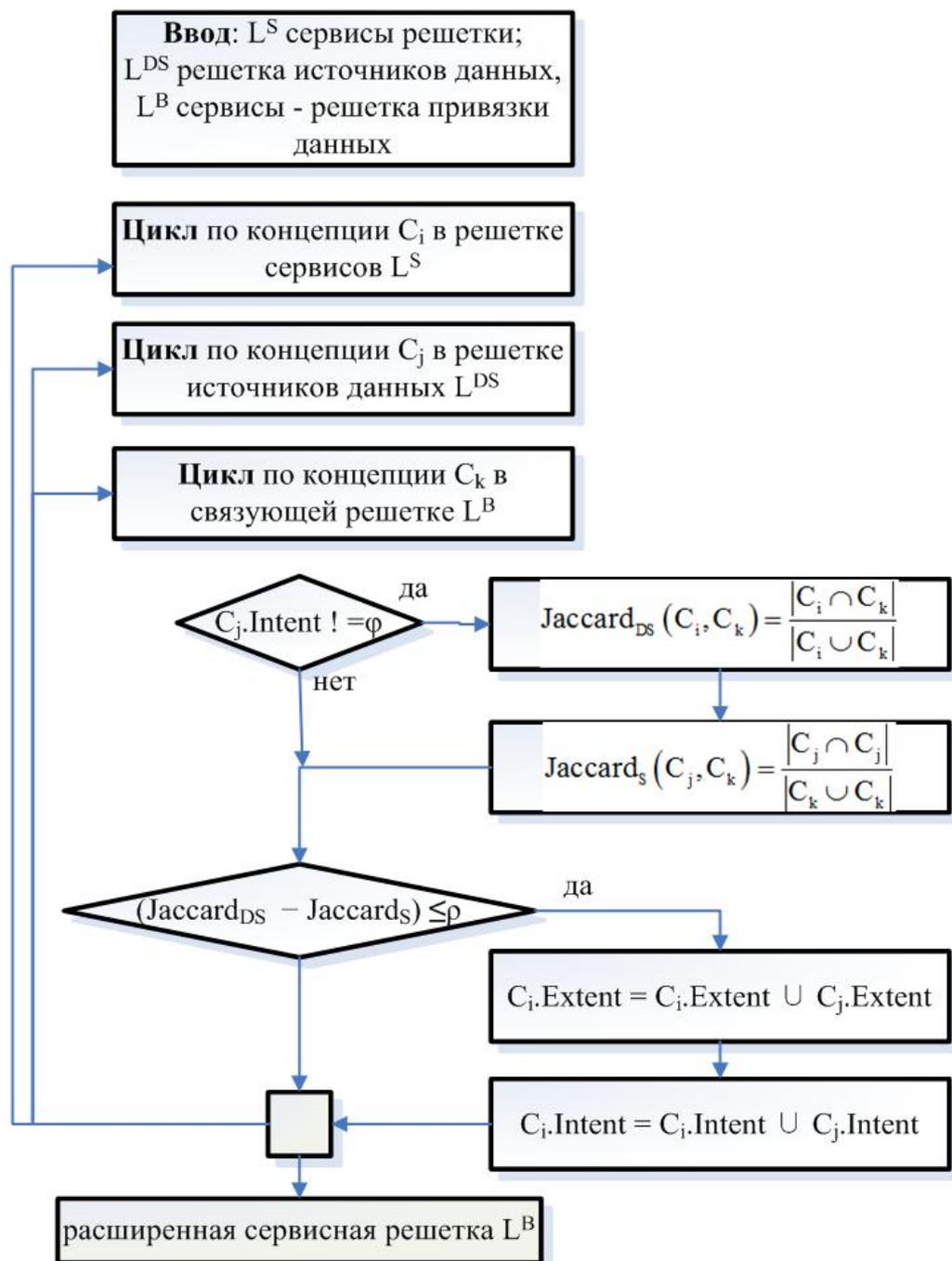


Рис. 2.2. Алгоритм 2.1 расширения решетки

Процесс компоновки больших сервисов (рис. 2.3) описан ниже:

1. Отбор сервисов-кандидатов: этот шаг заключается в просмотре репозитория BS (семейство расширенных решеток) с целью поиска сервисов, которые отвечают функциональным и нефункциональным требованиям пользователя, в мультиоблачной среде.

2. Комбинация сервисов BS: этот шаг состоит в поиске возможных комбинаций сервисов-кандидатов, которые были получены в результате первого шага. Затем запускается процесс ранжирования для вычисления оценки каждой комбинации сервисов на основе нескольких критериев (уровни QoS и QoD, степень сложности, стоимость взаимодействия между облаками и т.д.).

3. Оптимальный выбор BS: заключительный шаг направлен на выбор

оптимального плана компоновки, который не только отвечает требованиям QoS, QoD и безопасности, но и минимизирует количество источников данных и затраты на связь между задействованными облачными провайдерами.

Извлечение сервисов-кандидатов обеспечивается алгоритмом 2.2, который принимает в качестве входных данных набор расширенных решеток сервисов и пользовательский запрос. Запуская извлечение для каждой решетки, мы проверяем, удовлетворяется ли запрос пользователя сервисами, относящимися к формальным концепциям.

На этапе соединения больших сервисов сервисы-кандидаты вместе с их источниками данных, полученными в результате предыдущего шага, объединяются таким образом, чтобы они удовлетворяли функциональным требованиям, требованиям качества обслуживания и безопасности, указанным в запросе пользователя. Поскольку сервисы сгруппированы в соответствии с используемыми источниками данных, поиск подходящей структуры заключается в определении набора формальных понятий, которые объединяют запрашиваемые сервисы. Разработан алгоритм 2.3, который использует в качестве входных данных подиерархию расширенной решетки L^B и пользовательском запросе Q_{BS} . Результатом работы этого алгоритма является набор составленных оценок, которые позже будут оценены, чтобы вернуть ту, которая наберет наибольшее количество баллов.

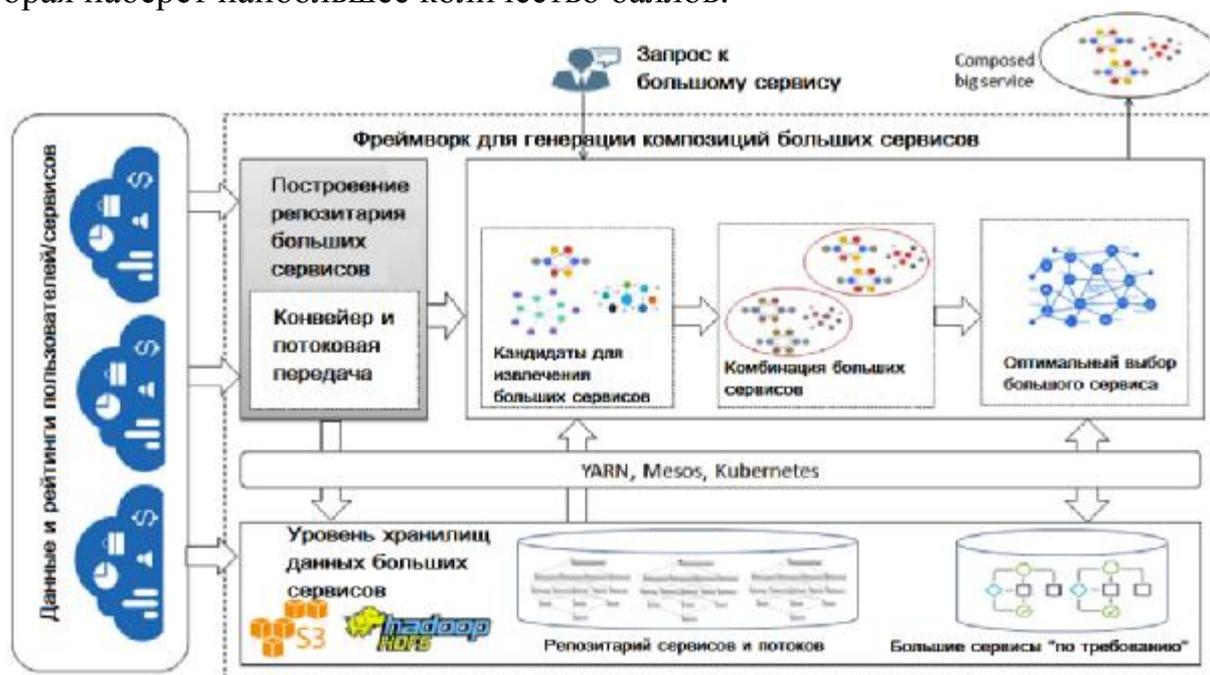


Рис. 2.3. Процесс компоновки веб-сервисов

Структура BS может быть найдена в нескольких комбинациях экстенгов, которые обеспечивают доступ к источникам данных с различными уровнями QoD и L-Severity. Поскольку существует компромисс между сокращением количества используемых источников данных и удовлетворением требований безопасности, применение нисходящего подхода к анализу, который позволяет найти наибольшее количество запрашиваемых сервисов при минимальном количестве формальных концепций, при максимальном уровне

качества источников данных, может позволить проанализировать минимальное количество формальные концепции, которые помогают сократить время вычислений сложной и перекрывающейся расширенной решетки. Проведены имитационные эксперименты.

Алгоритм 2.2. Извлечение сервисов-кандидатов

```

1: Input: Расширенная сервисная решетка  $L^B$ ,
пользовательский запрос  $Q_{BS}$  с требованиями QoS и QoD  $\tau$ 
2: Output: Субиерархия решетки сервисов-кандидатов  $L^B$ 
3:  $S_1 = \emptyset$ 
4: for each Концепции  $C_i$  в  $L^B$  do
5:   if  $(Q_{BS} - S_i) \cap C_i.Extent \neq \emptyset$  then
6:     if  $S_j \in Q_{BS}$  или  $L-severity(C_i.Intent) \leq T$  then
7:       удалить  $S_j$  из  $C_i.Extent$ 
8:     end if
9:     добавить  $C_i.Extent$  в  $S_1$ 
10:  else
11:    удалить  $C_i$  из  $L^B$ 
12:  end if
13: end for
14: Return Субиерархия решетки сервисов-кандидатов  $L^B$ 

```

Алгоритм 2.3. Композиция BS

```

1: Input: Решетчатая подиерархия сервисов-кандидатов
и источников данных  $L^B$ 
2: Output: Множество  $S^{BS}$  кандидатов BS
3:  $S^{BS} = \emptyset$ 
4: Сортировка в  $L^B$  формальных концепций в
порядке убывания размеров их экстенгов
5: while  $(L^B \neq \emptyset)$  do
6:    $S_1 = \emptyset$ 
7:    $D_1 = \emptyset$ 
8:   for each концепции  $C_i$  в  $L^B$  do
9:     if  $(Q_{BS} - S_1) \cap C_i.Extent \neq \emptyset$  then
10:      добавить  $C_i.Extent$  в  $S_1$ 
11:      add  $C_i.Intent$  to  $D_1$ 
12:      удалить  $C_i$  из  $L^B$ 
13:     if  $(S_1 = Q_{BS}$  and  $D_1 = Q_{BS}.data)$  then
14:        $S^{BS} = S^{BS} \cup S_1$ 
15:       break
16:     end if
17:   end for
18: end while
19: Return  $S^{BS}$ 
20:

```

Таким образом, представлен алгоритм компоновки больших сервисов, отличающийся учетом качества данных (QoD) и определением набора формальных понятий, которые объединяют запрашиваемые сервисы и обеспечивающий отбор сервисов-кандидатов, комбинацию сервисов и оптимальных выбор больших сервисов, отвечающий требованиям QoS, QoD и безопасности и улучшающий качество итогового большого сервиса в среднем на 3.4%.

Третья глава посвящена алгоритмизации управления распределением больших данных интернета вещей.

В соответствии с массовостью, пространственно–временной корреляцией, дисбалансом доступа и постоянной изменчивостью информации в Интернете вещей, для адаптации к ней необходим механизм распределения данных во временной области.

Архитектура распределенной системы баз больших данных реального времени состоит из четырех частей (рис. 3.1): узла управления, клиентского интерфейса узла хранения и порта сбора данных. На каждом узле хранения работает база данных в режиме реального времени.

На этапе инициализации узел управления распределяет точки данных по каждому узлу хранилища в соответствии с алгоритмом статического распределения и инициализирует глобальную таблицу распределения данных. На этапе эксплуатации узел хранения запускает модуль адаптивной обратной связи по нагрузке и регулирует временную область обратной связи в режиме реального времени в соответствии с нагрузкой, создаваемой обновлением данных. Узел управления запускает модуль динамического распределения данных, отслеживает информацию о загрузке, передаваемую каждым узлом

хранения, в режиме реального времени и регулирует распределение данных по каждому узлу хранения в режиме реального времени. Порт сбора данных синхронизируется с узлом управления для обеспечения согласованности глобального обновления распределения данных.

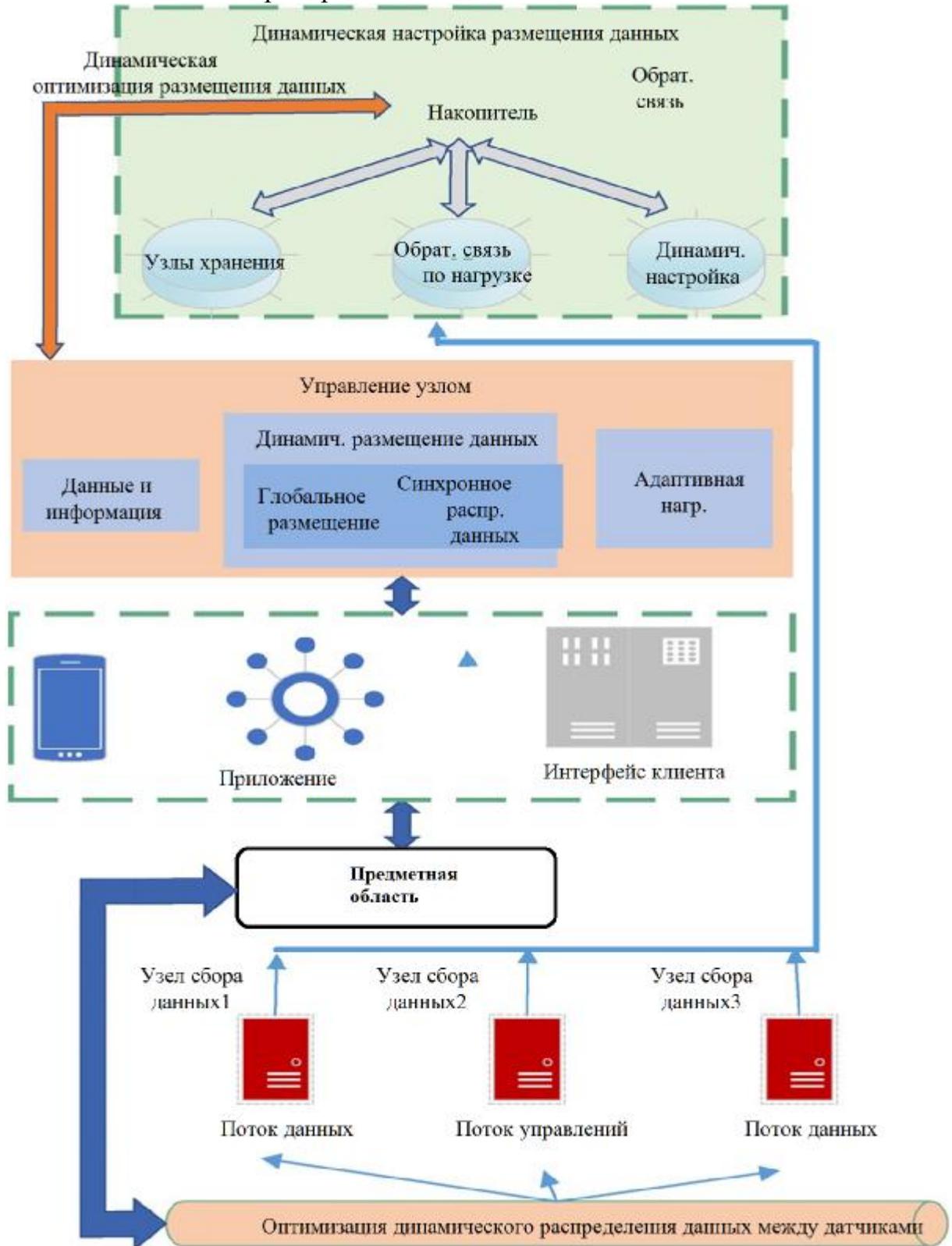


Рис. 3.1. Архитектура динамической системы распределения данных

Основная идея алгоритма для установки порога перегрузки узлов хра-

нения заключается в получении данных о нагрузке каждого узла хранения и вычислении среднего значения. Когда среднее значение невелико, допускается большая разница между пороговым значением нагрузки и средним значением, а также допускаются большие колебания нагрузки между каждым узлом. В этом случае перегрузки узлов нагрузки не будет. С увеличением нагрузки допустимый диапазон колебаний также уменьшается, то есть разница между допустимым порогом нагрузки и средним значением нагрузки уменьшается с увеличением среднего значения нагрузки. Наконец, когда средняя нагрузка становится слишком большой, чтобы максимально увеличить полезную информацию и уменьшить потребление, вызванное переносом нагрузки, устанавливается фиксированное значение разницы между порогом нагрузки и средним значением нагрузки. Эвристическая функция порога нагрузки определяется следующим образом:

$$K_M = \begin{cases} b + k \ln(e + e(K_{mean} - a)) \\ b + K_{mean} \end{cases}$$

где K_M - порог перегрузки узлов хранения, K_{mean} - средняя скорость загрузки каждого узла хранения, b - фиксированное пороговое значение нагрузки при слишком большой нагрузке, также известное как смещение порога перегрузки, обычно в пределах 6-11%, e - диапазон допустимых колебаний при низких нагрузках.

Пороговая функция и тренд среднего значения нагрузки показаны на рис. 3.2, где $a = 0 \dots 76$; $b = 0 \dots 06$; $e = 1 \dots 8$. Из рис. 3.2 видно, что эта функция лучше соответствует тренду изменения порогового значения узла хранения.

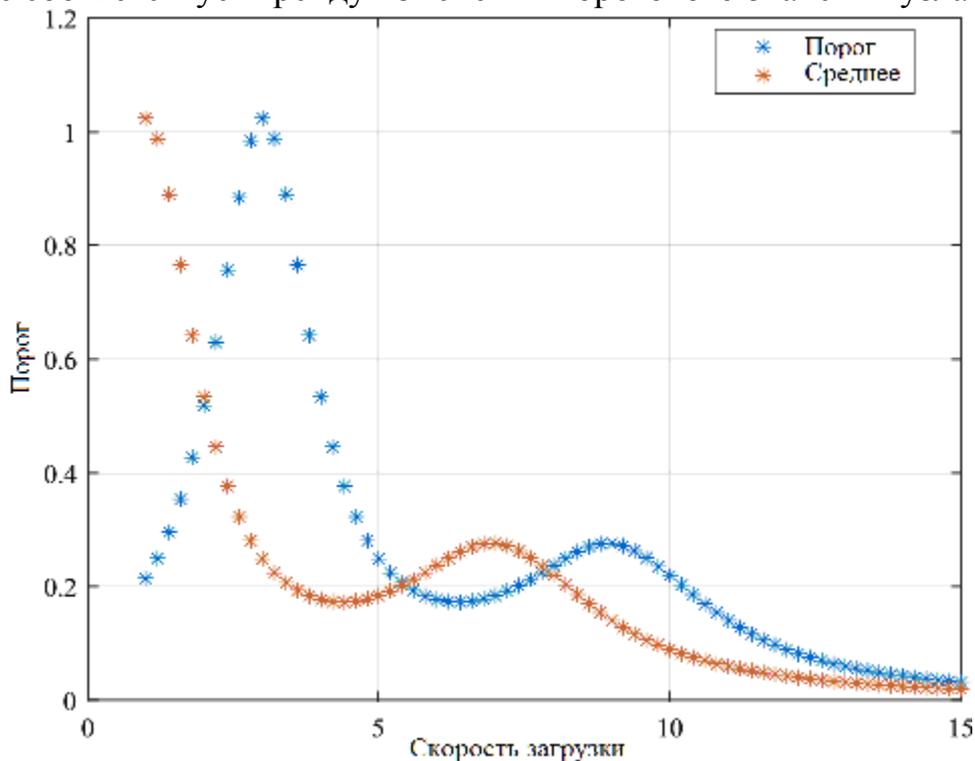


Рис. 3.2. Диаграмма порога нагрузки

Ключом к регулированию нагрузки на перегруженный узел хранения является сокращение миграции данных. В системе стоимость миграции каж-

дой точки данных одинакова, поэтому чем меньше количество перенесенных точек данных, тем ниже стоимость процесса настройки. В соответствии с «жадной» идеей, выберем точку данных в перегруженном узле хранения с максимальной нагрузкой и изменим точку на узел хранения с минимальной нагрузкой после того, как он перенесет нагрузку. Структурная схема соответствующей архитектуры приведена на рис. 3.3. Используется жадное правило для передачи данных узлу с малой загрузкой.

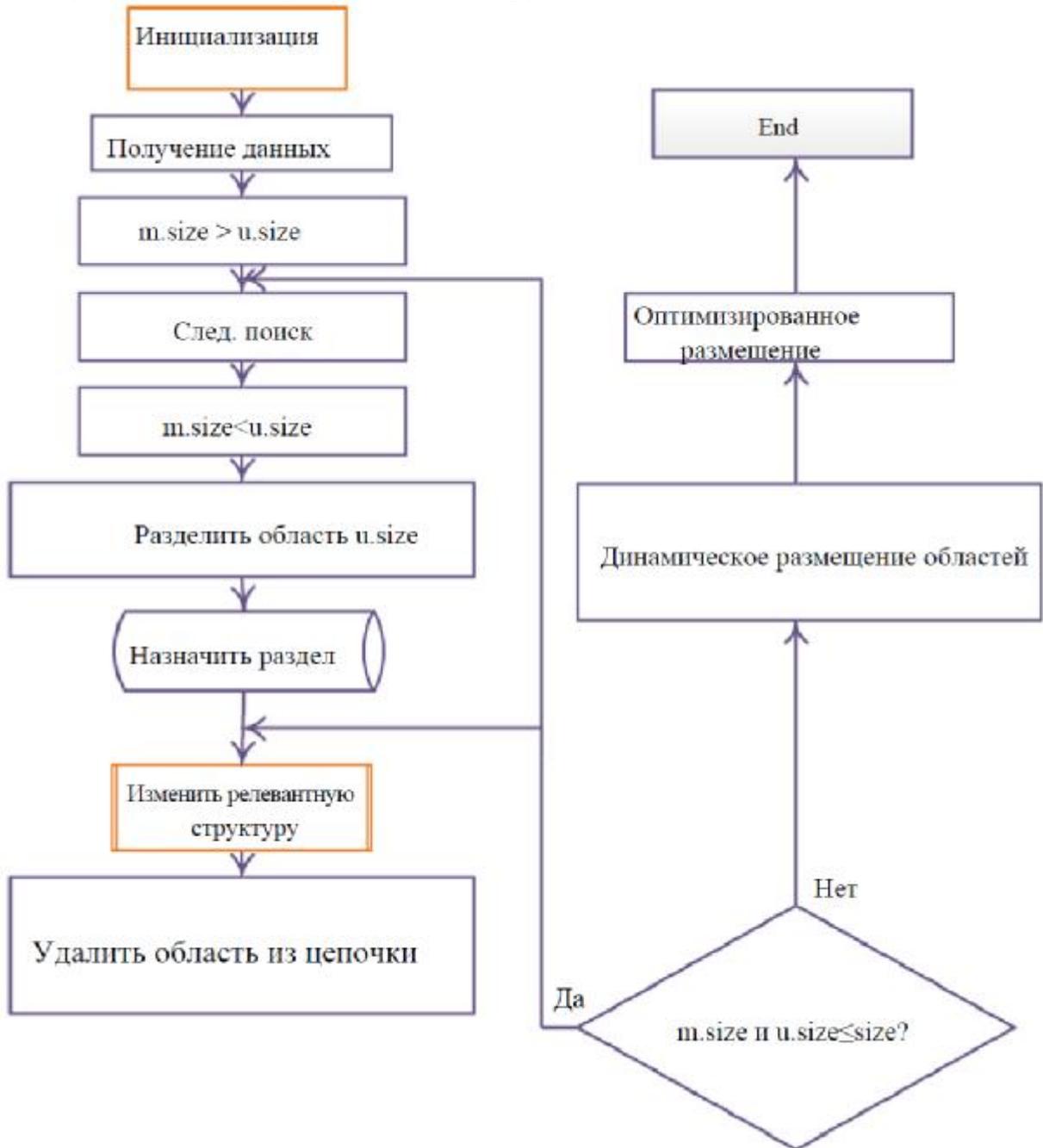


Рис. 3.3. Структурная схема архитектуры динамической системы распределения данных

Таким образом, предложена архитектура динамической системы распределения данных, отличающаяся использованием «жадного» алгоритма сокращения миграции данных с динамическим выбором точки

данных в перегруженном узле хранения с максимальной нагрузкой и обеспечивающий регулирование распределения данных по каждому узлу хранения в режиме реального времени.

В главе 4 проанализированы особенности интеграции больших данных в системы принятия решений.

Основной вопрос, который рассматривается, заключается в следующем: какие аспекты следует учитывать при разработке проекта с использованием больших данных, направленного на решение проблемы принятия решений в организации?

Модель BD-Da использует на три уровня, которые необходимо учитывать при разработке проекта с использованием больших данных, направленного на решение проблемы принятия решений в организации. Эти уровни - уровень данных, анализ данных, принятие решений (рис. 4.1).

- **Уровень данных:** основное внимание уделяется определению наборов данных, которые будут использоваться, то есть функций, характеризующих большие данные, а именно 4V, а также различных внутренних и внешних источников, которые предоставляют эти данные.

- **Уровень анализа данных:** Большие данные скрывают важную ценную информацию, которая позволяет лицам, принимающим решения, принимать более быстрые и разумные решения. Этот уровень направлен на достижение этого понимания путем сбора, хранения, обработки и анализа огромных объемов данных посредством применения мощных аналитических методов и использования новых инструментов, способных обрабатывать характеристики больших данных. Получив желаемую информацию, ее необходимо визуализировать и представить лицам, принимающим решения, в структурированном и понятном формате для использования на уровне принятия решений.

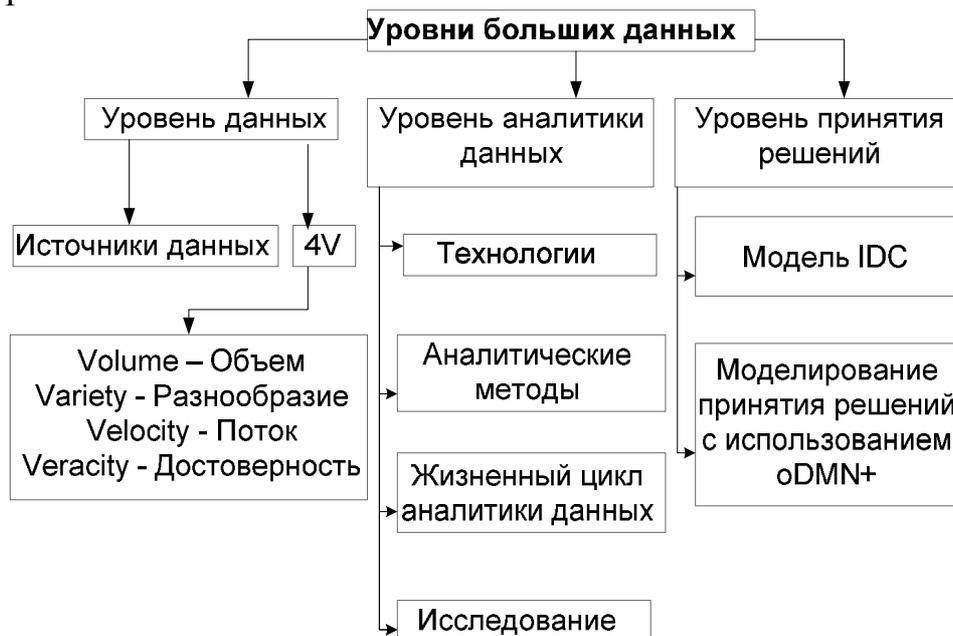


Рис. 4.1. Уровни модели BD-Da

- **Уровень решения:** он действует как аспект принятия решений в области больших данных, который был исключен из проектов по работе с большими данными. Основное внимание уделяется увязыванию ценности больших данных с их фактическим использованием, что способствует принятию решений. Уровень основан на двух элементах: модели IDC, включающей этапы анализа, проектирования и выбора; и моделирование решений с использованием расширенной модели стандарта oDMN+.

На рис. 4.2 представлена модель BD-Da. Модель BD-Da представляет концепцию больших данных, основанную на шести концепциях, а именно: наборы данных с новыми характеристиками, жизненный цикл анализа данных, технологии, аналитические методы, понимание и принятие решений.

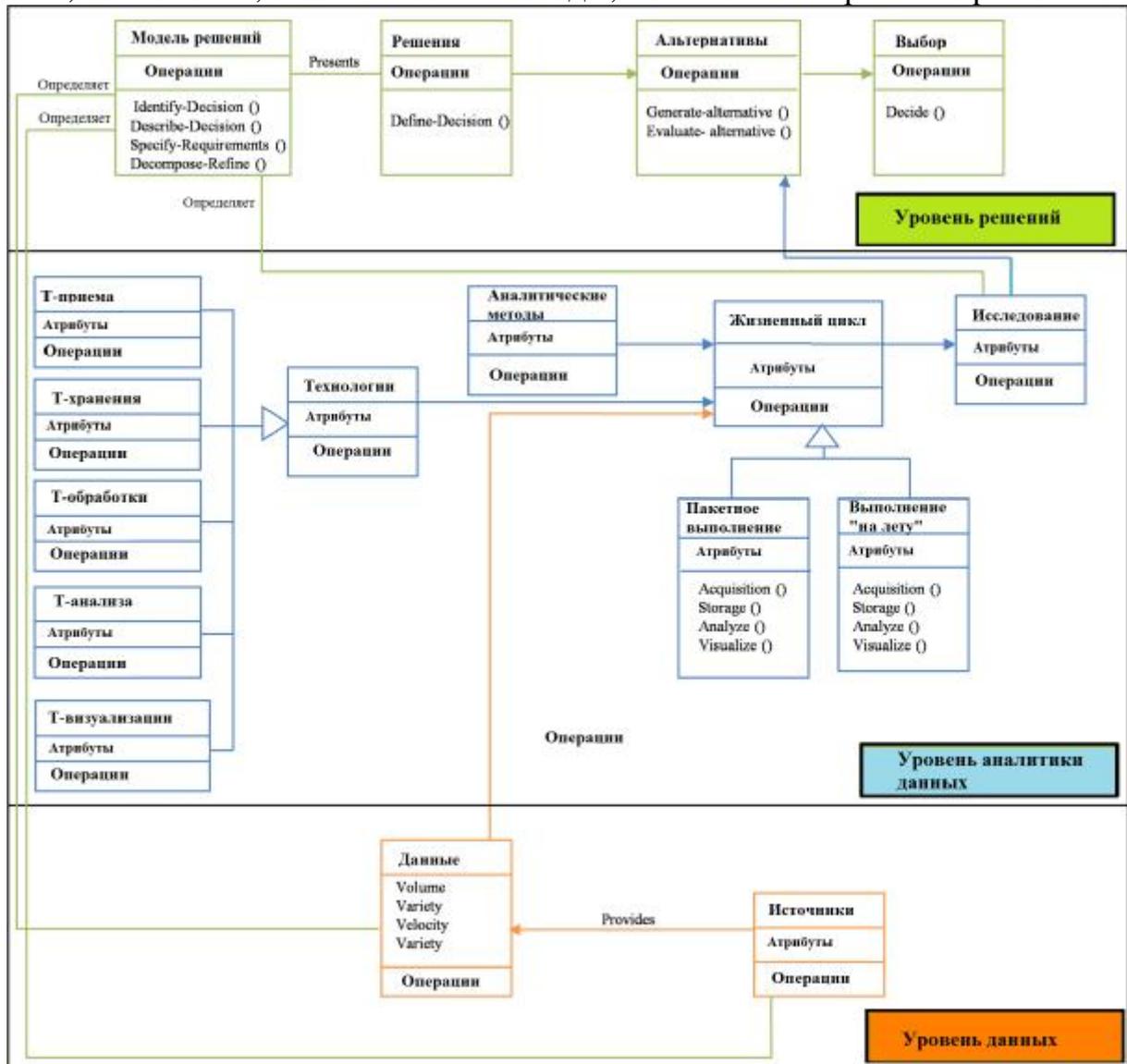


Рис. 4.2. Детализация модели BD-Da: Т-* - технологии *

Расширенная модель BD-Da учитывает источники данных, что позволяет определять и представлять потенциальные альтернативы источников, предоставляющих данные.

Таким образом, представлена графическая модель BD-Da интеграции принятия решений в большие данные, отличающаяся использова-

нием наборов данных с новыми характеристиками, жизненного цикла анализа данных, технологий, аналитических методов, понимания и принятия решений и обеспечивающая выделение трех уровней больших данных, которые необходимо учитывать при разработке проекта больших данных: данных, анализа и принятия решений.

Пятая глава описывает программные проекты управления большими данными.

Для повышения стабильности операций интеллектуального анализа гетерогенных больших данных в среде сложных атрибутов, таких как анализ и очистка данных, разработан алгоритм интеллектуальной кластеризации гетерогенных больших данных. Метод очистки данных применяется для очистки пространства параметров в среде сложных атрибутов, и вводится обычный термин кластеризации в разреженном подпространстве для устранения нерелевантной и избыточной информации из разнородных больших данных, и получается интеллектуальный индекс кластеризации разнородных больших данных. После измерения результатов кластеризации завершается разработка алгоритма интеллектуальной кластеризации гетерогенных больших данных в среде сложных атрибутов. Результаты экспериментов показывают, что алгоритм интеллектуальной кластеризации гетерогенных больших данных в среде сложных атрибутов обладает высокой стабильностью в процессе анализа и очистки данных.

На рис. 5.1 показана структурная схема алгоритма уменьшения размерности высокоразмерных данных. Алгоритм интеллектуальной кластеризации приведен на рис. 5.2.

Для проверки эффективности были проведены имитационные эксперименты в среде Matlab 7.0, VS2010 + opencv2.4.13, Windows 10, Intel Xeon CPU e5-2603v4 с частотой 2,20 ГГц и 32 ГБ оперативной памяти.

На рис. 5.3 мы видим, что количество кортежей увеличивается по мере увеличения количества вводимых записей. Доля аналогичных дубликатов и несоответствий в наборе данных по-прежнему очень высока, некоторые из них достигли более 25%, самый низкий показатель примерно 12%. Показано, что разработанный алгоритм является относительно стабильным.

Разработанная архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей представлена на рис. 5.4. Укрупненно архитектура содержит три больших блока – итерационное распределение больших данных, обработка данных в процессе итерации и оптимизация.

Таким образом, представлена архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей, отличающаяся итерационным распределением больших данных на основе упорядочивания объектов и ссылок и реализующая уменьшение доли дубликатов и несоответствий в наборе данных в среднем на 12%.



Рис. 5.1. Структурная схема уменьшения размерности высокоразмерных данных

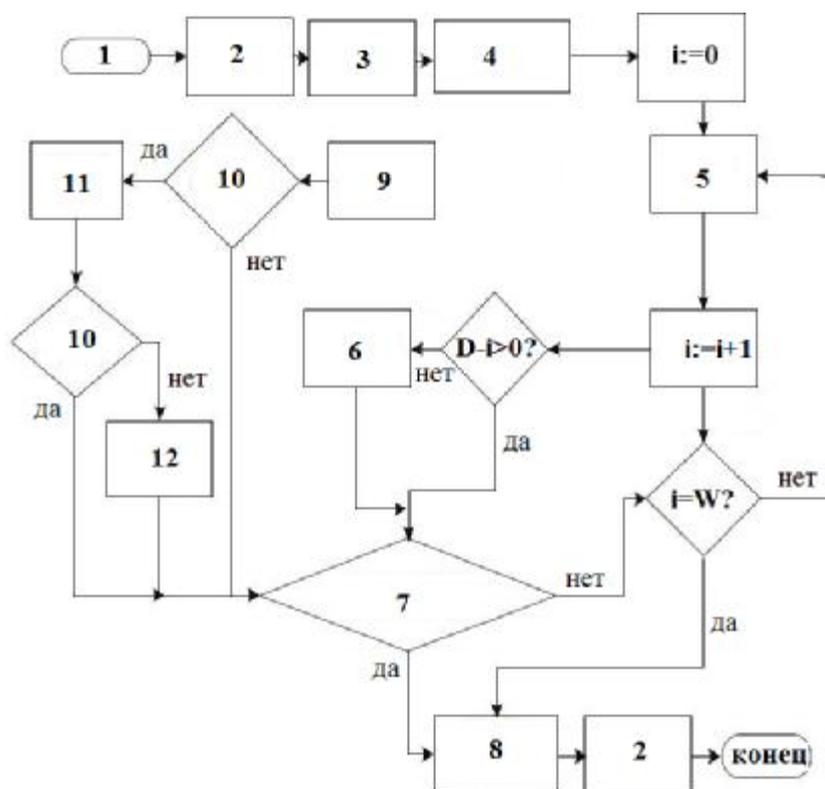
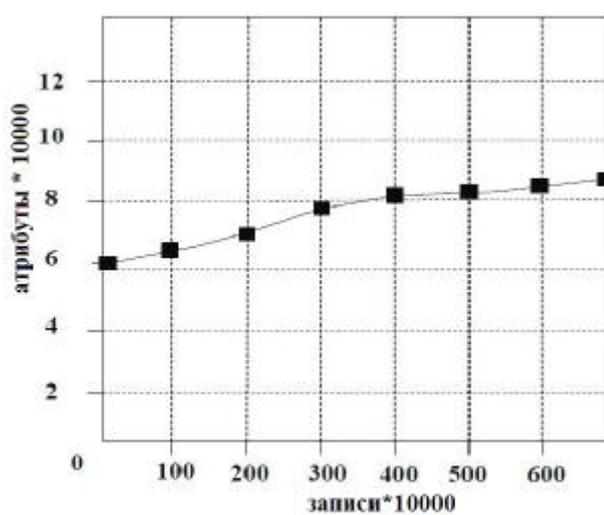
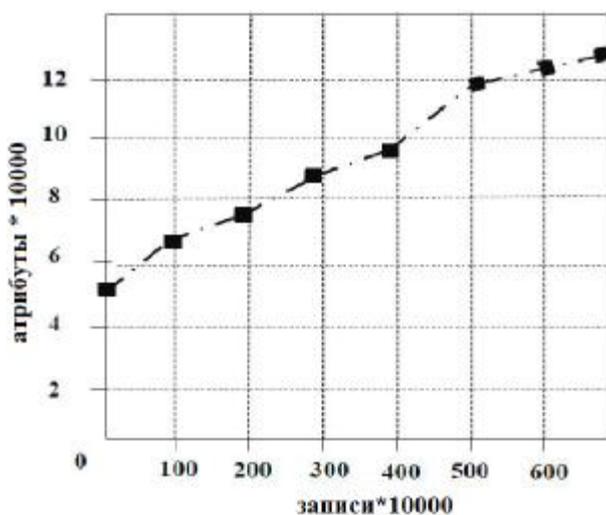


Рис. 5.2. Алгоритм интеллектуальной кластеризации разнородных больших данных: 1 – старт; 2 – N записей; 3 – сортировка; 4 - установка размера окна W равным M; 5 - перемещение окна вниз по очереди и запись его в переменную D; размещение D записей; 7 – сравнение записанных n-й и n-W-й записей; 8 – очистка дубликатов; 9 – быстрая сортировка; 10 – порог превзойден?; 11 - эффективное соответствие; 12 - улучшенные правила транзитивности



а) Экспериментальные результаты применения традиционных алгоритмов

б) Экспериментальные результаты работы предложенного алгоритма

Рис. 5.3. Экспериментальные результаты

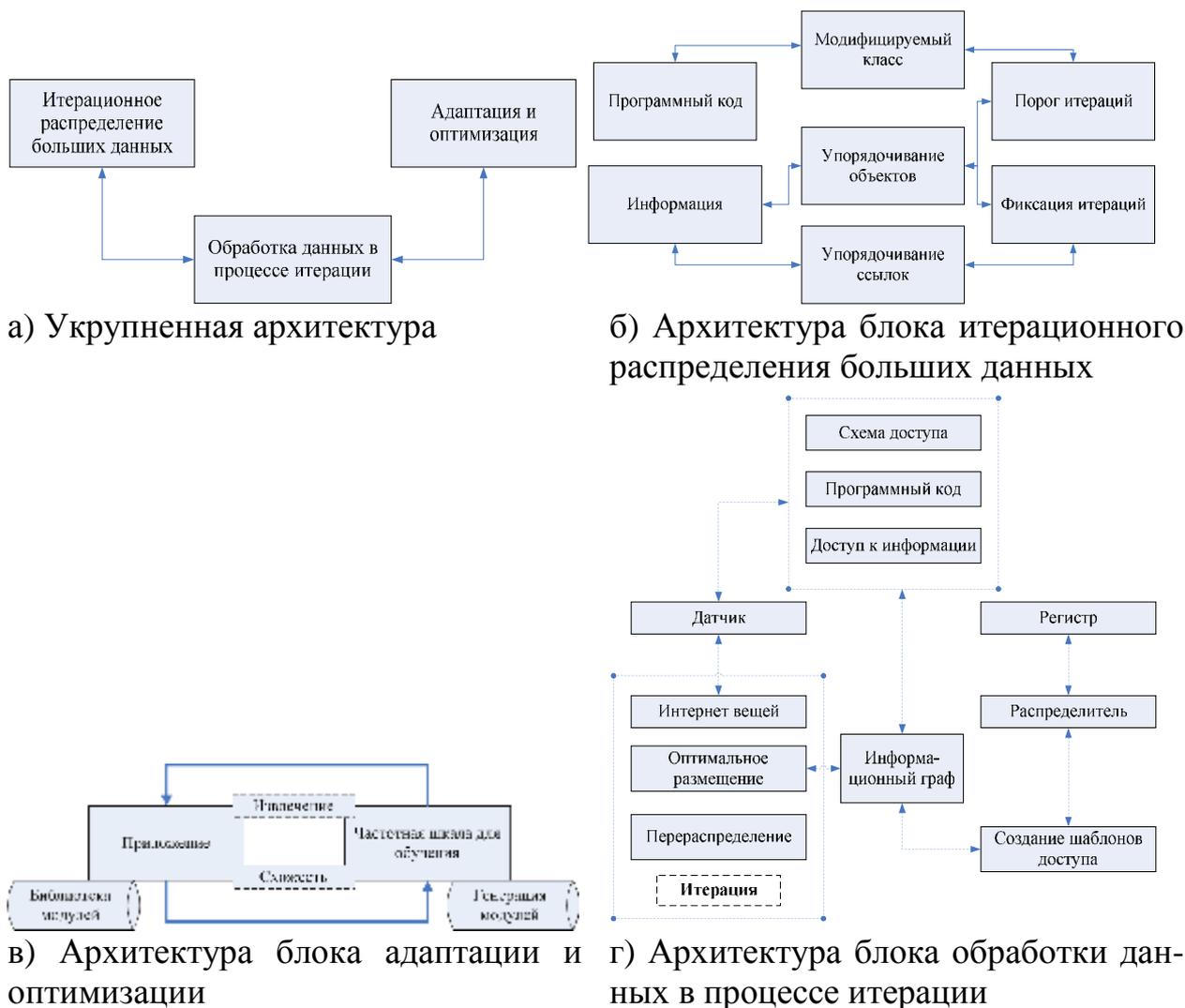


Рис. 5.4. Архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей

Заключение

В процессе выполнения диссертационного исследования получены следующие основные результаты:

1. Проведен анализ проблем управления большими данными облачных сервисов на основе многостадийных алгоритмов и динамического перераспределения данных.

2. Разработан алгоритм расширения хранилища больших сервисов в различных облачных зонах, обеспечивающий оценку близости формальных концепций, которые объединяют эти сервисы и источники данных.

3. Создан алгоритм компоновки больших сервисов, обеспечивающий отбор кандидатов, их комбинацию и оптимальных выбор больших сервисов, отвечающий требованиям QoS, качества данных и безопасности и улучшающий качество итогового большого сервиса в среднем на 3.4%.

4. Предложена архитектура динамической системы распределения данных, обеспечивающий регулирование распределения данных по каждому узлу хранения в режиме реального времени.

5. Разработана графическая модель интеграции принятия решений в

большие данные, обеспечивающую выделение трех уровней больших данных, которые необходимо учитывать при разработке их проекта: данных, анализа и принятия решений.

6. Разработана архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей, реализующую уменьшение доли дубликатов и несоответствий в данных в среднем на 12%.

7. Элементы программного обеспечения зарегистрированы в ФИПС.

Рекомендации и перспективы дальнейшей разработки темы

1. Результаты исследования рекомендуются к применению в задачах управления большими данными облачных сервисов на основе многостадийных алгоритмов и динамического перераспределения данных.

2. Дальнейшая разработка темы будет направлена на практическую реализацию теоретических и алгоритмических результатов, интеграцию в наиболее распространенные распределенные системы. Развитие результатов будет направлено на улучшение модифицируемости и реконфигурируемости программных систем.

Основные результаты диссертации опубликованы в следующих работах:

Публикации в изданиях списка ВАК

1. Сотников Д.В., Атласов И.В., Божко Л.М. Использование NoSQL моделей для хранилищ данных специального назначения// Системы управления и информационные технологии, №1(91), 2023. – С. 68-73.

2. Сотников Д.В., Кравец О.Я. Атласов И.В. Улучшение качества обслуживания на основе тензорной модели больших данных// Системы управления и информационные технологии, №4(94), 2023. – С 73-81.

3. Сотников Д.В., Атласов Д.И., Кравец О.Я., Красновский Е.Е. Исследование метода оптимизации данных для эксплуатации и сопровождения базы знаний программного обеспечения на основе облачных вычислений// Системы управления и информационные технологии, №2(100), 2025. С. 37-42

4. Атласов Д.И., Сотников Д.В., Кравец О.Я., Красновский Е.Е. Оценка неопределенности нулевых значений базы данных на основе искусственного интеллекта// Системы управления и информационные технологии, №2.1(100), 2025. С. 4-11

5. Аль-Имари М., Гетманская Д.В., Кравец О.Я., Сотников Д.В. Теоретические основы мониторинга изменений больших данных в крупномасштабных разреженных невзвешенных сетях с облачной обработкой// Моделирование, оптимизация и информационные технологии. 2025;13(3). URL: <https://moitvvt.ru/ru/journal/pdf?id=2004> DOI: 10.26102/2310-6018/2025.50.3.026.

Публикация в издании, входящем в базу данных WoS

6. Mutin D.I., Kapferko A.F., Sorokin S.A., Sotnikov D.V., Atlasov I.V., Ryndin N.A. Automation of adaptive control of complex objects states trajectories in artificial intelligence systems// International Journal on Information Technologies and Security, vol.16, no.1, 2024, pp. 57-64. <https://doi.org/10.59035/ZDGM9286>.

Свидетельство о государственной регистрации программы для ЭВМ

7. Атласов Д.И., Сотников Д.В., Васми Ихаб А Васми, Хуссейн Али Иед, Линкина А.В. Типовой интерфейс облачных вычислений. Свидетельство о регистрации программы для ЭВМ № 2025681822 от 18.08.2025. - М.: Роспатент, 2025.

Статьи и материалы конференций

8. Сотников Д.В., Кравец О.Я. Программные инструменты работы с облачными сервисами и большими данными// Информационные технологии моделирования и управления, №2(132), 2023. – С. 152-157.

9. Sotnikov D.V., Kravets O.Ja. A multi-module system for big data analysis based on machine learning// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2023'AS): Proceedings of the XXVIII-th International Open Science Conference (Yelm, WA, USA, January 2023). - Yelm, WA, USA: Science Book Publishing House, 2023. – Pp. 198-205.

10. Сотников Д.В. Сравнение MongoDB и MySQL при различных нагрузках// Решение: матер. XII Всеросс. научно-практической конференции. – Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2023, -С. 169-172.

11. Сотников Д.В. Изучение причин и преимуществ использования баз данных NoSQL// Сборник научных статей по материалам II Всеросс. научной конференции «Достижения науки и технологий-ДНиТ-II-2023». Выпуск 7. - Красноярск, 2023, С. 404-411

12. Сотников Д.В., Кравец О.Я. Применение тензорной модели больших данных для улучшения качества обслуживания// Оптимизация и моделирование в автоматизированных системах: тр. Междунар. молодежной научной школы. – Воронеж: ВГТУ, 2023. С. 62-66.

13. Sotnikov D.V., Kravets O.Ja., Potudinskii A.V. Analyzing and creating a model of big poorly structured data// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2024'SCT): Proceedings of the XXIX-th International Open Science Conference. - Yelm, WA, USA: Science Book Publishing House, 2024. - Pp.72-80.

14. Сотников Д.В., Кравец О.Я. Применение специализированных моделей NoSQL для хранилищ данных// Интеллектуальные информационные системы: тр. Междунар. НПК, посв. 40-летию кафедры САПРИС. - Воронеж, 2024. – с. 152-156.

15. Сотников Д.В. Концептуальные основы и состояние проблемы моделирования интеграции больших данных в системах принятия решений// Сб. тр. VI Всеросс. НПК «Информационные технологии в экономике и управлении». – Махачкала, 2024. С. 90-95.

16. Sotnikov D.V., Kravets O.Ja. An approach to modeling the integration of big data into decision-making systems// Modern informatization problems in the technological and telecommunication systems analysis and synthesis (MIP-2025'AS): Proc. of the XXX-th Int. Open Science Conf. - Yelm, WA, USA: Science Book Publishing House, 2025. – pp. 251-264.

17. Сотников Д.В., Кравец О.Я. Уровневая модель больших данных для

имплементации в систему принятия решения корпоративного программного обеспечения// Экономика и менеджмент систем управления, №1(55), 2025. – С. 80-90.

18. Сотников Д.В. Концепция больших данных как основа процесса принятия решений// Информатика. Экономика. Управление - Informatics. Economics. Management, 2025, 4(1), 2038–2042. <https://doi.org/10.47813/2782-5280-2025-4-1-2038-2042>

19. Sotnikov D.V. An intelligent algorithm for clustering big data in an environment with complex attributes// Modern informatization problems in simulation and social technologies (MIP-2026'SCT): Proceedings of the XXXI-th International Open Science Conference (Yelm, WA, USA, January 2026). - Yelm, WA, USA: Science Book Publishing House, 2026. – Pp. 117-128.

Подписано в печать 13.03.2026.

Формат 60x84/16. Бумага для множительных аппаратов.

Усл. печ. л. 1,0. Тираж 80 экз. Заказ №169.

ФГБОУ ВО «Воронежский государственный технический университет»
394026 Воронеж, Московский просп., 14