ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

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

Диссертация
на соискание учёной степени кандидата технических наук

Научный руководитель:
д.т.н., профессор Кравец Олег Яковлевич

Воронеж – 2026

Оглавление

Введение

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 управления большими данными облачных сервисов, интерес представляет и задача компоновки больших облачных сервисов. 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. Тема 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 данных, 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 данные, 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 принятых решений 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, отве-

чающий 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 натурных экспериментов.

Публикации. По результатам диссертационного исследования опубликовано 19 научных работ, в том числе 7 – в изданиях, рекомендованных ВАК РФ (из них 1 – в издании, индексируемых 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 хранилища 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. Сформулирована цель и задачи исследования.

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.

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.

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.

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.

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

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.

Модель BD-Da представляет концепцию больших данных, основанную на шести концепциях, а именно: наборы данных с новыми характеристиками, жизненный цикл анализа данных, технологии, аналитические ме-

тоды, понимание и принятие решений.

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

Представлена графическая модель BD-Da интеграции принятия решений в большие данные, 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 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.

Представлена архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей, отличающаяся итераци-

онным распределением больших данных на основе упорядочивания объектов и ссылок и реализующая уменьшение доли дубликатов и несоответствий в наборе данных в среднем на 12%.

В заключении представлены основные результаты работы, а также рекомендации и перспективы дальнейшей разработки темы.

1. Проблемы и особенности использовании фреймворков больших данных для создания больших сервисов

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”. В этой новой модели сервисы можно рассматривать как абстрактный уровень, который скрывает сложность обрабатываемых больших данных. Для удовлетворения сложных и разнородных потребностей пользователей в эпоху больших данных повторное использование сервисов является естественным и эффективным средством, которое помогает организовать работу доступных сервисов для предоставления больших сервисов по требованию клиентов. Несмотря на отличие от традиционной структуры веб-сервисов, создание больших сервисов подразумевает повторное использование не только существующих высококачественных сервисов, но и высококачественных источников данных с учетом их ограничений безопасности (например, происхождения данных, уровня угрозы и утечки данных). Кроме того, создание разнородных и крупномасштабных сервисов, ориентированных на обработку данных, сталкивается с рядом проблем, помимо угроз безопасности, таких как высокое время выполнения крупных сервисов и несовместимость политик поставщиков в нескольких доменах и облаках. Стремясь решить вышеуказанные проблемы, мы предлагаем масштабируемый подход к составлению большого количества сервисов, который учитывает не только качество повторно используемых сервисов (QoS), но и качество используемых источников данных (QoD). Поскольку правильное представление требований к крупным сервисам является первым шагом на пути к эффективной структуре, мы сначала предлагаем модель качества для крупных сервисов и количественно оцениваем утечки дан-

ных, используя показатели L-severity. Затем, чтобы облегчить обработку и извлечение информации, связанной с большими сервисами, в процессе компоновки, мы используем мощную математическую основу анализа нечетких реляционных концепций (нечеткий RCA) для создания хранилища больших сервисов в виде семейства решеток. Мы также использовали нечеткий RCA для кластеризации сервисов и источников данных на основе различных критериев, включая их уровни качества, домены и взаимосвязи между ними. Наконец, мы определяем алгоритмы, которые анализируют семейство решеток для параллельного выбора и создания высококачественных и безопасных крупных сервисов. Предложенный метод, который реализован поверх платформы Spark big data framework, сравнивается с двумя существующими подходами, и экспериментальные исследования доказали эффективность нашего подхода к составлению большого сервиса с точки зрения состава с учетом QoD, масштабируемости и устранения нарушений безопасности.

1.1. Большие данные и облачные вычисления

В эпоху больших данных облачные вычисления стали естественным выбором для предоставления массивных и разнообразных источников данных в качестве сервисов. Эта парадигма играет важную роль, поскольку обеспечивает программную и аппаратную поддержку для размещения, управления и хранения огромных объемов данных. Она также предоставляет данные в виде сервисов для удовлетворения потребностей пользователей, а также сервисов для взаимодействия с данными, позволяющих их обрабатывать и управлять ими. Эта стратегия дает множество преимуществ, таких как масштабируемость, гибкость, емкость хранилища и простота управления данными.

Кроме того, облачные вычисления обеспечивают базовый движок за счет использования платформ обработки больших данных, таких как

Hadoop, Spark, Storm, которые являются платформами распределенной обработки данных [2.19].

Синергия между ориентацией на обслуживание, облачными вычислениями и обработкой больших данных привела к появлению новой модели обслуживания, получившей название большие сервисы (BS) [2.43]. Большие сервисы рассматриваются в [2.33] как веб-сервисы, требующие больших объемов данных, и определяются в [2.43] как совокупность нескольких междоменных (виртуальных и физических) сервисов, которые получают доступ к большому объему данных и обрабатывают его. Сервисы обработки больших данных и приложения, ориентированные на большие данные, также могут быть результатом синергии между другими смежными областями, такими как мобильные вычисления [2.12], периферийные вычисления [2.13] и беспроводные сети [2.14]. В качестве примеров можно привести мобильные приложения с поддержкой киберпространства [2.13] и интеллектуальные производственные сервисы [2.14]. Первые имеют дело с различными требованиями к сбору данных и распределению задач, в то время как вторые используют преимущества растущего числа подключенных устройств Интернета вещей и их интенсивно обрабатываемых данных.

Чтобы удовлетворить сложные и масштабные требования пользователей, различные типы сервисов (например, веб, облачные, мобильные и т.д.) и источники данных (например, социальные сети, датчики и т.д.) из разных доменов и облачных зон должны быть объединены и предлагаться по запросу. Подобно веб- и облачным сервисам, составные крупные сервисы должны соответствовать функциональным требованиям пользователей, их качеству обслуживания и контекстуальности. Однако, учитывая огромный объем источников данных, потребляемых крупными сервисами, необходимо учитывать дополнительные ограничения, такие как происхождение данных, совместимость политик поставщиков, качество используемых ис-

точников данных (QoD), модель сбора данных крупными сервисами, корреляция между источниками данных и т.д. Действительно, источники данных, используемые крупными сервисами, предлагаются несколькими поставщиками, которые придерживаются различных политик безопасности и конфиденциальности. Следовательно, выбор неподходящих источников данных может привести к ненадежному поведению составленного большого сервиса и, следовательно, вызовет некоторые риски для безопасности, такие как утечки данных и угрозы или утечка данных [2.2]. Кроме того, в отличие от традиционной структуры веб-сервисов и облачных сервисов, высококачественная структура больших сервисов зависит не только от качества агрегированных доменных сервисов, но и от качества используемых источников данных. Кроме того, важно свести к минимуму количество источников данных, чтобы уменьшить неоднородность политики поставщиков и раскрытие конфиденциальной информации, что, следовательно, повысит доверие к единому крупному сервису.

Было предложено немного подходов для решения проблемы большого состава служб, поскольку эта тема появилась недавно. Некоторые из них были сосредоточены только на удовлетворении требований к качеству обслуживания [2.18, 2.21, 2.22], в то время как другие пытались снизить сложность создания больших сервисов, применяя модели параллельного программирования (например, Map Reduce) [2.17, 2.21]. Несколько других подходов находятся на более ранней стадии разработки, поскольку они разрабатывали только абстрактные архитектуры, зависящие от предметной области, для предоставления больших сервисов Интернета вещей [2.6, 2.34]. Поскольку большие сервисы рассматриваются как большие и сложные предметно-ориентированные рабочие процессы, некоторые исследования были сосредоточены на сокращении времени их выполнения и максимизации их надежности с использованием различных методов (например, смешанного целочисленного программирования, линейной регрессии,

анализа иерархий) [2.18, 2.26]. Существующие исследования не смогли достичь цели создания BS из-за недостаточного понимания возможностей больших сервисов. Действительно, ни в одном из них не была предложена подробная модель описания, которая правильно определяет поведение и требования BS. Кроме того, существующие подходы рассматривают состав большого сервиса только на уровне сервиса и полностью игнорируют уровень источников данных, что делает их решение далеким от ориентации на большие данные. Кроме того, несмотря на сложный процесс создания BS и масштабный характер среды BS, большинство существующих подходов при внедрении своих решений игнорируют рамки больших данных.

Стремясь решить вышеперечисленные проблемы, мы предлагаем комплексный подход к предоставлению больших сервисов, который использует преимущества мощного математического метода, а именно анализа нечетких реляционных концепций (нечеткий RCA) [2.8]. Этот метод кластеризации будет использоваться для обеспечения явного представления компонентов крупных сервисов вместе с используемыми ими источниками данных. Мы также используем нечеткий RCA для группировки сервисов-кандидатов в соответствии не только с их уровнями QoS, но и с их источниками данных и уровнями безопасности, которые мы будем количественно оценивать с помощью показателя L-severity [2.39]. Предлагаемый подход реализован с использованием хорошо известной платформы обработки больших данных Apache Spark.

Ниже мы суммируем наши основные вклады, касающиеся текущих проблем с составом больших сервисов:

• Модель обеспечения качества BS: В отличие от существующих подходов, которые игнорируют этап описания BS и не дают полного представления об их возможностях, мы начинаем с расширения традиционной модели QoS за счет атрибутов качества, связанных с данными, таких как своевременность, полнота, согласованность и т.д. Это помогает понять ха-

рактеристики BS не только с точки зрения функционального поведения, но и с точки зрения качества потребляемых данных (QoD), что облегчает широкий спектр сервисов.

• Количественная оценка утечки данных: Когда речь идет о составе сервисов, ориентированных на большие данные, некоторые существующие подходы оценивают BS на уровне обслуживания (функциональное поведение и QoS) и игнорируют уровни качества и безопасности используемых источников данных. Чтобы решить эту проблему, мы определяем уровень безопасности используемых BS данных с помощью показателя L-severity [2.39], который оценивает серьезность утечек данных на основе объема и чувствительности источников данных BS.

• Построение репозитория BS: Существующие подходы используют традиционную структуру, учитывающую качество обслуживания, без учета других важных ограничений структуры, таких как отношения между сервисом и источником данных, междоменные отношения, экологические ограничения и ограничения совместимости и т.д. Чтобы облегчить задачу поиска сервисов-кандидатов и соответствующих им источников данных из разных доменов, мы используем нечеткое расширение реляционного концептуального анализа [2.8] для моделирования среды BS в виде семейства решеток. Это более позднее решение, дополненное вышеуказанными ограничениями, является результатом фаззификации хостинговой инфраструктуры BS и представляет собой распределенное хранилище BS.

• Метод определения структуры большого сервиса: Мы определяем набор алгоритмов, которые анализируют семейство решеток BS и определяют уровни QoS и утечки данных, основываясь на чувствительности данных, потребляемых каждой службой, участвующей в структуре большого сервиса. Также определена функция оценки, которая принимает информацию как о QoS, так и о качестве обслуживания, чтобы оценить составленный большой сервис.

• Внедрение и эксперименты: Мы внедряем подход к составлению большого сервиса поверх Apache Spark, одной из основных платформ обработки больших данных. Набор данных SW-DREAM [2.45] используется для проведения обширных экспериментов по оценке производительности и качества больших сервисных систем в сравнении с двумя недавними подходами.

1.2. Связанные с проблемой состава сервиса работы

Следует отметить, что проблема компоновки веб-сервисов (BSCo) - это недавняя тема, которой занимались немногие исследователи. Помимо традиционных характеристик и проблем, унаследованных от выбора и компоновки веб-сервисов, BSCo сталкивается с дополнительными проблемами. Последние в основном связаны с объемом и скоростью передачи данных, а также с неоднородностью сервисов, доменов и хостинговой инфраструктуры. Создание BS подразумевает повторное использование не только доступных сервисов, но и используемых ими источников данных. В этом контексте было предложено очень мало работ. По этой причине мы также представляем некоторые из существующих подходов к созданию служб передачи данных, которые также рассматриваются как предшественники BS.

Чтобы лучше понять модель BS, мы начнем со сравнения между BS и ее предшественниками - сервисными моделями, в основном веб-сервисами и сервисами передачи данных (см. подраздел 1.2.1). Также представлены примеры сервисов больших данных от ведущих компаний Alibaba, Oracle и IBM. Затем подходы BSCo рассматриваются и обсуждаются в подразделах 1.2.2 и 1.2.3.

1.2.1. Большие сервисы

В качестве новой модели абстрагирования больших данных и сокры-

тия их сложности BS рассматриваются как управляемая интеграция массивной, сложной серии разнородных сервисов, ориентированных на большие данные [2.43]. Такая сложная и масштабная сервисная экосистема способна обрабатывать огромные объемы данных и предлагать их в качестве сервисов по требованию клиентов.

Как и характеристики big data 5V, характеристики BS включают в себя неоднородность, массовость, клиентоориентированность, сложность, достоверность, конвергенцию и ценность [2.43]. Эти характеристики отличают BS от существующих моделей обслуживания, таких как веб-сервисы и сервисы обработки данных. Хотя BS имеют некоторые общие нефункциональные свойства (например, время отклика, доступность, стоимость, надежность и т.д.) с веб-службами и службами передачи данных, их модель описания включает дополнительные свойства QoS, зависящие от модели и домена. В отличие от веб-сервисов, которые рассматриваются как разновидность интернет-приложений, BS представляют собой набор разнородных виртуализированных и физических сервисов (веб-сервисы, облачные сервисы, IoT-сервисы и т.д.), ориентированных на большие данные. Хотя службы передачи данных также сосредоточены на данных, однако они не способны обрабатывать огромные объемы данных и, в отличие от BS, не подходят для киберфизических сценариев. Последние полезны при решении бизнес-задач за счет объединения разнородных цепочек сервисов с добавленной стоимостью из разных типов и областей (администрирование, транспорт, здравоохранение и т.д.). Другое отличие заключается в том, что репутация BS часто зависит от качества используемых ими данных. Известно, что традиционные веб-сервисы не ориентированы на данные. Следовательно, такие параметры качества данных, как полнота, своевременность, точность, согласованность и т.д., не включены в их модель качества.

Примерами BS являются интеллектуальные городские транспортные

экосистемы, которые предлагают различные сервисы, такие как дорожные карты, управление дорожным движением и отчеты, отслеживание транспортных средств и т.д. [2.43]. Данные этих служб собираются из различных источников в дорожной сети (например, источников освещения, камер и т.д.). Улучшенная среда обитания была представлена в [2.41] в качестве других возможных контекстов BS, которые используют преимущества датчиков, облачных вычислений и прикладных программ.

В настоящее время предпринимается несколько попыток предложить реальные сервисы обработки больших данных, такие как IBM Big Data Service on Silo и Big Data on Cloud. Компания Alibaba использует в Европе сервис обработки больших данных Max-Compute. Его цель - получить доступ к передовым технологиям искусственного интеллекта и глубокого обучения Alibaba Cloud, а также алгоритмам хранения, моделирования и аналитики данных. Oracle Big Data Cloud Service - это еще одно решение, предоставляющее Hadoop как сервис корпоративного уровня с комплексной безопасностью, высокой производительностью, простотой управления и возможностью обновления.

1.2.2. Компоновка больших сервисов

Состав служб, ориентированных на данные, первоначально рассматривался исследователями в контексте служб данных. Например, авторы в [2.4] представили систему составления служб данных с учетом конфиденциальности, которая позволяет выполнять запросы к нескольким службам данных, сохраняя при этом информацию, не раскрываемую ни одной из вызываемых служб. На первом этапе для шифрования числовых значений данных использовалась комбинация операций (схема шифрования с сохранением порядка). Таким образом, исполнитель композиции может получить доступ только к анонимизированной информации. На втором этапе модель использует концепцию K-защиты, которая ограничивает утечку

информации службами, которые являются частью композиции во время выполнения.

В [2.33] предложенный подход направлен на оптимизацию задержки в сети и свойств QoS, таких как время отклика и стоимость. Авторы начали с определения сетевой модели, в которой выполняется оценка задержки в сети между сервисами больших данных в облаке. Эта модель используется для прогнозирования значений времени в оба конца (RTT) между сервисами, не являющимися соседними. Полезная информация, такая как RTT между сервисами передачи данных, затем передается в систему компоновки с учетом сетевого взаимодействия, которая основана на генетическом алгоритме. Авторы рассчитывают время RTT, применяя децентрализованную матричную факторизацию на основе обучения (LADMF). Генетический алгоритм был также применен к задаче о составлении многоцелевых сервисов, в которой составные сервисы представлены геномами.

В [2.46] авторы использовали онтологию для создания и публикации веб-сервисов, требующих больших объемов данных. Для описания содержимого сервиса данных была определена модель сервиса данных, основанная на семантических веб-языках. Модель была создана путем описания семантики контента и использования графических шаблонов RDF. Служба данных содержит описание ограничений, в котором указаны ограничения, связанные с некоторыми источниками данных. Кроме того, для динамической генерации сервисных интерфейсов WSDL/SOAP в соответствии с потребностями запрашивающей стороны был применен подход, основанный на переписывании запросов. Затем планировщик структуры сервиса определяет исполняемый план запроса, используя как описание контента, так и требования службы данных к вводу/выводу.

Одной из важных особенностей подходов к составлению данных в BS является семантическое представление пользовательских запросов. Однако таких методов, как переписывание запросов или описание контента

на основе RDF, недостаточно для описания сложных и масштабных требований пользователей. Кроме того, описанные выше подходы ограничены несколькими критериями QoS, такими как задержка в сети, время отклика и конфиденциальность. Кроме того, источники данных указаны неправильно, за исключением нескольких ограничений.

В контексте BS основное внимание исследователей было сосредоточено на сокращении больших затрат времени на вычисления и выполнение, вызванных крупномасштабным характером BS. Например, [2.21] нацелен на улучшение BSCo с поддержкой QoS с точки зрения производительности и времени выполнения. MapReduce был применен для облегчения определения сервисов-кандидатов с различными уровнями QoS. Функция Map обрабатывает входные подгруппы сервисов как пары <ключ-значение>. Функция Reduce определяет наилучшие сервисы-кандидаты в соответствии с набором атрибутов QoS. Предлагаемый модифицированный подход EA/G сочетает в себе генетический алгоритм и алгоритм оценки распределения. Исходное решение генерируется с использованием оператора skyline, который сокращает пространство поиска, чтобы избежать избыточности в сервисах и повысить скорость конвергенции. Управляемая мутация позволяет получать решения от родительских служб на основе модели вероятностного управления. Основным ограничением этой работы является то, что запрашиваемый BS рассматривается как простой абстрактный рабочий процесс с набором требований к качеству обслуживания, несмотря на очевидную разницу между веб-сервисами и BS. Добавим к этому, что авторы проигнорировали уровни источников данных, и проблема компоновки представлена в виде традиционного выбора веб-сервиса на основе MapReduce.

Для решения таких задач, как вычисления и коммуникация, авторы в [2.22] предложили фреймворк под названием Mayan, который работает с BSCo между облаками. Фреймворк Mayan находит наилучшие варианты

компоновки, принимая во внимание уровни сети и обслуживания, доступность ресурсов и требования пользователей. Mayan отдает приоритет сервисам, которые имеют максимальное количество альтернатив, чтобы обеспечить бесперебойную работу сервиса. Однако авторы рассматривают BSCo как последовательность веб-сервисов или микросервисов и рассматривают ее как проблему выбора, которая заключается в замене перегруженных стандартных блоков.

В работе [2.20] авторы использовали обобщение стратифицированных графов и объединение сервисов для обнаружения и компоновки распределенных сервисов больших данных. Чтобы преодолеть проблемы несоответствия между вводимыми пользователем данными и форматом сервиса, в качестве потенциального решения были использованы внешние подключаемые модули, позволяющие сочетать парадигму "mix and match".

Как только гибкое сопоставление выполнено, запускается объединение сервисов, позволяющее комбинировать сервисы из разных источников. Однако описание потребляемых данных в графовых представлениях сервисов отсутствует. Кроме того, операции сопоставления семантических графов являются дорогостоящими, а в подходе отсутствуют детали, касающиеся оценки составленных сервисов.

В контексте мобильных сред в [2.17] был предложен двухэтапный подход BSCo, который сочетает оптимизацию роя частиц (PSO) и кластеризацию K-средних с использованием параллельного кластера с MapReduce. На первом этапе осуществляется выбор сервисов с использованием алгоритма PCPSO (Parallel Clustered PSO), где выходные данные каждого кластера представляют собой оптимальный сервис для данной задачи. На втором этапе лучшие сервисы объединяются с помощью алгоритма PSO. Сервисы выбираются и компонуются в соответствии с их уровнями качества обслуживания. Кроме того, оценка пригодности мобильных сервисов рассчитывается на основе различных параметров QoS. Что касается приня-

того параллельного метода, то каждая задача MapReduce обрабатывается на другой машине путем сортировки частиц и применения итераций с использованием k-средних. Основным недостатком этого подхода является то, что авторы сосредоточили свои усилия на оптимизации композиции и не потратили много времени на описание требований и особенностей композиции, связанных с большими данными.

В [2.34] авторы применили графическое решение для решения BSCo в средах Интернета вещей. Для этого определены сервис-ориентированная модель проектирования и фреймворк для обработки сервисов Интернета вещей, ориентированных на данные. Предлагаемый подход заключается в организации этих сервисов в древовидную структуру. Эта древовидная структура имеет многоуровневую структуру, что позволяет выполнять возможные сценарии создания различных сервисов, которые соответствуют требованиям конечных пользователей. Предлагаемая структура рассматривает три вида сервисов: сервисы доступа к ресурсам, сервисы обработки данных, относящиеся к конкретной предметной области, и динамические сервисы по запросу. Несмотря на то, что аспекты проектирования и структурное описание IoT BS являются многообещающими, в этом подходе отсутствуют подробные сведения о качестве IoT BS. Кроме того, неясно, как в предлагаемой структуре определяется взаимосвязь между источниками данных и тремя типами сервисов.

В работе [2.44] авторы обсудили новую парадигму разработки и внедрения сервисных решений, которые эффективно соответствуют огромным требованиям пользователей. Предложенный авторами фреймворк состоит из двух этапов: разработка сервиса, ориентированного на предметную область, который представляет собой подход "снизу вверх", основанный на шаблонах обслуживания, и разработка требований, ориентированных на обслуживание. Таким образом, упрощая представление и обработку требований, он позволяет предлагать эффективные сервисные решения. Основ-

ным слабым местом этого подхода является сосредоточенность только на сервисных ресурсах в качестве основного условия для создания и эксплуатации БС с добавленной стоимостью, в то время как в реальных БС также должны учитываться уровни QoS и качества данных.

В [2.6] авторы предложили подход, основанный на объединении сервисов в контексте Интернета вещей. Предлагаемый подход фокусируется на аспекте проектирования сервисов Интернета вещей и определяет платформу обработки больших данных, которая реализует сложные процессы, ориентированные на данные. Эта платформа состоит из трехуровневой архитектуры, которая основана на IoT BS и организована в виде многокорневого дерева, где каждый узел представляет IoT BS. Процесс BSCo IoT состоит из трех основных этапов:

1. вызов обнаружения сервисов для поиска BS более низкого уровня,

2. создание плана составления сервиса на основе функций агрегирования, и как только запрос будет получен,

3. состав сервиса будет выполнен, и результаты будут возвращены. Однако в предлагаемом подходе отсутствуют детали, касающиеся описания IoT BS, информационного потока между компонентами IoT-приложения, а также экспериментальной проверки.

Некоторые исследователи также рассматривают выбор BSCo как важный этап реализации. Для повышения надежности обслуживания авторы в [2.18] предложили метод выбора BS, который сочетает в себе смешанное целочисленное программирование (MIP) и коэффициент вариации. Такой подход позволяет свести к минимуму временные затраты и максимально повысить надежность обслуживания. Основное внимание уделяется временным затратам и неопределенности QoS в BS. Во-первых, неопределенность QoS вычисляется с использованием коэффициента вариации. Вычисленный

QoS преобразуется из количественных значений QoS в качественные

понятия QoS, чтобы сократить пространство поиска при выборе сервисов. Авторы также применили алгоритм MIP для извлечения списка надежных сервисов из ранее отфильтрованных сервисов. Однако авторы проигнорировали другие важные критерии QoS и рассматривают неопределенность качества BS только на уровне QoS, не принимая во внимание степень неопределенности качества данных.

В [2.26] авторы применили модель анализа, основанную на QoS, в своем подходе к выбору BS. Этот последний этап состоит из трех основных подэтапов:

1. оценка времени выполнения сервисов больших данных с помощью линейной регрессии,

2. использование анализа AHP для оценки качества обслуживания,

3. использование метода обратного отслеживания для оптимизации алгоритма выбора сервиса больших данных.

Для расчета QoS в модель QoS были включены пять критериев качества. В двух вышеупомянутых подходах [2.18, 2.26] ограничение выбора BS несколькими критериями QoS (временные затраты и надежность) не приведет к достижению цели BS. Кроме того, эти два подхода не учитывали аспекты качества данных и другие характеристики BS, такие как массовость и ценность.

1.2.3. Анализ существующих подходов

В этом подразделе мы классифицируем и анализируем рассмотренные выше подходы, основываясь на их целях и контексте составления, принятом методе, целевых параметрах и типе обрабатываемых данных (см. табл. 1.1).

Таблица 1.1

Сравнение подходов

Работа	Цель	Методы	Основной параметр	Контекст	Тип данных
2.4	Состав службы	OPES	Конфиденциальность	Облако	–
2.6	Состав службы	Теория графов	QoS	Ориентированный на данные IoT BS	IRMA сервисы Smart city
2.6	Состав службы	–	QoS	IoT BS	Данные датчиков, открытые данные
2.17	Состав службы	PSO с K-means, MapReduce	QoS	Большие мобильные сервисы	Сервисы Smart city
2.18	Выбор службы	CV & MIP	QoS, надежность	BS	Веб-сервис данных QoS
2.20	Обнаружение и состав служб	Сопоставление графов	Задержка	–	SQL-запросы
2.21	Состав службы	MapReduce, ГА с управляемой мутацией	QoS	BS	Случайно сгенерированные данные
2.22	Состав службы	MapReduce, Промежуточное программное обеспечение, ориентированное на передачу сообщений	QoS	Межсетевое взаимодействие	–
2.27	Интеграция сервисов и данных	Семантика	–	Обработка данных	–
2.33	Состав службы	ГА + LADMF	Сетевая задержка, QoS	Основанный на облаке BDS	Планетарный лабораторный меридиан
2.44	Состав службы	Двунаправленный подход	QoS	BS	Данные Smart city

Работа	Цель	Методы	Основной параметр	Контекст	Тип данных
2.46	Выбор службы	Переписывание запросов, жадный алгоритм	Время сборки	Сервисы, требующие больших объемов данных	-

Из табл. 1.1 видно, что мультиоблачные и межоблачные среды являются наиболее часто используемыми контекстами при решении проблем, связанных с BS, будь то выбор и состав служб или другие решаемые проблемы. Это понятно, поскольку BS развертываются, используются и управляются в нескольких облачных зонах.

Большинство подходов основаны на традиционных критериях QoS, в то время как в эпоху больших данных важно фильтровать и составлять BS на основе не только их уровней QoS, но и качества доступных данных. Исследователи сосредоточили свои исследования на таких критериях выбора сервиса, как продолжительность выполнения, цена, доступность, вероятность успеха, конфиденциальность и репутация. Было решено, что выбор BS будет зависеть от объема вводимых данных [2.43]. Что касается типов обрабатываемых данных, то очевидно, что во всех подходах отсутствуют наборы данных, используемые или создаваемые реальными БС. Это, безусловно, связано с новизной темы исследования.

Из табл. 1.1 мы также можем заметить, что в большинстве существующих подходов используются методы оптимизации для решения проблем выбора и состава BS. Возьмем в качестве примера проблему композиции, которая рассматривалась как NP-сложная задача, и, таким образом, была решена с использованием таких методов, как PSO и генетический алгоритм (ГА) [2.17, 2.21]. Были применены и другие методы, такие как методы, основанные на графах [2.20, 2.34].

Мы могли заметить, что некоторые исследователи начали применять

модели параллельного программирования к BSCo [2.17,2.21], поскольку BS изначально были разработаны для того, чтобы скрыть сложность управления огромным объемом данных, которые поступают из различных областей и источников. Однако подходы, основанные на MapReduce, находятся на ранней стадии разработки и рассматривают компоновку сервисов как традиционную проблему MapReduce с точки зрения разделения данных сервисов и распараллеливания задачи компоновки.

На основе приведенного выше анализа выявлены следующие недостатки:

• Из-за отсутствия подробной модели описания BS все подходы рассматривали общий аспект BS и не давали четкого определения и полного понимания возможностей BS. Действительно, исследователи характеризуют крупные сервисы только традиционными свойствами QoS. В рамках этих подходов BS рассматриваются как веб-сервисы, без учета уровня источника данных и параметров качества данных. Следовательно, правильное представление характеристик BS (например, QoS, качество данных, уровни безопасности и т.д.) должно быть первым шагом на пути к созданию высококачественных композиций. Мы решаем эту задачу, определяя модель описания BS, которая сочетает в себе показатели качества обслуживания (QoS) и качества данных (QoD).

• Существующие подходы рассматривают BSCo как традиционную проблему повторного использования сервиса с учетом QoS, в то время как в эпоху больших данных эту проблему необходимо рассматривать с точки зрения больших данных. На самом деле, ни один из существующих подходов не рассматривает используемые источники данных и их уровни качества в качестве критериев состава. Существующие решения также игнорируют взаимосвязи между сервисами и источниками данных в нескольких доменах. Это затрудняет определение ограничений по составу (например, политики совместимости между поставщиками) и может повлиять на каче-

ство конечного BSCo. Мы решаем эту проблему, применяя реляционное представление среды BS с помощью нечеткого RCA [2.31].

• Несмотря на то, что BS ориентированы на большие данные, большинство подходов игнорируют рамки больших данных при решении проблем с большими сервисами. Эти подходы основаны на немасштабируемых алгоритмах, которые не могут справиться с масштабируемостью и сложностью BS. Внедряя Apache Spark в качестве одной из основных платформ обработки больших данных, мы стремимся не только снизить сложность компоновки, но и обеспечить эффективную работу с данными за счет точного анализа и оценки доступных сервисов и источников данных.

1.3. Примеры сценариев компоновки веб-сервисов

Как упоминалось в разделе 1.1, BSCo подразумевает повторное использование различных типов виртуализированных и физических сервисов для создания экосистем, ориентированных на большие данные. Этот процесс включает в себя определение не только высококачественных сервисов, но и наиболее надежных поставщиков данных, основываясь на нескольких ограничениях, включая качество источников данных (QoD), их происхождение и информацию о безопасности. Действительно, создание и предоставление сервисов в эпоху больших данных в значительной степени зависит от качества используемых источников данных, которые различаются по полноте, последовательности, точности, своевременности и т.д.

BSCo отличается от традиционной компоновки веб-сервисов и облачных сервисов, главным образом, масштабностью проблемы компоновки, неоднородностью задействованных объектов и моделей качества. На самом деле, степень сложности BSCO очень высока по сравнению с веб-сервисами, сервисами обработки данных или даже облачными сервисами. Это понятно, поскольку согласование и выполнение BS обеспечивается в

различных доменах и нескольких разнородных облачных зонах (например, облачные федерации, микрооблачки, мобильные облака и т.д.). Более того, среда BS рассматривается как большой распределенный контейнер с огромными объемами данных, которые инкапсулируются и обрабатываются BS.

Другое отличие касается неоднородности компонентов BS. На самом деле, BS являются результатом сочетания различных типов доменных сервисов, включая веб-сервисы и сервисы передачи данных, облачные сервисы и сервисы Интернета вещей, объединенные приложения, виртуализированные ресурсы и т.д. Это отличается от традиционной компоновки веб-сервисов, которая заключается в объединении программных приложений на базе Интернета. Даже в контексте компоновки облачных сервисов разнородность составных сервисов ограничивается вертикальной компоновкой (т.е. объединением сервисов из разных слоев облака, таких как сервисы SaaS и PaaS). Неоднородность в контексте BS также связана с характером используемых источников данных, которые поступают из разных доменов и имеют разные схемы, политики контроля доступа и т.д.

Третье важное отличие касается модели качества BS, которая определяется как на уровне обслуживания, так и на уровне данных. В отличие от традиционной структуры сервиса, которая оценивает глобальное качество составного сервиса только с учетом его возможностей QoS (например, время отклика, надежность, доступность и т.д.), процесс BSCo должен учитывать как уровни QoS составляющих сервисов, так и уровни качества (QoD) используемых ими источников данных. В этом случае функция оценки BS также основана на таких характеристиках качества больших данных, как полнота, согласованность, точность, своевременность и т.д.

Чтобы лучше понять проблему BSCo, на рис. 1.1 показаны три сценария из областей здравоохранения, государственного управления и социальных сетей.

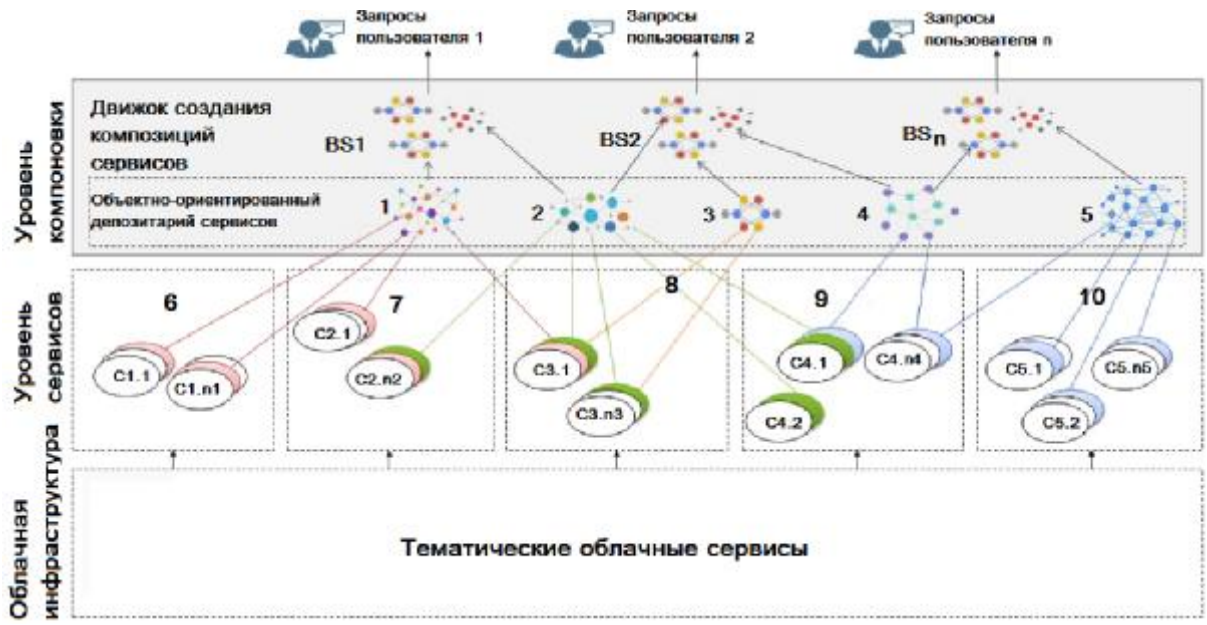


Рис. 1.1. Сценарии BS по требованию: BS_k – композиция сервисов k; 1 – потоки данных службы здоровья; 2 – потоки данных трафика; 3 - потоки данных о погоде; 4 - потоки данных системы административного управления; 5 - потоки данных медиа и социальных сетей; 6 - домен службы здоровья; 7 - домен данных трафика; 8 – домен данных о погоде; 9 – домен системы административного управления; 10 – домен медиа и социальных сетей; C_in_j – сервисы домена i группа j

Как можно видеть, элементы среды BS организованы в три уровня.

• Облачный уровень: предоставляет базовую инфраструктуру для размещения различных источников данных, относящихся к конкретной предметной области, включая большие медицинские данные (например, от страховых компаний, больниц и центров неотложной помощи), данные о дорожном движении (например, от датчиков Интернета вещей, дорожных сетей, управления дорожным движением и отчетов и т.д.), информацию о погоде (например, от метеорологических институтов и сетей датчиков), данные государственного управления (например, от социальных учреждений и правительственных департаментов), данные средств массовой информации и социальных сетей (например, от организаций средств массовой информации, социальных сетей и сетей, ориентированных на бизнес, и т.д.). Другие данные, относящиеся к конкретной предметной области, так-

же предоставляются в виде сервисов, таких как сервисы smart living environments [2.41].

• Уровень обслуживания: на этом уровне большие данные, относящиеся к конкретной предметной области, инкапсулируются в виртуализированные сервисы, такие как SaaS, PaaS, IaaS, мобильные сервисы, сервисы передачи данных и т.д. Доступ к этим данным также осуществляется с помощью специализированных сервисов с высокой степенью детализации. В качестве примеров мы приводим сервисы по оказанию первой медицинской помощи и страхованию из области здравоохранения; сервисы по определению местоположения и составлению дорожных карт из области дорожного движения; государственные, общественные и охранные сервисы из области государственного управления; социальные и бизнес-ориентированные сервисы из области СМИ и социальных сетей и т.д. Чтобы удовлетворить сложные требования пользователей, сервисы на этом уровне объединяются в рабочие процессы, зависящие от конкретной предметной области (например, рабочие процессы здравоохранения, управления трафиком, администрирования и т.д.), затем организуются и предлагаются в виде BS по запросу.

• Уровень компоновки: помимо облачного уровня и уровня сервисов, модуль компоновки отвечает за предоставление BS по требованию, составляя и организуя доступные рабочие процессы из разных доменов и в нескольких облаках. Этот уровень использует преимущества предметно-ориентированного хранилища BS, содержащего огромное количество рабочих процессов. Последние скрывают сложность и масштабность сервисного и облачного уровней, что облегчает объединение нескольких дополнительных сервисов.

На рис. 1.1 показаны три сценария BSCo. Первый сценарий BSCO - это комплексный медицинский сервис, который объединяет рабочие процессы в области здравоохранения и дорожного движения. В процессе ока-

зания медицинской помощи используются две медицинские сервисы (первая медицинская помощь и страхование), служба определения местоположения и служба прогноза погоды. Что касается управления дорожным движением, то оно сочетает в себе функции сервиса дорожной карты и двух сервисов из области погоды. Что касается второго BS, то функциональные возможности служб из трех разных областей (трафик, погода и администрирование) объединены для создания системы общественного транспорта. В этом BS операции служб прогноза погоды и информации о погоде выполняются с помощью рабочих процессов погоды и дорожного движения. Этот последний также использует существующий сервис составления дорожных карт в дополнение к данным, предоставляемым двумя сервисами из домена администрирования (правительственными службами и службами безопасности). Последний BS - это социально-ориентированная административная система, которая объединяет государственные и общественные службы (рабочий процесс администрирования) с тремя медиа-сервисами и сервисами социальных сетей для предоставления административных сервисов по требованию.

Для каждой BS существует огромное количество возможных рабочих процессов и сервисов, ориентированных на предметную область. Каждый из них использует данные, предоставляемые разными поставщиками с разным уровнем качества и разнородными политиками безопасности (например, погодная сеть для прогнозирования погоды и информационных служб). Следовательно, поиск подходящего источника данных с надежным качеством и источником происхождения данных имеет важное значение для определения наилучших доступных сервисов, включенных в окончательный состав. Кроме того, выбор служб, которые получают доступ к общим источникам данных (например, к данным управления дорожным движением для служб определения местоположения и составления дорожных карт), помогает свести к минимуму количество этих последних, что

уменьшает неоднородность политик доступа к данным и повышает доверие к составленной BS.

Основываясь на вышеуказанных различиях и требованиях, мы приходим к выводу, что существующие решения по составу сервисов не могут быть адаптированы к масштабному сервисному контексту. Следовательно, важно внедрить систему компоновки поверх фреймворков больших данных, чтобы справиться со сложной и крупномасштабной природой BS. Что касается неоднородности компонентов BS, мы решаем эту проблему путем моделирования отношений между междоменными сервисами и источниками данных с использованием нечеткого расширения анализа реляционной концепции. Наконец, оценка составленных BS обеспечивается за счет определения новой функции оценки, которая учитывает как параметры QoS, так и качества обслуживания, в отличие от традиционных подходов, учитывающих только качество обслуживания.

1.4. Постановка задач работы

В исследовании мы рассмотрели проблему повторного использования сервисов в эпоху больших данных. BS, рассматриваемые как сложные экосистемы, создаются и предоставляются путем повторного использования разнородных сервисов (веб, мобильных, облачных, данных и т.д.) из разных доменов в нескольких облачных зонах доступности. Чтобы справиться с проблемами BSCo, в основном с проблемами QoS и безопасности, мы начали с понимания свойств BS и определения модели качества для BS. Предлагаемая модель расширяет традиционную модель QoS веб-сервисов, используя характеристики, связанные с “большими данными” (атрибуты QoD).

На втором этапе мы использовали нечеткое расширение анализа реляционных концепций (fuzzy RCA) для моделирования среды BS в виде семейства решеток. Известный как мощное средство представления дан-

ных, кластеризации и анализа, fuzzy RCA использовался для представления взаимосвязей между компонентами BS не только с точки зрения возможностей QoS и QoD, но и в зависимости от их доменов и облачных сред размещения.

Также, учитывая сложный и распределенный характер среды BS, которая рассматривается как большой распределенный контейнер для различных типов сервисов и источников данных, мы внедрили наш подход BSCo поверх хорошо известной платформы Spark big data. Это позволило параллельно обрабатывать информацию о BS из нескольких облаков. Экспериментальные исследования подтвердили способность нашего подхода предоставлять безопасные и высококачественные БС в сжатые сроки.

Результат анализа потребовал формализации данных задач, а также алгоритмизации их решения с учетом особенностей, отраженных на рис. 1.2.

Сформулирована цель и задачи исследования.

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:

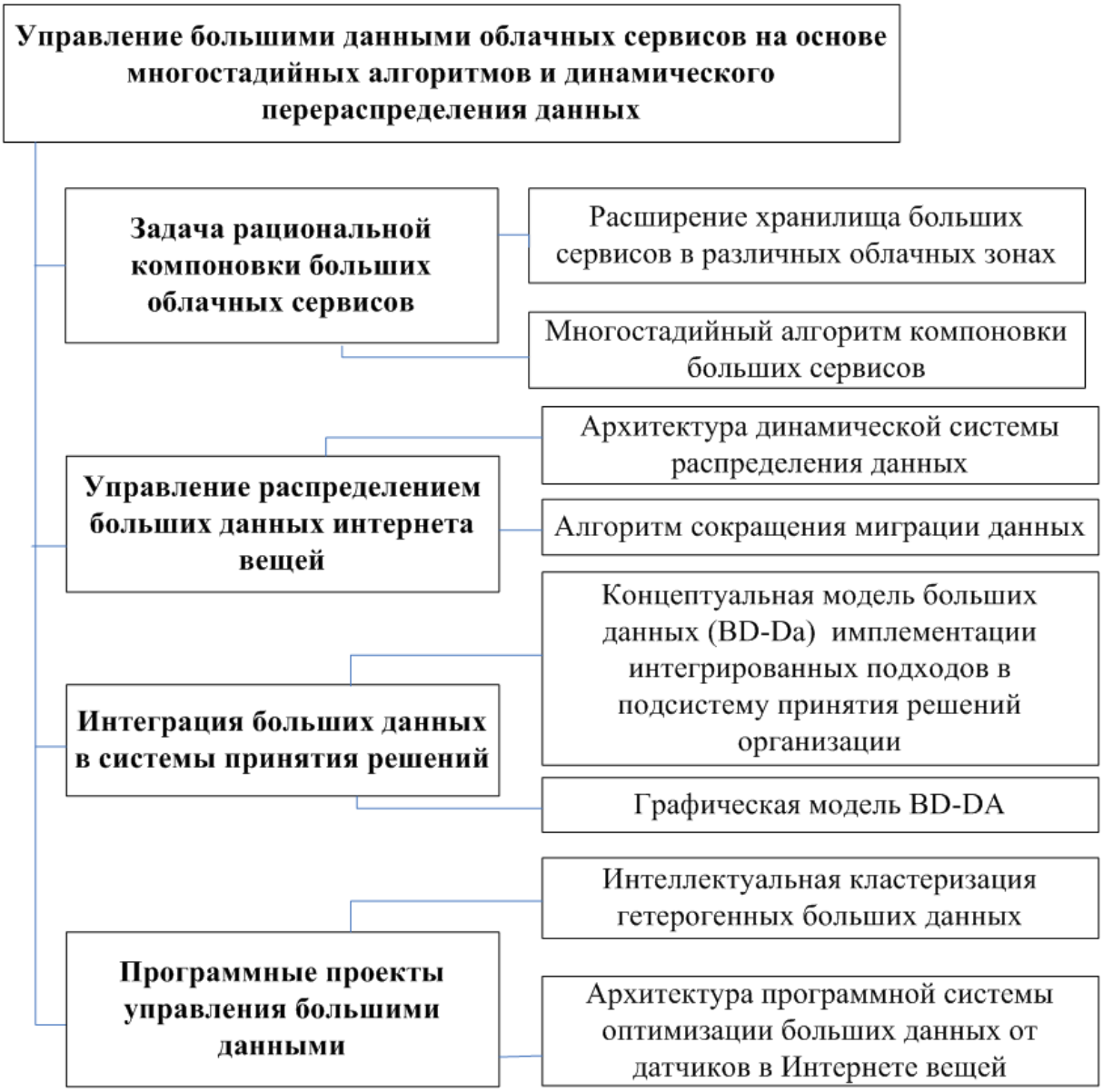


Рис. 1.2. Дизайн исследования

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 сервисов-кандидатов, комбинацию сервисов и оптимальных

выбор больших сервисов, отвечающий требованиям QoS, качества данных и безопасности.

4. Предложить архитектуру динамической системы распределения данных, обеспечивающий 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.

2. Управление большими данными при компоновке больших сервисов

2.1. Предварительный анализ больших данных

2.1.1. Качество больших данных

Как многомерное понятие, качество данных является важной характеристикой, определяющей надежность данных для принятия решений в организациях или бизнесе [2.1, 2.16, 2.40]. Согласно [2.7], высококачественные данные должны соответствовать их назначению для работы и принятия решений. Следовательно, они должны быть обрабатываемыми, точными, полными, своевременными, непротиворечивыми, заслуживающими доверия и актуальными. Качество данных измеряется с помощью различных параметров, которые оцениваются с помощью определенных показателей. В литературе представлено множество показателей качества данных, но наиболее часто используемыми из них являются точность, полнота, непротиворечивость и своевременность [2.7]. Точность и полнота оценивают данные с точки зрения их корректности и численного расширения. Например, точность веб-сервиса прогноза погоды может быть низкой из-за низкого качества данных о запасах и неполных/отсутствующих данных об истории продаж.

Полнота измеряется на уровне записей в наборе данных и определяется следующим образом [2.3]:

$$CSC = \frac{\sum_{i=1}^{K} Ci,k}{m} \cdot 100\% \qquad (2.1)$$

где m - количество записей, $C_{i,k} = 0$, если запись завершена, или равно 0, в противном случае.

Своевременность означает степень в [0,1] актуальности данных. Формально своевременность определяется следующим образом [2.3]:

$$\text{Timeliness} = \max\left(\left(1 - \frac{\text{Currency}(v)}{\text{Volatility}(v)}\right), 0\right)^{s} \quad (2.2)$$

где v представляет собой единицу данных, а s обозначает контрольное значение для чувствительности коэффициента волатильности. Его увеличение или уменьшение зависит от воздействия, оказываемого волатильностью [2.3].

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

2.1.2. RCA как формальный метод анализа данных и извлечения знаний

Как расширение, которое привносит отношения в формальный концептуальный анализ, RCA представляет собой формальный метод анализа данных и извлечения знаний [2.31]. Математическая основа RCA и решетчатое представление позволяют не только дать явное описание объектов (компонентов, свойств, ограничений и т.д.), но и связать объекты между собой. Такие объектно-атрибутивные и объектно-объектные отношения могут быть организованы с помощью набора формальных контекстов, которые могут обрабатываться отдельно или вместе, в зависимости от требований вовлеченных субъектов (например, служб, источников данных и т.д.). Эти формальные контексты затем преобразуются в иерархическое представление, которое помогает анализировать скрытые отношения между участниками [2.2].

Базовой структурой в RCA является семейство реляционных контек-

стов (RCF) [2.31]. Он состоит из набора формальных контекстов между таблицами и набора отношений между объектами, возможно, разных контекстов в RCF.

Определение 2.1.1. 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) - это наборы объектов формальных контекстов $(K_k,K_l)\in K$; $I_j\subseteq O_k\times O_l$; K_k - исходный контекст/контекст предметной области; и K_l - целевой контекст/контекст диапазона.

RCA использует реляционное масштабирование для генерации новых атрибутов r:C, которые будут добавлены к формальным контекстам. Эти новые атрибуты представляют собой описания концепций в форме $\forall$r.C или $\exists$r.C. Каждый C рассматривается как формальная концепция формального контекста в семействе RCF. В процессе итеративного реляционного масштабирования из всех формальных контекстов создается семейство решеток. Затем эти решетки могут быть использованы для извлечения скрытой информации из данных [2.31].

Определение 2.1.2. Семейство решеток (LF) - это 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 экстентом, I - это набор атрибутов, называемых намерением, и все объекты в E находятся в отношении R со всеми атрибутами в I.

Аналогичным образом, мы используем RCA для моделирования отношений между сервисами и источниками данных, а также для группировки сервисов в соответствии с их доменами и доступными источниками данных в набор формальных контекстов, зависящих от предметной облас-

ти, которые образуют семейство формальных контекстов. Однако в реальных сценариях существует большое количество (big) данных, включая большое количество облаков, провайдеров и различных типов сервисов, которые генерируют огромный объем данных (например, диалоговые и транзакционные сервисы, сервисы передачи данных, а в последнее время и BS). В такой ситуации сгенерированные формальные контексты будут большими и с перекрывающимися связями.

Одним из преимуществ применения нечеткого RCA в нашей работе является возможность применения нисходящего подхода, который позволяет анализировать минимальное количество концепций в решетке и получать минимальный набор сервисов или данных, которые перегруппированы в минимальное количество частей экстента. Используя этот метод синтаксического анализа, мы можем найти оптимальный BSCo на первых нескольких уровнях сгенерированных решеток без необходимости просматривать все решетки целиком. Это положительно сказывается на времени обработки. Таким образом, нечеткий RCA может быть принята в качестве эффективного решения даже в крупномасштабных средах и даже в случае меньшей плотности формальных контекстов сервисов и источников данных.

Формальный концептуальный анализ и его расширения (fuzzy FCA, RCA и т.д.) были успешно применены для решения крупномасштабных задач, таких как потоковая обработка и большие медицинские данные [2.9, 2.10, 2.15]. Кроме того, предыдущие работы, в которых использовались варианты FCA, такие как [2.10, 2.25, 2.28, 2.29, 2.32],, доказали, что их алгоритмы, основанные на FCA, не зависят от плотности формального контекста. Однако в определенных сценариях решения RCA могут быть сопряжены с риском высокой сложности и времени выполнения, например, когда системе необходимо повторно создавать решетку из формального контекста. Это относится к динамическим приложениям реального времени

или потоковым системам, которым необходимо перестроить структуру, чтобы принять решение или обработать новую поступающую/сгенерированную информацию. В нашем контексте нам не нужно создавать решетку при каждом поступающем запросе, потому что решетка, представляющая хранилище BS, создается только один раз в начале, то есть перед получением любого запроса пользователя. Тем не менее, чтобы учесть высокую динамичность среды обработки больших данных (например, развертывание новых сервисов, обновление или удаление существующих, управление данными, к которым осуществляется доступ, и т.д.), что приводит к некоторым обновлениям в формальном контексте сервисов и данных, мы применяем алгоритмы поэтапного построения решетки и применяем подход параллельной обработки, выполняя BSCo поверх фреймворков больших данных.

Стремясь свести к минимуму время поиска и компоновки данных в сервисе, мы объединили нечеткое расширение RCA с платформой Spark big data framework. Это помогает сократить время, затрачиваемое на изучение и использование крупномасштабной среды BS. Мотивацией для применения такой модели параллельного программирования является ее успешное сочетание с FCA, как в работе [2.10].

2.2. Создание хранилища больших сервисов

Большой сервис $S = \{s_1, s_2, \dots, s_n\}$ определяется как управляемая интеграция массивной, сложной серии сервисов, ориентированных на большие данные. Каждый сервис S_i представляет собой объединение m множества 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 $C_i = \{s^i_1, \dots, s^i_{ki}\}$. Каждая служба-кандидат s^i использует блок данных ch_t в наборе конфиденциальных источников данных $CH_z = \{ch_1, \dots \dots, ch_t\}$. Каждая служба s^i размещается у одного или нескольких

поставщиков облачных сервисов CP = CP = $\{cp_1, cp_2, \ldots, cp_t\}$. Сервисы во множестве C_i функционально схожи, но могут отличаться уровнем качества обслуживания и серьезностью утечек данных.

Служба S_i имеет набор атрибутов QoS $Q^s = \{q^s_1, q^s_2, \ldots, q^s_t\}$ (например, надежность, доступность, время выполнения и т.д.) и набор атрибутов QoD $Q^d = \{q^d_1, q^d_2, \ldots, q^d_t\}$ (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.

Сначала мы вводим L-Severity [2.38, 2.39], которая является показателем для количественной оценки утечек данных. L-Severity оценивает серьезность утечек данных на основе чувствительности, отличительного фактора и объема утечки данных.

Определение 2.2.1. Пусть $ST(a_1, \ldots, a_n)$ - исходная таблица, $L(b_1,\ldots,b_m)$ – «утекшая» таблица с $\{b_1, \ldots, b_m\} \subseteq \{a_1, \ldots, a_n\}$, $S=\{s_1,\ldots,s_m\} \subseteq A$ - множество из чувствительных атрибутов в модели данных $(b_1,\ldots,b_m)$ и DM=(T,I,HR,IR,SL,PL). Задана запись $r \in RD^{L(b_1,\ldots,a_m)}$ в $L(b_1, \ldots, a_m)$ и $DF_r^{ST(a_1,\ldots,a_n)}$, чувствительность записи r равна:

$$RSENS_r = DF_r^{ST(a_1,\ldots,a_n)} \cdot \sum_{s_i \in S} NS(s_i[r])$$

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

Определение 2.2.2. Пусть $ST(a_1,\ldots,a_n)$ - исходная таблица, а $L(b_1,\ldots,b_m)$ – «утерянная» таблица с $\{b_1,\ldots,b_m\} \subseteq \{a_1,\ldots,a_n\}$. Учитывая чувствительность записи $RSENS_r$ для каждой записи $r \in R^{L(a_1,\ldots,a_n)}$, степень утечки (L-Severity) из $(b_1,\ldots,b_m)$ вычисляется следующим образом:

$$L\text{-}Severity_L = \sum_{r \in R^{L(b_1,\ldots,b_m)}} RSENS_r$$

Основываясь на приведенной выше метрике L-Severity, введем L-Severity (DS) источника данных.

Определение 2.2.3. Пусть $DS(s_1,\ldots,s_n)$ - служба данных, осуществляющая доступ к источникам данных $D(a_1,\ldots,a_n)$ с помощью $s_1,\ldots,s_n \subseteq (a_1,\ldots,a_n)$ и $L(b_1,\ldots,b_m)$ «утечки» записей при доступе к источникам данных с помощью $\{b_1,\ldots,b_m\} \subseteq \{s_1,\ldots,s_n\}$. Учитывая чувствительность $RSENS_r$ для каждой записи $r \in R^{L(s_1,\ldots,s_n)}$, степень утечки (L-Severity) $DS(s_1,\ldots,s_n)$ вычисляется следующим образом:

$$L\text{-}Severity(DS) = \sum_{r \in R^{L(b_1,\ldots,b_m)}} RSENS_r$$

Отсюда L-Severity набора $BS=(ds_1,ds_2,\ldots,ds_n)$ определяется следующим образом:

$$L\text{-}Severity(BS) = \sum_{ds_i \in BS} L\text{-}Severity(ds_i)$$

Чтобы лучше понять степень L-Severity в контексте BS, на рис. 2.1 показан пример плана BSCo, в котором каждая из участвующих служб имеет свою собственную степень L-Severity, основанную на потребляемых источниках данных.

На рис. 2.1 также показано влияние использования различных источников данных одним и тем же сервисом на уровни L-severity. Следовательно, выбор подходящих доступных источников данных гарантирует высокое качество составленной BS.

Правильное представление сущностей среды BS (сервисов, источников данных, поставщиков и т.д.) и их свойств является первым шагом на пути к эффективной BSCo. Для этой цели в этом разделе мы представляем нечеткое RCA-моделирование среды BS (BSE). Этот процесс состоит из четырех основных этапов, как показано на рис. 2.2.

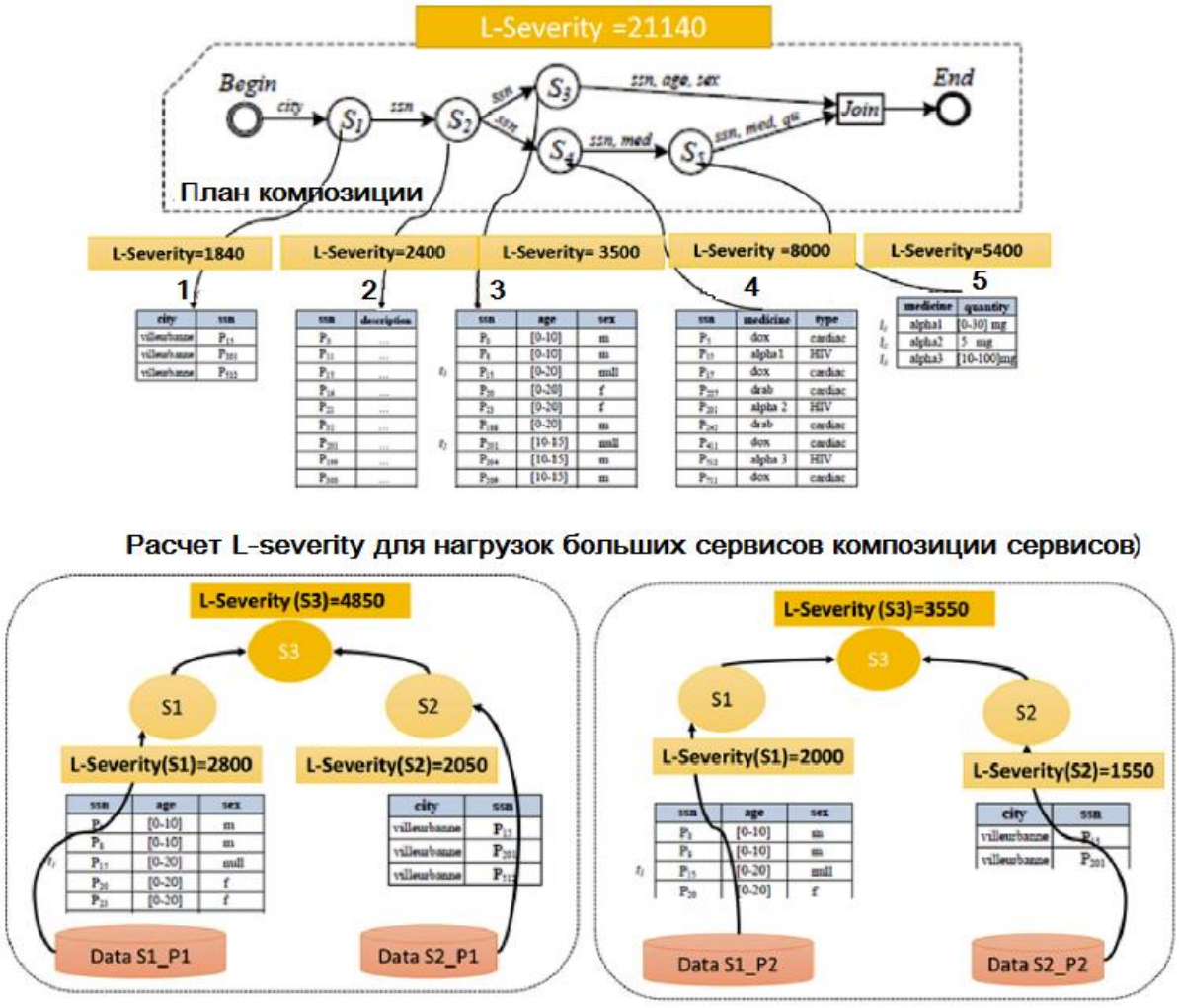


Рис. 2.1. Пример расчета L-severity для BS: а) степень утечки большого сервиса со схемой первичного размещения данных; б) степень утечки большого сервиса со вторичного размещения данных; 1-5 – данные от пользователей различных типов

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

2. Генерация облачной решетки: Этот шаг позволяет иерархически моделировать взаимосвязи между источниками данных, сервисами и их описаниями (QoS, QoD и т.д.), используя возможности группировки fuzzy RCA.

3. Расширение структуры сервисов: Оно 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: этот шаг направлен 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.

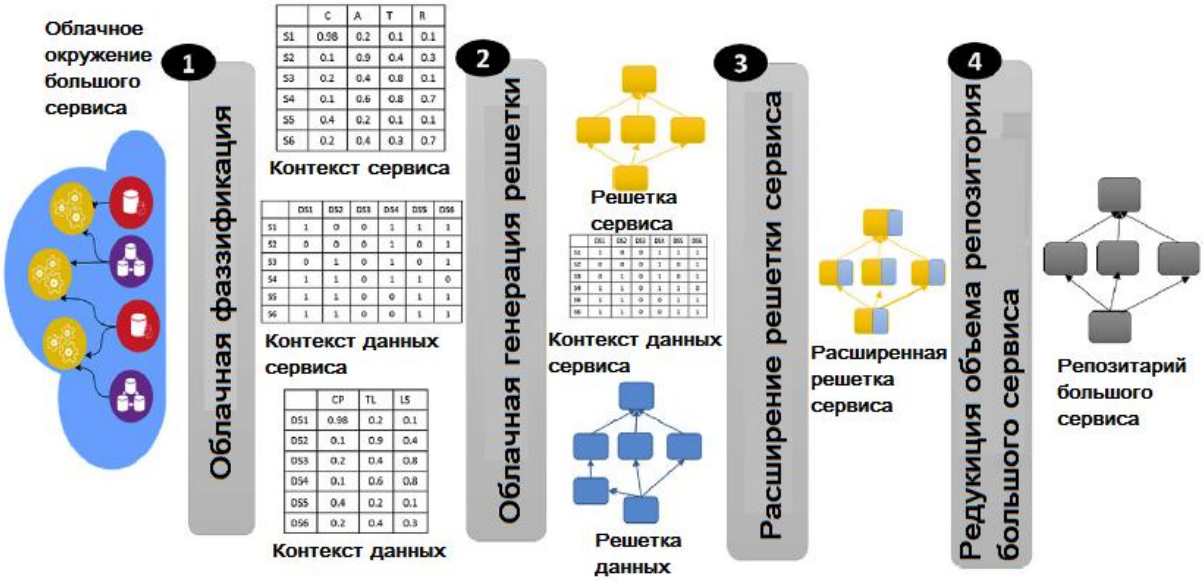


Рис. 2.2. Создание хранилища BS

2.2.1. Фаззификация облаков и создание решеток

Мы смоделировали среду BS, включая описания сервисов, источники данных и уровни их качества, как набор нечетких формальных контекстов с перекрестными таблицами, образующих семейство контекстов. Каждое облако хостинга описывается набором (семейством) нечетких формальных контекстов, из которых генерируется полное семейство lattice для группировки сервисов в соответствии с их уровнями QoS и общими источниками данных. Также генерируется реляционный формальный контекст

для выражения обязательных отношений между сервисами и используемыми ими источниками данных.

В литературе доступно несколько алгоритмов построения решетки, таких как Dowling, Faster, Bordat, Next closure, Godin и др. [2.23]. Поскольку мы имеем дело с крупномасштабными средами BS, которым требуются огромные формальные контексты для хранения информации об сервисах, источниках данных, а также обязательной информации, мы используем алгоритм Bordat, который является частью Galicia tool. Этот алгоритм хорошо работает не только в случае больших формальных контекстов, но и в случае формальных контекстов с низкой плотностью [2.23].

Первый формальный контекст, называемый “Нечеткий формальный контекст сервисов” и обозначаемый как K^S, представляет сервисы, размещенные в определенном облаке, в дополнение к их описанию QoS. Этот формальный контекст указывает, предлагает ли сервис S функциональность с определенным уровнем QoS (см. Определение 2.2.4). Второй формальный контекст, а именно “Формальный контекст нечетких данных” (обозначаемый как K^D), описывает взаимосвязи между источниками данных и их показателями качества данных (атрибутами QoD), а также степени чувствительности (L-severity) доступных данных (см. Определение 2.2.5). Третий формальный контекст, называемый “Обязательный формальный контекст” (обозначаемый как K^{BS}), описывает отношения между сервисами и источниками данных (см. Определение 2.2.6).

Определение 2.2.4 (Нечеткий формальный контекст сервисов). 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. Объект s имеет атрибут q, если s и q находятся в отношении R (обозначается sRq). Это означает, что поставщик сервиса s предоставляет определенный уровень для атрибута QoS q.

Для моделирования уровней качества данного сервиса создается нечеткий формальный контекст сервисов, обозначаемый как K^S, как показано на рис. 2.3. K^S представляет доступные сервисы. Каждый сервис имеет набор атрибутов качества, обозначаемых $Q=\{q_1,q_2,\ldots,q_n\}$. Примерами атрибутов QoS являются доступность, безопасность, надежность, стоимость, пропускная способность и т.д. В контексте BS атрибуты качества варьируются в зависимости от модели сервиса (веб, облачный, мобильный, сервис передачи данных, IoT-сервис и т.д.). Значения QoS, описанные в K^S, показывают, соответствует ли сервис определенному критерию пользователя или имеет высокое значение.

На рис. 2.3 показано, что сервис S_1 имеет высокие значения доступности (AV) и надежности (RL) (соответственно 0,86 и 0,83), но среднее значение времени отклика RT (0,76). S_1 не удовлетворяет критерию стоимости (C). С другой стороны, S_9 предлагает отличные показатели доступности и стоимости (соответственно 0,86 и 0,88), но имеет низкие значения по остальным критериям QoS (0,24 и 0,52). Пустые записи в K^S (например, S_5.RL) означают неполное описание сервиса.

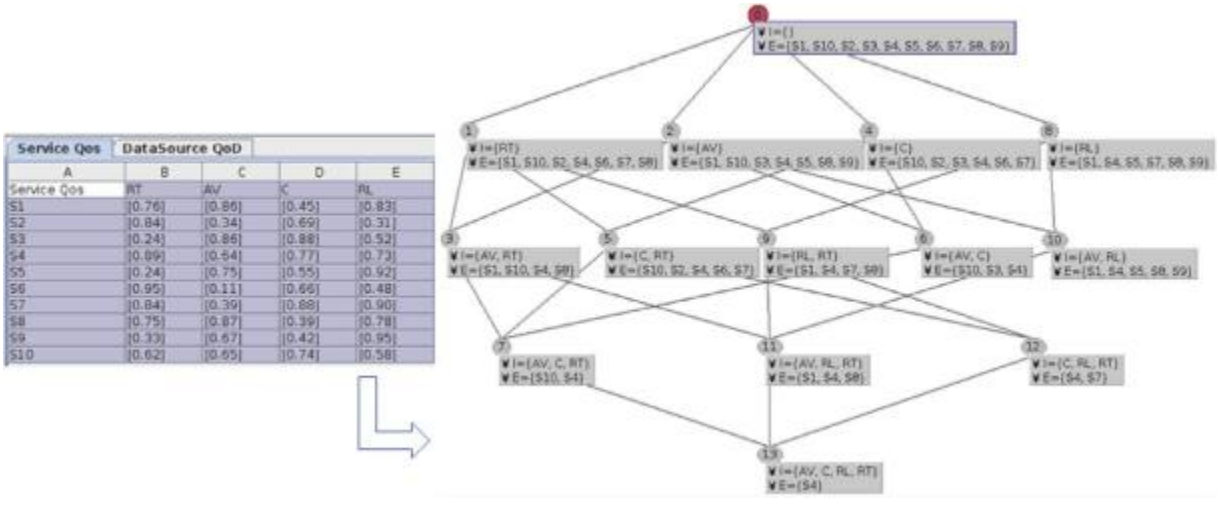


Рис. 2.3. Сервисы с нечетким формальным контекстом K^S и сервисы с нечеткой решеткой L^S

Из приведенного выше нечеткого формального контекста выводится

набор нечетких концепций для формирования иерархической структуры, называемой нечеткой решеткой сервисов (см. рис. 2.3). Рассматриваемая как небольшой кластер, каждая нечеткая формальная концепция состоит из двух частей. Часть Extent группирует доступные службы с одинаковыми уровнями QoS, в то время как часть Intent содержит атрибуты QoS, которым удовлетворяют службы в части Extent. В качестве примера пятая концепция показывает, что пять служб (S_2, S_4, S_6, S_7 и S_{10}) имеют высокую стоимость и время отклика. Одиннадцатая формальная концепция показывает, что параметрам доступности, надежности и времени отклика удовлетворяют только три службы (S_1, S_4 и S_8). Одной из сильных сторон fuzzy RCA является его способность исключать бесполезную информацию, т.е. объекты со значениями свойств, меньшими заданного порогового значения. В нашей работе сервисы с уровнем QoS, меньшим заданного порогового значения, исключаются из нечеткой решетки сервисов. Это относится к сервису S_9, который имеет низкие значения QoS (в случае атрибутов RT и C). Следовательно, нечеткая решетка будет содержать только высококачественные сервисы-кандидаты.

Определение 2.2.5 (Формальный контекст нечетких данных). 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´ Q указывает, удовлетворяет ли источник данных определенному уровню качества для определенного атрибута QoD. Объект d имеет атрибут q, если d и q находятся в отношении R (обозначается как dRq).

Нечеткий формальный контекст источников данных на рис. 2.4 показывает, что источник данных DS_7 имеет более чем средние значения своевременности и полноты (соответственно 0,75 и 0,87), но высокий риск L-Severity (0,39). С другой стороны, DS_5 предлагает отличные показатели своевременности и L-Severity (соответственно 0,94 и 0,96), но имеет сред-

нее значение полноты (0,69).

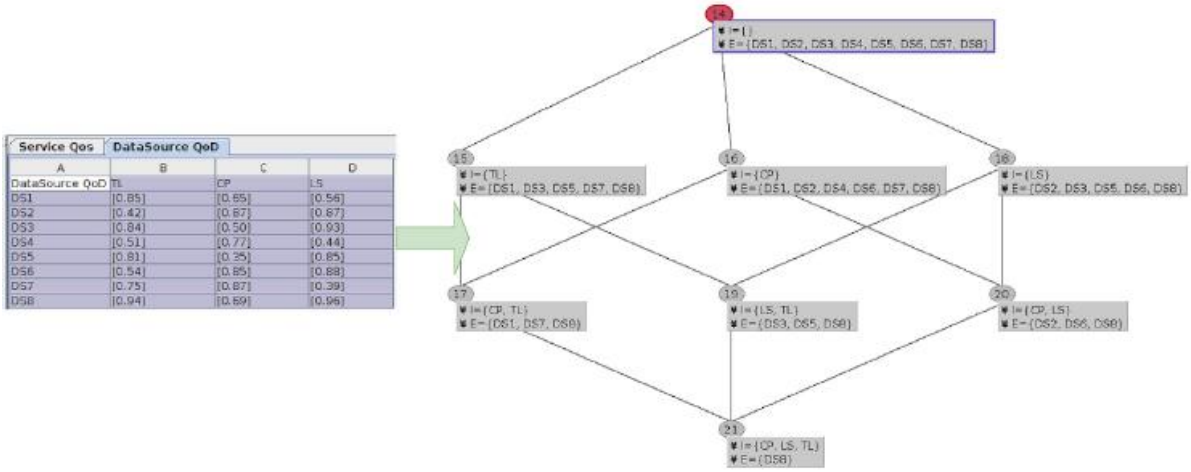


Рис. 2.4. Нечеткий формальный контекст источников данных KD; нечеткая решетка источников данных LD

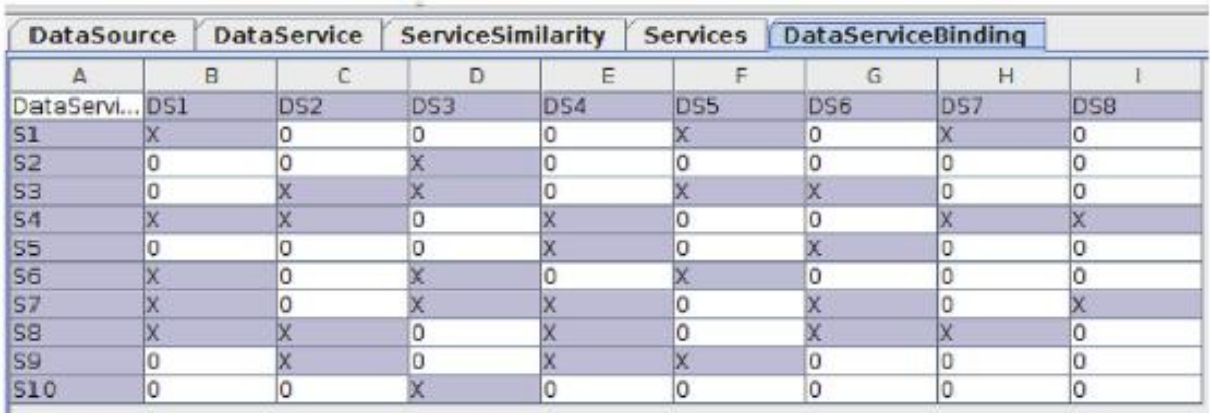
На рис. 2.4 также показана нечеткая решетка, сгенерированная из приведенного выше формального контекста данных. В разделе "Экстент" источники данных группируются в соответствии с их сходством с точки зрения уровней качества. Последние отображаются в нечеткой решетке, только если их значения превышают заданный порог. В противном случае источники данных с низким уровнем качества будут исключены из нечеткой сетки. В качестве примера возьмем источник данных DS4, который удовлетворяет только критерию полноты (CP=0,77). Следовательно, он не будет учитываться в процессе BSCo.

Нечеткие формальные контексты K^S и K^D представляют только уровни качества доступных сервисов и источников данных. Чтобы смоделировать отношения между сервисами и используемыми ими источниками данных, предлагается следующее определение обязательного формального контекста.

Определение 2.2.6 (Формальный контекст привязки). Формальный контекст привязки данных к сервису - это триплет K^B=(S,D,R), где S - набор объектов, представляющих сервисы, а D - набор атрибутов, представ-

ляющих источники данных. Бинарное отношение R=S´ D указывает, использует ли служба определенный источник данных. Объект s имеет атрибут d, если s и d находятся в отношении R (обозначается как sRd).

Пример на рис. 2.5 показывает бинарные отношения между доступными сервисами и источниками данных. Пустые записи означают, что источник данных не используется сервисом.



DataSource | DataService | ServiceSimilarity | Services | DataServiceBinding

A	B	C	D	E	F	G	H	I
DataServi...	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8
S1	X	0	0	0	X	0	X	0
S2	0	0	X	0	0	0	0	0
S3	0	X	X	0	X	X	0	0
S4	X	X	0	X	0	0	X	X
S5	0	0	0	X	0	X	0	0
S6	X	0	X	0	X	0	0	0
S7	X	0	X	X	0	X	0	X
S8	X	X	0	X	0	X	X	0
S9	0	X	0	X	X	0	0	0
S10	0	0	X	0	0	0	0	0

Рис. 2.5. Пример формального контекста для привязки сервисов к источникам данных

Мы использовали бинарные отношения в K^B для создания привязочной сетки, как показано на рис. 2.6. Формальная концепция Cpt_{86} в этой иерархии показывает, что службы S_5, S_7 и S_8 используют два общих источника данных (DS_4 и DS_6). В формальной концепции Cpt_{79} раздел Intent содержит только один источник данных (DS_3), что означает, что службы в части Extent (S_2, S_3, S_6, S_7 и S_{10}) имеют одинаковые политики управления доступом к DS_3. Следовательно, такое представление в виде решетки поможет объединить высококачественные сервисы, которые потребляют минимальное количество источников данных, как мы покажем в разделе 2.3.

2.2.2. Увеличение размера решетки и сокращение объема хранилища

Этот шаг состоит в обогащении каждой решетки служб в семействе lattice полезной информацией о потребляемых источниках данных (напри-

мер, QoD, L-Severity). Сокращение объема репозитория BS заключается в применении определенных операций к формальным концепциям, чтобы уменьшить избыточность и дублирование концепций. Результатом этого шага является набор расширенных решеток, представляющих среду BS. Поскольку в разных облачных зонах может быть доступно несколько экземпляров сервисов и источников данных, нам необходимо оценить сходство между формальными концепциями, которые объединяют эти сервисы и источники данных. Итак, мы использовали сходство по Жаккарду [2.24], как показано в нашем алгоритме увеличения решетки (см. строки 6-8 в алгоритме 2.1).

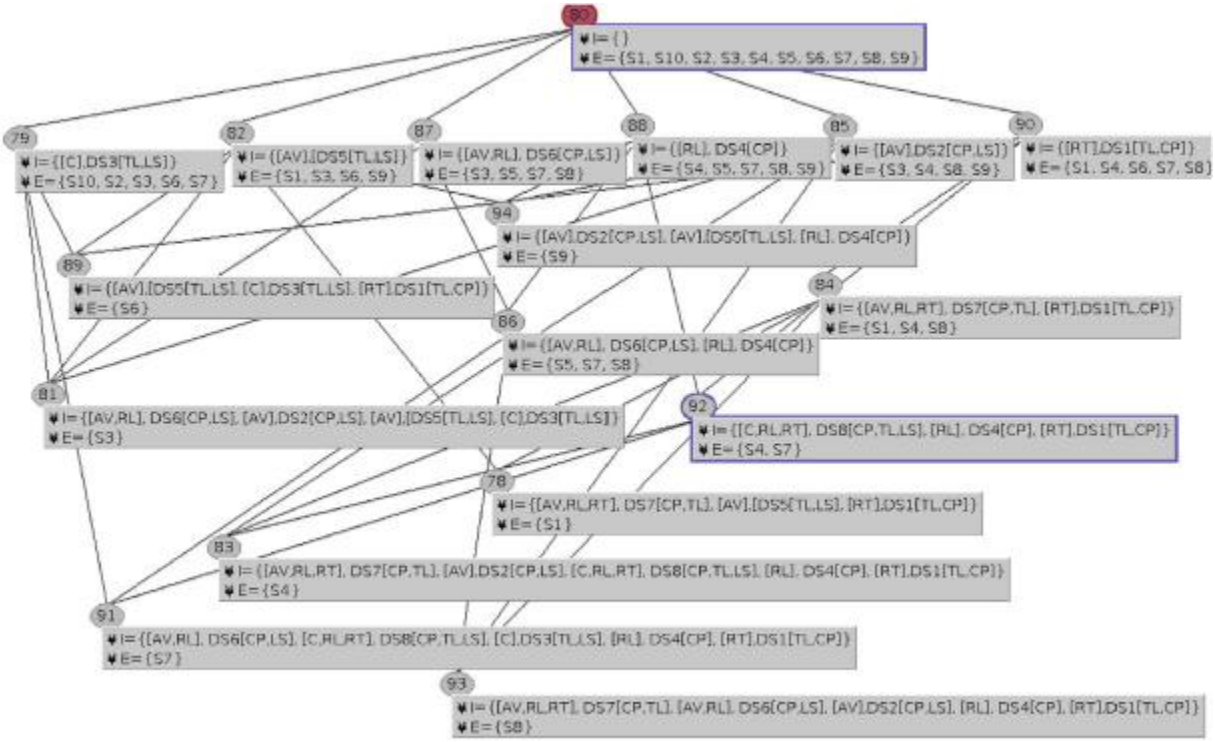


Рис. 2.6. Расширенные сервисы

Алгоритм 2.1. Сервисы по расширению решетки

1: **Input:** L^S сервисы решетки; L^{DS} решетка источников данных, L^B сервисы-решетка привязки данных.

2: **Output:** Расширенная решетка BS L^B

3: **for each** концепции C_i в решетке сервисов L^S **do**

4: **for each** концепции C_j в решетке источников данных L^{DS} **do**

5: **for each** концепции C_k в связующей решетке L^B **do**
6: **if** C_j.Intent ! =φ **then**
7: $\text{Jaccard}_{DS}(C_i, C_k) = \frac{|C_i \cap C_k|}{|C_i \cup C_k|}$
8: $\text{Jaccard}_{S}(C_j, C_k) = \frac{|C_j \cap C_j|}{|C_k \cup C_k|}$
9: **if** $(\text{Jaccard}_{DS} - \text{Jaccard}_{S}) \leq \rho$ **then**
10: C_i.Extent = C_i.Extent $\cup$ C_j.Extent
11: C_i.Intent = C_i.Intent $\cup$ C_j.Intent
12: **end if**
13: **end if**
14: **end for**
15: **end for**
16: **end for**
17: **Return** расширенная сервисная решетка L^B

Для каждой формальной концепции в таблице сервисов L^S и таблице источников данных L^D алгоритм 2.1 проходит через привязочную таблицу L^B, чтобы дополнить таблицу сервисов используемыми источниками данных и их уровнями качества и L-Severity (строка 3). Алгоритм использует сходство по Жаккарду [2.24] для вычисления сходства между источниками данных в решетке привязки и источниками данных в решетке источников данных (строка 7). Аналогичным образом алгоритм вычисляет сходство между сервисами в решетке L^S и сервисами в привязочной решетке L^B (строка 8). Если разница между сходствами ниже определенного порога (строка 9), текущая формальная концепция C_i дополняется источниками данных и их атрибутами QoD. Эта задача выполняется для каждой концепции C_i в таблице сервисов. Затем алгоритм добавляет источники данных, относящиеся к экстенту j-й концепции, к набору аналогичных сервисов (строка 10). Кроме того, информация о качестве (Intent-часть C_j) этих источников данных добавляется в Intent-часть C_i (строка 11). Этот процесс повторяется до тех пор, пока все службы не будут дополнены полезной информацией об используемых ими источниках данных, а также об уров-

нях качества обслуживания и L-Severity. Результатом работы алгоритма 1 является новая расширенная сетка (см. рис. 2.6), которая содержит только доступные сервисы высокого качества с наилучшими используемыми источниками данных.

Расширенную сетку можно рассматривать как набор небольших кластеров, называемых формальными понятиями. Каждый из них состоит из двух частей. В разделе Extent представлены сервисы-кандидаты с высоким уровнем QoS. Некачественные сервисы уже были исключены из таблицы сервисов на основе степени достоверности атрибутов QoS (см. раздел 2.2.1). В разделе "Назначение" содержатся источники данных высокого качества, которые обычно используются сервисами в разделе "Экстент". В разделе "Цель" мы также можем найти информацию о удовлетворенных атрибутах QoD, в дополнение к L-Severity, если она удовлетворена. Например, концепция Cpt_{15} показывает, что три службы (S_2, S_3 и S_5) получают доступ к одному и тому же источнику данных (DS_3). Последний удовлетворяет двум параметрам QoD (своевременность и L-severity). В концепции Cpt_{19} раздел "Объем" содержит службы S_2 и S_5. Последние обычно используют два источника данных: DS_3 с высоким уровнем своевременности и L-Severity и DS_4, который характеризуется высокой степенью полноты.

В следующем разделе мы покажем, как обрабатывается семейство расширенных решеток, чтобы извлекать и комбинировать лучшие сервисы, одновременно сокращая количество доступных источников данных и максимизируя их качество.

2.3. Процесс компоновки веб-сервисов

В этом разделе описываются основные этапы процесса компоновки веб-сервисов. Как показано на рис. 2.7, механизм компоновки принимает в качестве входных данных пользовательский запрос с точки зрения функциональности, требований к качеству обслуживания и QoD и выполняет

поиск в репозитории BSCo доступных объектов, которые могут сформировать запрашиваемую BSCo.

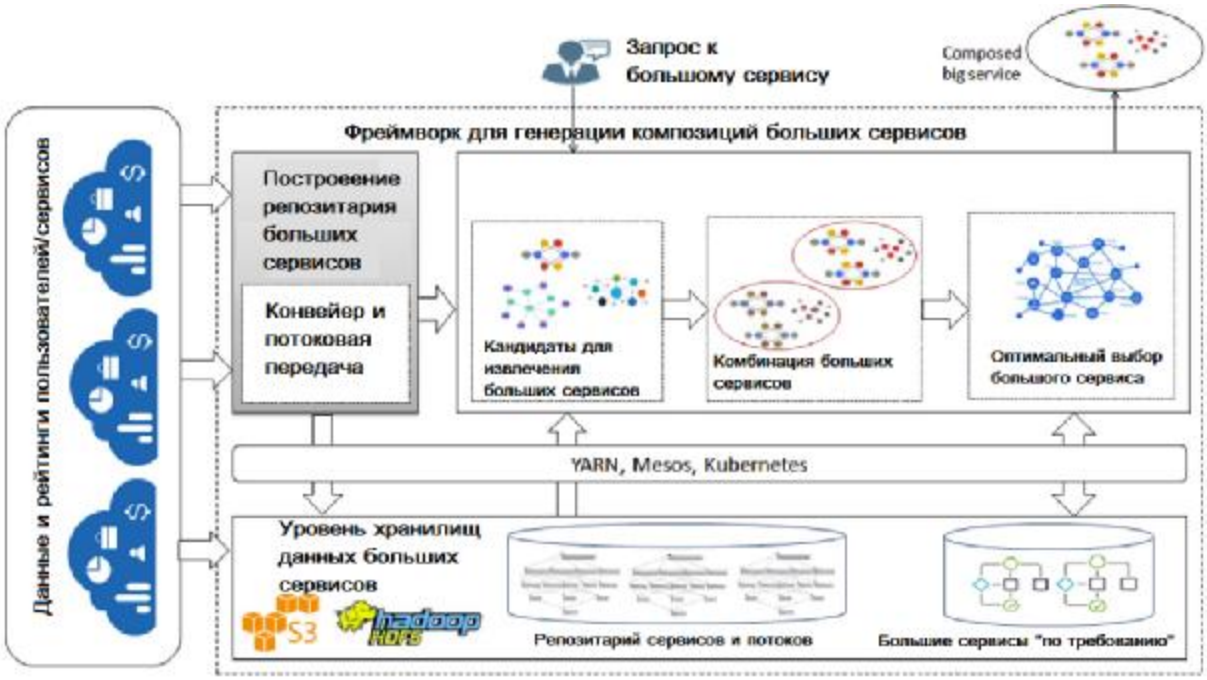


Рис. 2.7. Процесс компоновки веб-сервисов

Процесс компоновки веб-сервисов вкратце 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: этот заключительный шаг направлен на выбор оптимального плана компоновки, который не только отвечает требованиям QoS, QoD и безопасности, но и минимизирует количество источ-

ников данных и затраты на связь между задействованными облачными провайдерами. Более подробная информация об этих шагах приведена в следующих подразделах.

2.3.1. Извлечение сервисов-кандидатов

Этот шаг 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.

Алгоритм 2.2 проходит через каждую расширенную решетку, чтобы выделить подиерархии решетки, содержащие сервисы в каждом облаке, которые отвечают требованиям пользователей. Применяется нисходящий подход, гарантирующий обработку формальных концепций, имеющих наибольшие размеры экстента в каждой решетке. Такой подход к сортировке с уменьшением количества запросов ускоряет определение максимального количества запрашиваемых сервисов, которые используют минимальное количество общих источников данных.

Алгоритм проходит через каждую формальную концепцию в таблице L^B, чтобы определить подходящую BS (строка 4). Он проверяет, содержит ли часть Extent текущей концепции все или некоторые из требуемых сервисов (строка 5). Затем алгоритм оценивает для каждой службы L-Severity используемых ею данных, которые относятся к части намерений (строка 7). Если используемые источники данных не удовлетворяют требованиям безопасности (L-Severity меньше порогового значения), сервис и его источники данных не будут учитываться в остальной части процесса создания (строка 8). Как только текущая формальная концепция будет доработана, она будет добавлена к набору сервисов, найденных на данный момент (строка 11). Этот процесс повторяется до тех пор, пока не будут

уточнены все формальные концепции, и, наконец, будет возвращена решетчатая подиерархия L^B, содержащая только лучшие сервисы-кандидаты и их источники данных (строка 16).

Алгоритм 2.2. Извлечение сервисов-кандидатов

1: **Input:** Расширенная сервисная решетка L^B, пользовательский запрос Q_{BS} с требованиями QoS и QoD r
2: **Output:** Субиерархия решетки сервисов-кандидатов L^B
3: S_a =Æ
4: **for each** Концепции C_i в L^B **do**
5: **if** (Q_{BS} − S_a) ∩ C_i.Extent ≠ Æ **then**
6: **if** S_jÏ Q_{BS} или L−severity(C_i.Intent)≤T **then**
7: удалить S_j из C_i.Extent
8: **end if**
9: добавить C_i.Extent в S_a
10: **else**
11: удалить C_i из L^B
12**:** **end if**
13: **end for**
14: **Return** Субиерархия решетки сервисов-кандидатов L^B

2.3.2. Соединение больших сервисов

На этом этапе службы-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. Мы разработали алгоритм 2.3, который использует в качестве входных данных подиерархию расширенной решетки L^B и пользовательский запрос Q_{BS}. Результатом работы этого алгоритма является набор составленных оценок, которые позже будут оценены, чтобы вернуть ту, которая наберет наибольшее количество баллов.

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}=Æ
4: Сортировка в L^B формальных концепций в порядке убывания размеров их экстентов
5: **while** ($L^B \neq$ Æ) **do**
6: S_a=Æ
7: D_a=Æ
8: **for each** концепции C_i в L^B **do**
9: **if** ($Q_{BS}-S_a$) ∩ C_i.Extent≠Æ **then**
10: добавить C_i.Extent в S_a
11: add C_i.Intent to D_a
12: удалить C_i из L^B
13: **if** (S_a =Q_{BS} **and** D_a =Q_{BS}.data) **then**
14: $S^{BS} = S^{BS} \cup S_a$
15: **break**
16: **end if**
17: **end if**
18: **end for**
19: **end while**
20: **Return** S^{BS}

Структура BS может быть найдена в нескольких комбинациях экстентов, 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.

Для этого алгоритм 2.3 проходит через расширенную решетку и на-

чинает с анализа формальных концепций, имеющих наибольшее количество запрашиваемых сервисов (строка 8). Затем для каждой формальной концепции проверяется, содержит ли ее раздел Extent необходимые сервисы (или их часть) (строка 9). В этом случае службы-кандидаты (C_i.Extent) и их источники данных (C_i.Intent) добавляются к текущей комбинации BS (строки 10-11), а подиерархия решетки обновляется (строка 12). Поскольку некачественные сервисы и источники данных с высокими значениями L-Severity уже были исключены во время генерации расширенной решетки, алгоритм обрабатывает только высококачественные сервисы и источники данных. Следовательно, он проверяет, соответствует ли текущая комбинация данных (S^a), после ее формирования, требованиям к данным (строка 13), чтобы решить, будут ли источники данных и связанные с ними сервисы учитываться в BSCo (строка 14). Если в построенной BS отсутствуют некоторые источники данных, которые были исключены из расширенной сетки (по причинам качества или L-Severity), алгоритм может выполнить этап настройки, который заключается в выборе некоторых источников данных низкого качества для заполнения недостающих.

Поскольку из набора формальных концепций может быть получено несколько возможных значений BS, процедура синтаксического анализа повторяется, чтобы найти возможные комбинации значений BS в оставшихся формальных понятиях (строка 5). В лучшем случае запрашиваемая информация о безопасности может быть найдена в рамках одной формальной концепции с общими источниками данных, что оказывает большое влияние на уровень безопасности (схожие политики безопасности). Наихудший случай - это когда каждый сервис вместе с используемыми им источниками данных находится в отдельной формальной концепции, что означает большое количество разнородных источников данных с различными политиками безопасности.

Результатом работы алгоритма 2.3 является набор комбинаций сте-

пеней (строка 20), каждая из которых обозначает кандидата в BSCo. Последние оцениваются и ранжируются в соответствии с их значениями QoS, QoD и L-Severity, как мы покажем в следующем подразделе.

2.3.3. Оптимальный выбор больших сервисов

Существующие подходы к выбору сервисов как в веб-, так и в облачных средах определяют их функции оценки только на основе критериев QoS, таких как стоимость и время отклика. Ни один из них не учитывает качество используемых источников данных в качестве входных данных для своих функций оценки. В рамках нашего подхода мы определяем новую функцию, которая вычисляет не только локальные и глобальные показатели QoS BSCo, но и их показатели QoD.

На наш взгляд, качество BSCo - это последовательность кортежей, $Q=\{< QoS_1, QoD_1>, <QoS_2, QoD_2> , ... , <QoS_k, QoD_k >\}$, где QoS_1, QoS_2, … , QoS_k - уровни качества сервисов компонентов, а QoD_1, QoD_2, … , QoD_k представляют уровни QoD для источников данных, используемых каждой службой. Оптимальное качество BS - это последовательность кортежей, представляющих наивысшие значения QoS и QoD.

Как и в веб-сервисах, BS следует модели YAWL [2.11] и имеет четыре основные структуры. В последовательной структуре задачи BS выполняются последовательно. В циклической структуре задача BS выполняется итеративным образом. В параллельной структуре набор задач выполняется одновременно, и как только все они будут завершены, будут выполнены следующие задачи в BS. В структуре ветвей должна быть выполнена только одна задача из подмножества задач.

Поскольку структура BS состоит из четырех вышеперечисленных базовых структур, QoS конкретного состава служб может быть рассчитано на основе QoS ее базовых структур. Например, глобальное время отклика BS в случае последовательной структуры вычисляется путем сложения време-

ни отклика служб компонентов, тогда как в случае структуры ветвей учитывается минимальное значение времени отклика.

Качество BS (QoBS) зависит от качества входящих в него сервисов. На последнее влияет качество используемых источников данных. Для этого мы начинаем с расчета качества источников данных (QoD), чтобы использовать его при расчете уровней QoS. Ожидается, что такие атрибуты QoD, как своевременность (TL), полнота (CP) и согласованность (CS), должны быть максимально возможными, в то время L-Severity (LS) должна быть сведена к минимуму.

2.3.4. Изучение вычислительной сложности

Показатель качества используемого источника данных вычисляется следующим образом:

$$QoD_i^s = w_1 Cp_i^s + w_2 T_i^s + w_3 Co_i^s + w_4 I_i^s \qquad (2.3)$$

где wj (j=1...4) обозначает веса, присвоенные каждому критерию качества. Cp^s_i, T^s_i, Co^s_i и I^s_i обозначают соответственно атрибуты полноты, своевременности, согласованности и целостности для i-го источника данных, используемого s-й службой.

Значения атрибутов QoD не определяются пользователем, поскольку его основное внимание уделяется функциональному поведению и качеству обслуживания составленной BS, а скорее они определяются структурой компоновки в зависимости от домена BS. На самом деле, важность атрибутов QoD может варьироваться в зависимости от домена. Например, в потоковых сервисах и сервисах, основанных на социальных сетях, полнота и своевременность доступных данных важнее, чем атрибут точности. Это отличается от других сервисов, таких как службы прогнозирования погоды и трафика, в которых точность передаваемых потоковых данных имеет первостепенное значение.

Оценка QoS каждого компонента BS рассчитывается следующим об-

разом:

$$QoS = w_1 C + w_2 A + w_3 T + w_4 R \qquad (2.4)$$

где w_k (k=1, 2, 3, 4) отображает степень важности атрибутов QoS: стоимость, доступность, время отклика и надежность. Эта формула используется в зависимости от структуры рабочего процесса BS. В уравнении (2.5) мы приводим пример последовательной структуры. Подробности о других структурах рабочего процесса описаны в нашей предыдущей работе по созданию веб-сервиса [2.30].

Здесь n представляет собой количество служб, участвующих в составе потенциальной BS, m отображает количество источников данных для i-й компонентной службы, QoS_i - это оценка i-й службы, QoD_i^j - это оценка качества j-го источника данных, используемого i-й службой. Эти баллы рассчитываются с использованием значений серьезности утечки из каждого источника данных. Чем ниже уровень L-severity, тем выше общая оценка составленного BS. Наконец, кандидаты на получение BSCo сортируются в соответствии с глобальными значениями QoBS, и пользователю возвращается BSCo, набравший наибольшее количество баллов.

$$QoS = w_1 \sum_{i=1}^{N} C_i + w_2 \prod_{i=1}^{N} A_i + w_3 \sum_{i=1}^{N} T_i + w_4 \prod_{i=1}^{N} R_i \qquad (2.5)$$

где Q_k (k=1, 2, 3, 4) - значения атрибутов QoS для состава сервисов, n - количество задач, Q_{ki} (k=1,..., 4; i=1,..., n) обозначает k-е значение QoS для сервиса-кандидата в i-й задачу, и $\sum_{k=1}^{4} w_k = 1$, $0<w_k<1$).

В (2.5) показано, что для процесса обслуживания с n последовательными задачами, $Q_1 = w_1 \sum_{i=1}^{N} C_i$, $Q_3 = w_3 \sum_{i=1}^{N} T_i$ являются глобальными уровнями качества для атрибутов стоимости и времени отклика, соответственно. $Q_2 = w_2 \prod_{i=1}^{N} A_i$ и $Q_4 = w_4 \prod_{i=1}^{N} R_i$ являются глобальными уровнями качества для атрибутов доступности и надежности, соответственно.

На основе выражений (2.3) и (2.4) общая оценка составной системы BS вычисляется путем объединения значений QoS для составляющих служб и значений QoD для их потребляемых данных (см. выражение (2.6)). Общий балл также корректируется в зависимости от значений L-Severity источников данных.

$$QoBS = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{QoS_i \cdot QoD_j^i}{m \sum_{R^{L(b_1,\ldots,b_b)}} RSENS_r} \qquad (2.6)$$

Здесь n представляет количество служб, участвующих в составе потенциальной BS, m отображает количество источников данных для i-й компонентной службы, QoS_i - это оценка i-й службы, QoD_j^i это оценка качества j-го источника данных, используемого i-й службой. Эти значения рассчитываются с использованием значений серьезности утечки из каждого источника данных. Чем ниже уровень L-severity, тем выше общая оценка составленного BS.

Наконец, кандидаты на получение BSCo сортируются в соответствии с глобальными значениями QoBS, и пользователю возвращается BS, набравший наибольшее значение.

2.3.5. Сложность алгоритмов и их влияние на производительность композиции

Было доказано, что Fuzzy RCA является эффективным решением, которое представляет, группирует и обрабатывает информацию в системе обработки [2.8]. Однако за эти преимущества иногда приходится платить. Действительно, разбор семейства сложных решеток означает обработку огромного количества формальных понятий, особенно в случае высокой плотности формальных контекстов.

Поскольку мы имеем дело с параллельным подходом, который заключается в разделении связанных с BS данных на набор формальных

контекстов, сложность процесса BSCo будет значительно снижена, поскольку секционированные данные будут обрабатываться параллельно с использованием одной из существующих платформ обработки больших данных. В этом подразделе мы изучаем сложность определенных алгоритмов и их влияние на производительность системы компоновки.

• Извлечение сервисов-кандидатов: стоимость этого шага в основном зависит от размера запроса BS ($|Q_{BS}|$), а также от количества сервисов-кандидатов (N_S) и источников данных (N_D). В худшем случае для извлечения сервисов-кандидатов будут обработаны формальные концепции $|L^B|$. Для каждой концепции алгоритм проверяет, содержит ли раздел Extent все (или часть) запрошенные сервисы. Сложность этой операции составляет $O(|L^B|•|Q_{BS}|)$. Поскольку для каждого источника данных в разделе Intent оцениваются значения L-Severity, этот процесс будет повторен для каждого источника данных, участвующего в компоновке. Следовательно, сложность этого алгоритма равна $O(|L^B|•|Q_{BS}|•N_D)$.

• Комбинирование BS: сложность этого шага зависит от размера подиерархии расширенной решетки (L^B), а также от количества извлеченных сервисов (N_S) и доступных источников данных (N_D). Операция комбинирования разделов экстента направлена на проверку того, образуют ли они требуемую BS. В худшем случае ее стоимость составляет ($|L^B|•|Q_{BS}|$) (т.е. каждый запрашиваемый сервис относится к отдельному объему). Поскольку несколько BSCo с разными уровнями качества могут быть выделены из субрешетки путем объединения различных уровней, операция объединения учитывает все сервисы-кандидаты в L^B. Это общая сложность, равная $O(|L^B|•|Q_{BS}|)$.

2.4. Экспериментальное исследование

2.4.1. Реализация и протокол эксперимента

Чтобы проверить осуществимость подхода BSCo, мы использовали язык Java для разработки движка компоновки поверх Spark 2.4 big data framework 5 (см. рис. 2.8), который развертывается в автономном режиме. Мы также использовали библиотеку fuzzy FCA 6 для построения и анализа определенных нечетких решеток (см. раздел 2.3). Кроме того, AWS S3 (Amazon Simple Storage Service) используется для хранения реестра BS (то есть семейства нечетких решеток).

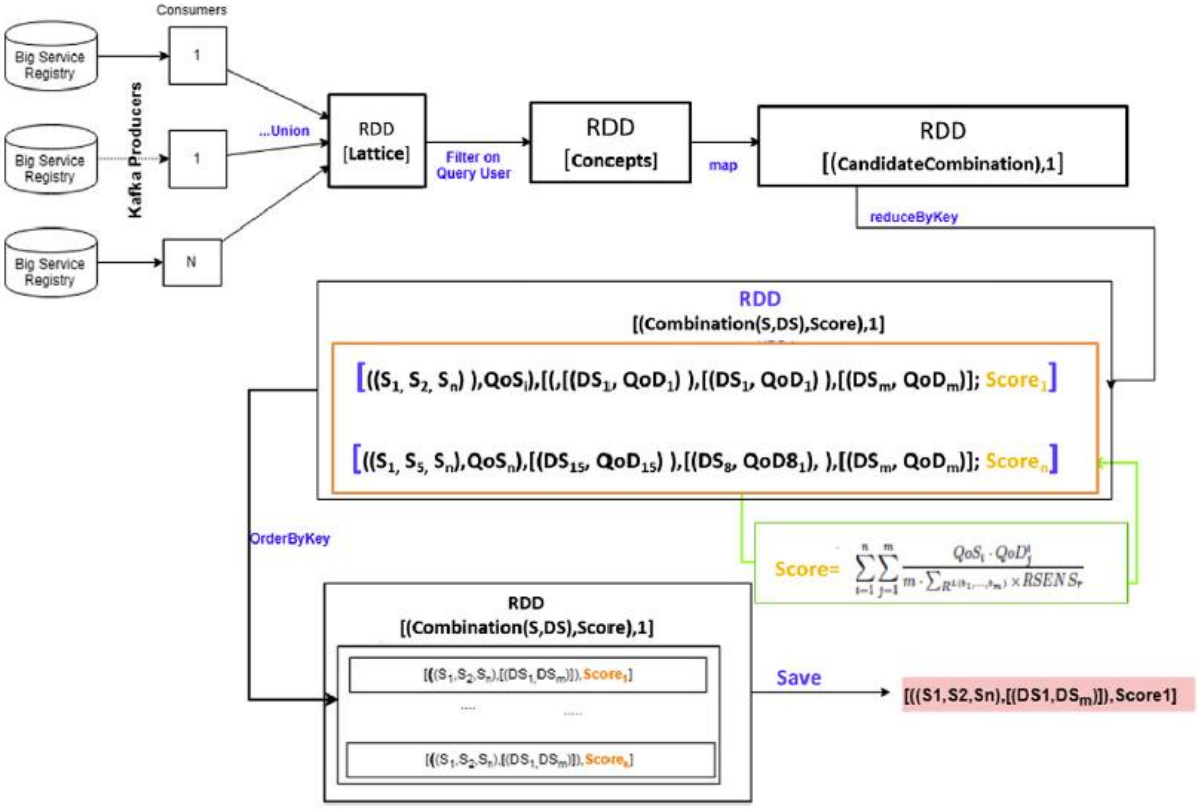


Рис. 2.8. Распределенный движок BSCo

Ключевой концепцией Spark является RDD (Устойчивые распределенные наборы данных), которые представляют собой неизменяемые коллекции элементов, секционированных и распределенных по узлам кластера Spark [2.19]. Как показано на рис. 2.8, RDD сначала создаются на основе формальных концепций в расширенных решетках, которые хранятся в ви-

де файлов в HDFS/S3. Затем, выполняя этап извлечения сервисов-кандидатов (операция сопоставления), RDD фильтруются и преобразуются в новые RDD, каждый из которых представляется в виде кортежа, принимающего форму пар <ключ, значение>. Сгенерированные кортежи {Комбинация (сервисы, источники данных), оценка} содержат выбранные сервисы, которые соответствуют запросу пользователя. После этого в операции уменьшения оценка каждого кандидата BS вычисляется с использованием уравнения (2.6). Наконец, путем сортировки полученных оценок (операция orderByKey) генерируется новый RDD, содержащий оптимальный BSCo, и доставляется конечному пользователю после сохранения в HFDS/S3.

Все эксперименты проводились на реальном кластере с использованием Amazon AWS EMR, ведущей в отрасли облачной платформы для обработки больших данных. Кластер состоит из 8 машин EC2. Каждая из них оснащена 8 процессорами, 16 ГБ оперативной памяти и 100 ГБ локального хранилища. Для оценки мы использовали значения параметров EMR Spark Cluster по умолчанию.

Некоторые из этих параметров были изменены, чтобы проанализировать их влияние на производительность композитного движка.

Мы сравнили нашу работу с подходами, описанными в [2.18, 2.21], которые являются единственными подробными работами, опубликованными для BSCo. Первый подход: MR-EA/G, который был реализован на кластере Hadoop [2.21], сочетает в себе модель программирования MapReduce и эволюционный алгоритм с управляемой мутацией для выполнения BSCo с поддержкой QoS. Второй подход, называемый CVMIP, основан на коэффициенте вариации и использует смешанное целочисленное программирование для выбора оптимального BSCo [2.18]. Основываясь на нашем обзоре литературы и в соответствии с вышеупомянутыми работами [2.18, 2.21], набор данных BS отсутствует. С этой целью в наших экспериментах

мы следовали той же логике, что и в [2.18, 2.21], используя комбинацию синтетических данных и широко используемого набора данных реального мира, называемого WS-DREAM [2.45], из домена веб-сервиса.

В репозитории WS-DREAM хранятся 3 набора данных: обзорный набор данных, набор данных журнала и набор данных QoS. Последний описывает реальные оценки QoS в 4500 веб-сервисах от 142 пользователей за 64 последовательных отрезка времени. WS-DREAM предоставляет и другие данные, такие как время обращения к службе, загруженность службы, местоположение пользовательской службы, частота обращения к службе и количество успешных обращений. Поскольку набор данных WS-DREAM не содержит данных о BS или, по крайней мере, о других связанных моделях обслуживания, таких как службы передачи данных, мы случайным образом сгенерировали данные, относящиеся к контексту BS. Мы также дополнили набор данных WS-DREAM дополнительной информацией о доступных источниках данных, которая необходима главным образом для создания нечеткой решетки источников данных и привязки нечеткой решетки. Чтобы можно было построить семейство lattice, данные, предоставляемые WS-DREAM, а также данные, сгенерированные случайным образом, преобразуются в XML-файлы, которые представляют семейство нечетких формальных контекстов.

2.4.2. Оценка производительности

В первой серии тестов мы оценили качество решения BSCo с поддержкой QoD по сравнению с традиционным подходом, который учитывает только атрибуты QoS (см. рис. 2.9). Кроме того, цель этого набора тестов - проверить, повлияет ли увеличение размера хранилища BS и плотности формальных контекстов на оценку оптимального BSCo и время вычислений. С этой целью были выполнены тесты с разным количеством сервисов (от 100 до 1000) и абстрактные задания (25, 50, 75, 100) для разных

значений плотности (от 20% до 50%). Результаты по средним баллам и проценту использованных источников данных представлены на рис. 2.10 и рис. 2.11 соответственно.

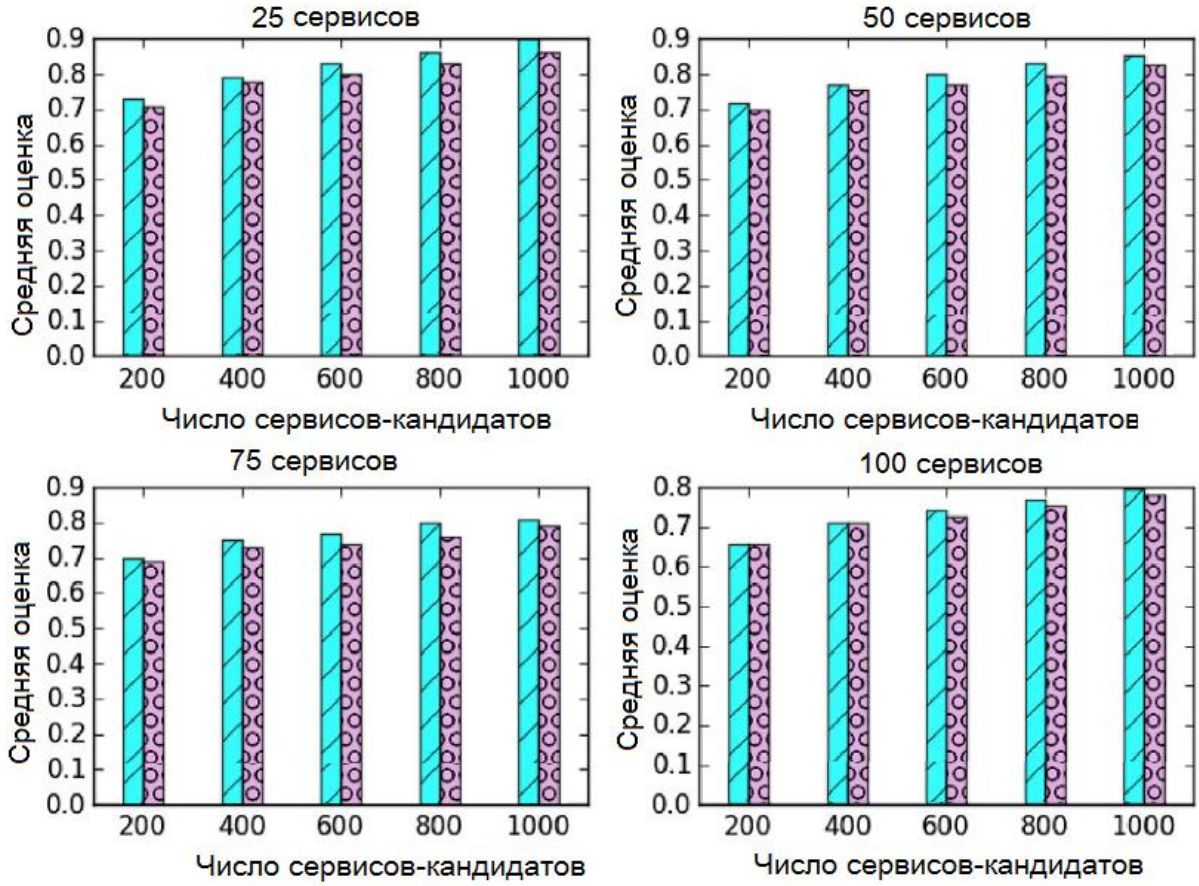


Рис. 2.9. BSCo с поддержкой QoD в сравнении с традиционным BSCo: - BSCo, учитывающий QoD; - традиционный BSCo

Влияние параметров QoD. Из рис. 2.9 видно, что качество составленной BSC_O значительно улучшилось, когда мы включили измерения QoD в процесс составления. Действительно, BSCo, поддерживающий QoD, учитывает качество как сервисов, так и источников данных. Кроме того, использование степени достоверности, предлагаемой fuzzy RCA, позволило исключить сервисы с уровнем QoS ниже, чем требуется пользователю. Более того, источники данных с более низкими значениями QoD (полнота, своевременность и т.д.) были исключены из нечеткой решетки источников данных. Таким образом, анализ расширенной сетки, в которой группиру-

ются только высококачественные сервисы и источники данных, помог обеспечить оптимальные показатели BSCo, которые впоследствии были уточнены с учетом L-Severity. В отличие от нашего подхода, основанного на оценке качества обслуживания, традиционный подход BSCo оценивает возможные решения, принимая во внимание только критерии качества обслуживания (время отклика, надежность, доступность и т.д.). На рис. 2.9 четко показано, что на средние оценки этих решений влияет отсутствие информации о качестве обслуживания и уровнях сложности.

Фактически, сервисы, выбранные для составления композиции, могут быть высокого качества, но с ненадежным поведением (низкая степень L-Severity) и источником данных (уровни QoD). Это относится к подходам CVMIP [2.17] и MREA/G [2.38], как мы покажем в разделе 2.4.3.

Влияние на оценку BS. Что касается качества BSCo (см. рис. 2.10), то возможности группировки fuzzy RCA, особенно в условиях высокой плотности формальных контекстов, позволили объединить сервисы, использующие общие источники данных, с высокими уровнями QoD и низкими значениями L-Severity. Фактически, сокращение числа источников данных гарантирует низкий уровень риска, который может быть вызван происхождением этих данных и даже неоднородностью политик контроля доступа, принятых поставщиками сервисов и источников данных. Кроме того, значительное количество используемых источников данных приводит к увеличению времени передачи данных между службами, что, безусловно, повлияет на общее время выполнения и оценку BS. Этот разрыв в оценках отчетливо виден при более низкой плотности, так как в худших случаях он достигал 0,19 (см. рис. 2.10).

Влияние на количество и качество источников данных. Из рис. 2.11 мы также видим, что плотность семейства контекстов напрямую влияет на количество доступных источников данных. Действительно, более плотный формальный контекст означает большее количество общих ис-

точников данных, которые могут использоваться службами-кандидатами. В такой ситуации было бы выгодно объединить сервисы, которые получают доступ к одним и тем же данным, чтобы гарантировать одинаковые политики и уровни безопасности.

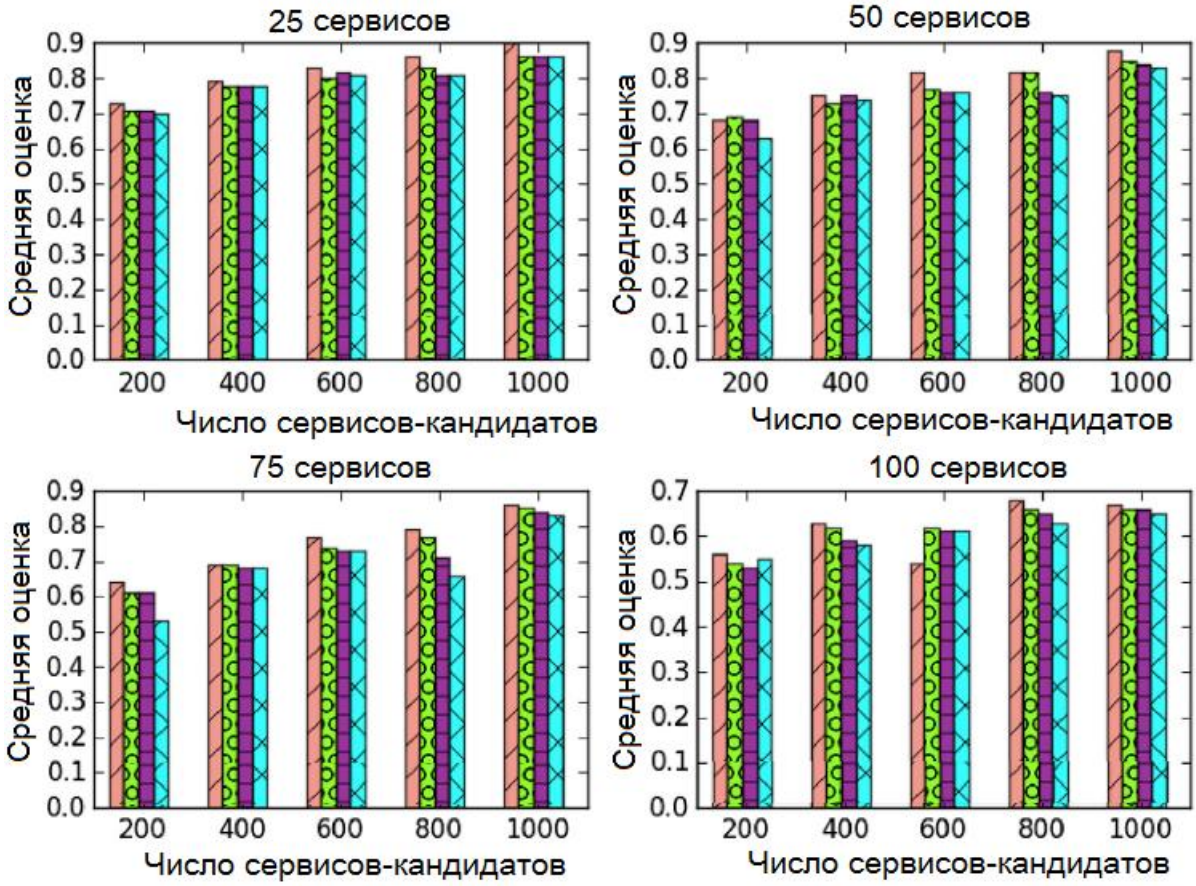


Рис. 2.10. Влияние размера и плотности решеток на оценку BS: - D=50; - D=40; - D=30; - D=40

На рис. 2.12 показано, что количество источников данных для составленной BS уменьшается с увеличением плотности контекста, что напрямую влияет на процент ненадежных источников данных. Например, процент источников данных в оптимальном BSCo составляет около 70,5%, когда плотность достигает 50%, в случае 25 абстрактных сервисов. Однако мы отчетливо замечаем улучшение качества источников данных в случае сложных запросов (100 абстрактных сервисов). На самом деле, процент ненадежных источников данных становится ниже 22,4% в худших случаях

(более низкая плотность формальных контекстов). Это понятно, потому что BS, как правило, формируются из огромного количества задач. Общее количество источников данных с низкой плотностью формальных контекстов может быть равно количеству запрашиваемых сервисов в худшем случае (разные источники данных для каждого сервиса). Фактически, низкая плотность означает, что сервисы используют очень небольшое количество общих источников данных. Следовательно, количество формальных концепций, содержащих сервисы компонентов, вместе с их источниками данных будет стремиться к размеру пользовательского запроса.

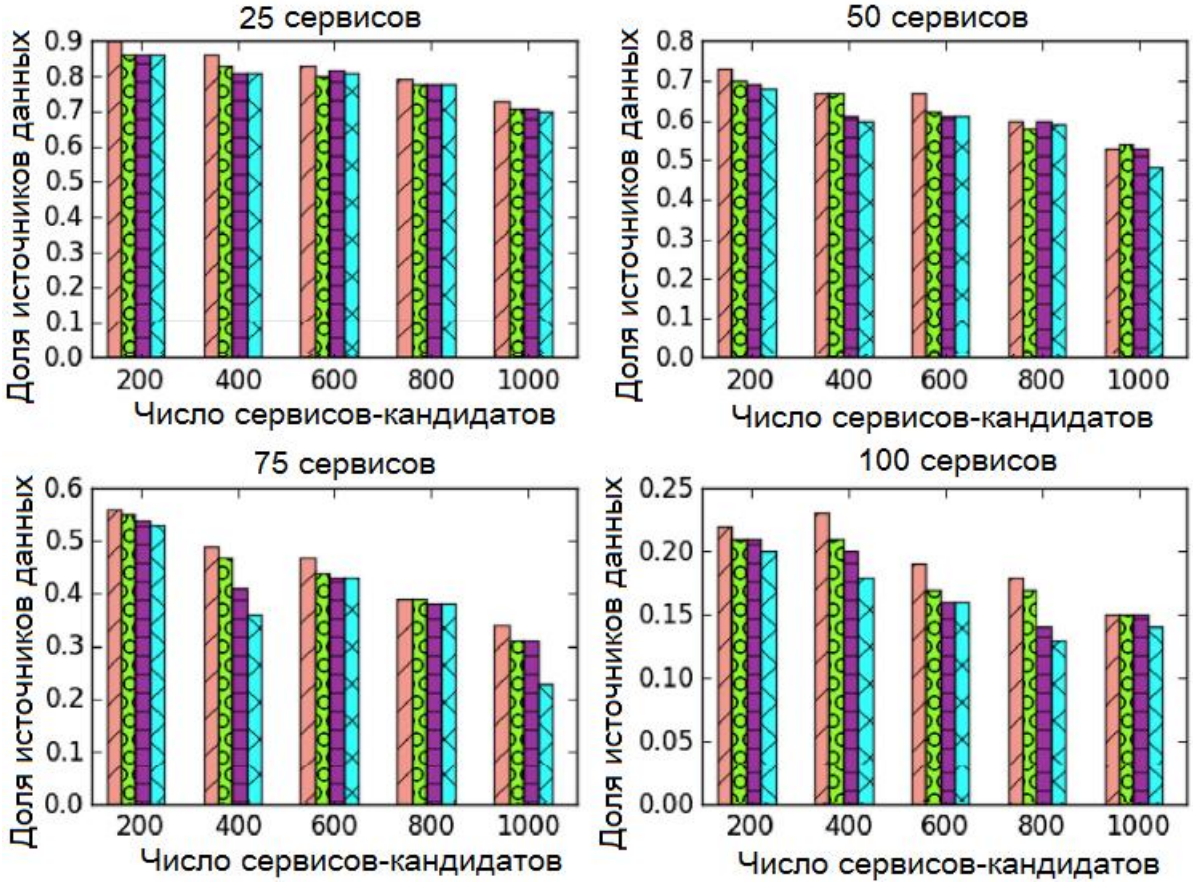


Рис. 2.11. Доля ненадежных источников данных BS для различных запросов пользователей и различных размеров решетки: - D=20; - D=30; - D=40; - D=50

2.4.3. Сравнение с другими подходами

Чтобы сравнить наш подход с CVMIP [2.17] и MREA/G [2.38], мы варьировали количество сервисов-кандидатов от 100 до 1000. Тесты были повторены для разного количества абстрактных сервисов (25, 50, 75 и 100). Среднее время выполнения и оценки BS показаны на рис. 2.12 и 2.13 соответственно. Из рис. 2.12 видно, что для всех тестовых случаев оценки BSCo, полученные с помощью нашего подхода, выше, чем при использовании подходов MR-EA/G и CVMIP. Разница в оценках обусловлена функцией оценки, основанной на штрафных баллах, используемой в нашей работе.

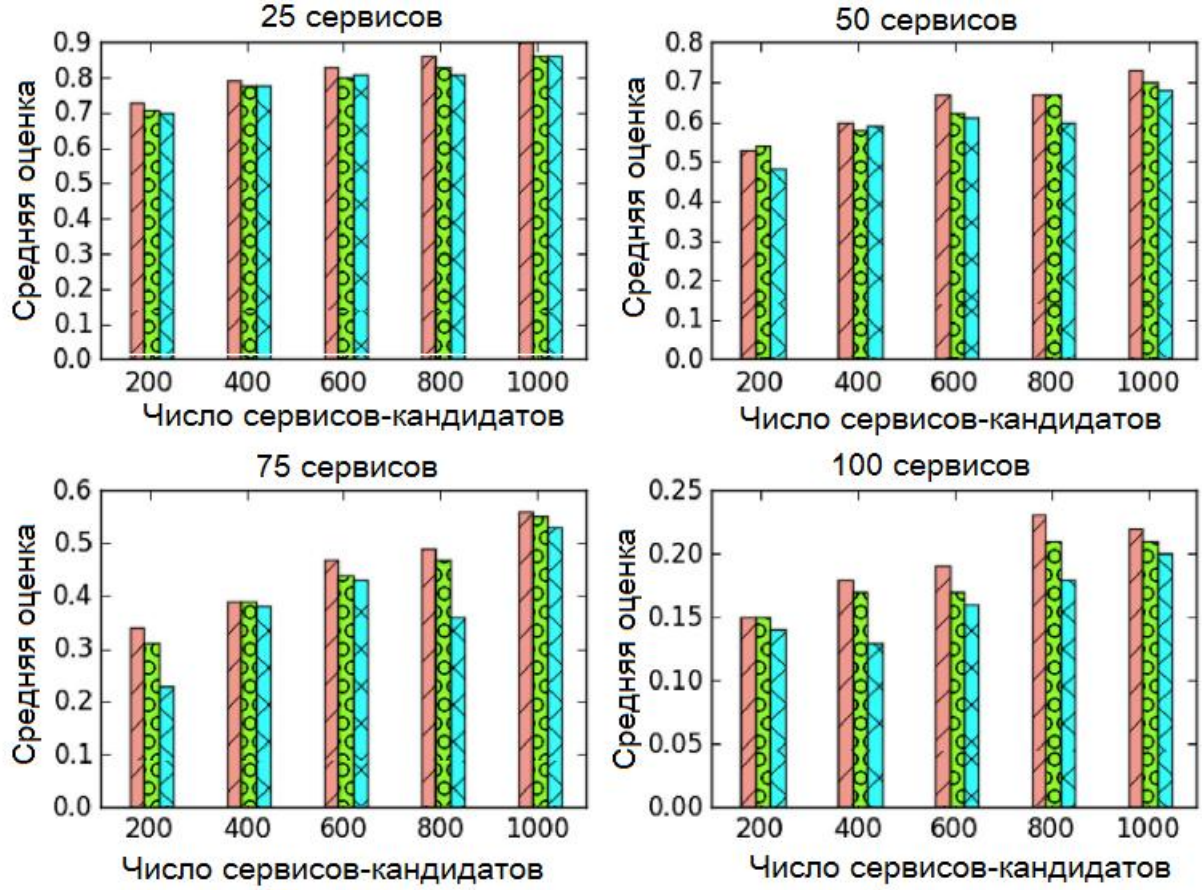


Рис. 2.12. Средние оценки BS за разное количество сервисов для кандидатов: - предложенный подход; - MR-EA/G; - CVMIP

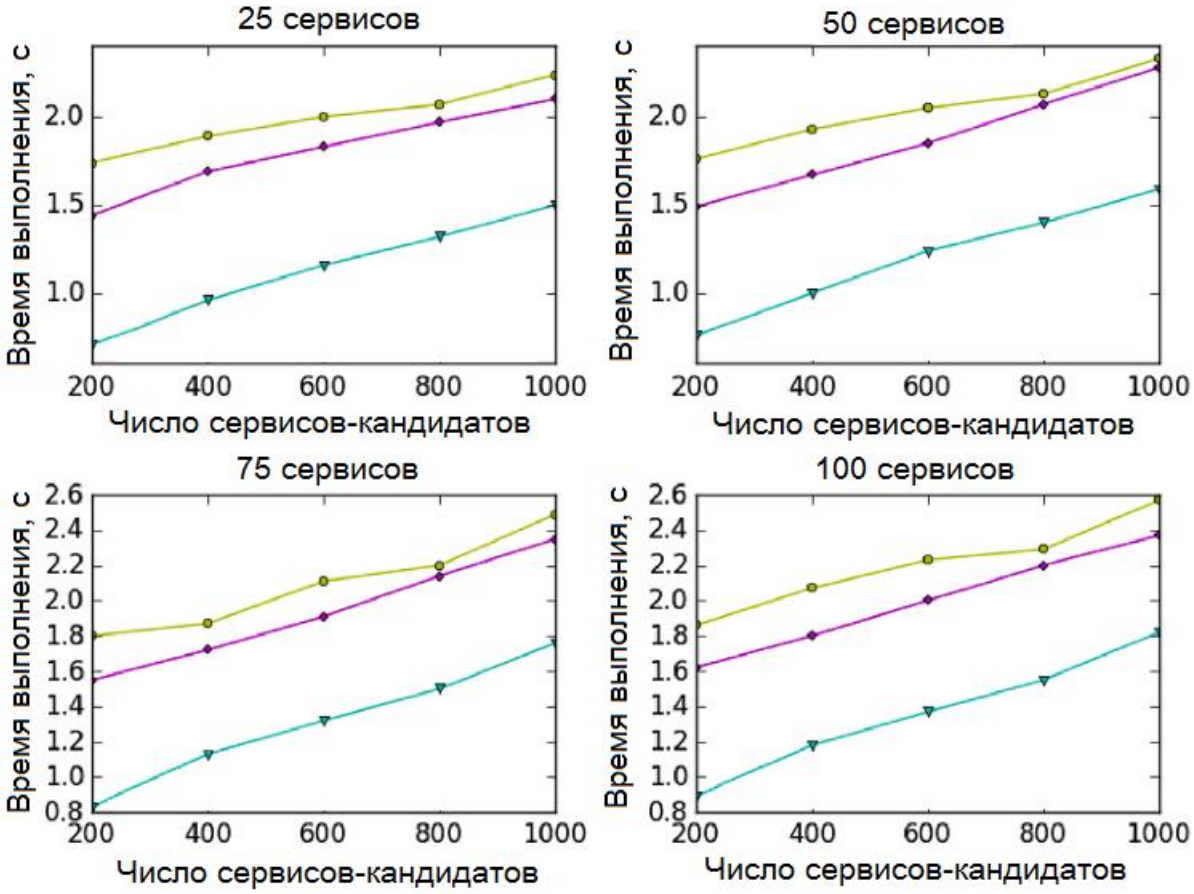


Рис. 2.13. Среднее время выполнения для различного количества сервисов-кандидатов и моделей: - предложенный подход; - MR-EA/G; - CVMIP

Фактически, подсчет оценок осуществляется на двух уровнях. Первый уровень используется при исключении некачественных источников данных из процесса составления, тогда как второй уровень используется при определении оценки составленного BS с помощью значений L-Severity источников данных, участвующих в составлении. Обратите внимание, что этот показатель улучшается в случае высокой плотности семейства формальных контекстов, как мы объяснили на рис. 2.11. На оценки, полученные с помощью MR-EA/G и CVMIP, влияет большое количество источников данных и отсутствие учета уровней безопасности (в нашем случае - L-Severity).

Что касается времени составления, то, как видно из рис. 2.13, наш подход работает лучше, чем MR-EA/G и CVMIP. Низкое время выполнения нашего подхода и MR-EA/G объясняется использованием моделей параллельного программирования, которые позволили разделить пространство поиска и компоновки и выполнять BSCo параллельно.

Наш подход основан на Spark, который преобразует расширенную решетку в набор устойчивых распределенных наборов данных (RDD). Что касается MR-EA/G, то процесс BSCo распараллелен с использованием модели MapReduce, в которой функции Map отвечают за идентификацию сервисов-кандидатов, в то время как функции Reduce вычисляют уровни QoS этих сервисов.

Время компоновки пропорционально количеству сервисов-кандидатов и плотности формальных контекстов (сервисы FC, источники данных FC и привязка FC). Более плотный формальный контекст означает, что сервис использует большое количество источников данных. В такой ситуации этап объединения BS включает в себя разбор значительного количества формальных понятий, что увеличивает время составления всего текста. Более плотный формальный контекст также означает, что службы, участвующие в составлении, могут использовать одни и те же источники данных, что может сократить время, затрачиваемое на обработку набора источников данных для каждой из этих служб. В отличие от более плотных формальных контекстов, низкая плотность означает значительное количество формальных концепций. В этом случае запрос пользователя будет удовлетворен большим количеством разделов экстента, поскольку доступные источники данных будут найдены в отдельных формальных концепциях, что означает, что составленные компоненты BS будут использовать разные источники данных. Следовательно, алгоритм должен проанализировать всю решетку, чтобы извлечь службы-кандидаты, и даже может достичь минимальной формальной концепции.

2.4.4. Эксперименты с иными конфигурациями кластеров

В этой серии тестов мы настроили конфигурацию кластера по умолчанию, чтобы увидеть ее влияние на выполнение заданий BSCo и качество создаваемых композиций.

Влияние распределения на время выполнения. Эта серия тестов включает в себя изменение размера кластера с различным количеством ядер на одного исполнителя. Поскольку решения на основе FCA требуют больших затрат памяти, мы также изучили влияние изменения объема памяти для каждого исполнителя (от 1024 до 4096), что определяет размер кучи в YARN, принимая во внимание накладные расходы на выполнение. Время выполнения различных тестовых заданий показано на рис. 2.14.

На рис. 2.14 показано, что при увеличении размера кластера становится доступно больше ресурсов, что сокращает время выполнения нашего подхода BSCo. Однако мы отмечаем незначительное улучшение времени выполнения подхода MR-EA/G. Это может быть связано с узкими местами в производительности Hadoop/MapReduce. Кроме того, зная, что решения fuzzy RCA являются надежными, но в то же время требуют больших затрат памяти и процессора (из-за большого количества концепций lattice), влияние изменения объема памяти, а также количества ядер на одного исполнителя можно четко увидеть на рис. 2.14. Чем больше ресурсов (памяти и центрального процессора) у исполнителя, тем лучше оптимизируется рабочая нагрузка на выполнение, что значительно увеличивает время BSCo. Например, при использовании 4 ядер центрального процессора и 4 ГБ оперативной памяти время выполнения значительно увеличивается. Это объясняется использованием вычислительных ресурсов нашими алгоритмами синтаксического анализа решеток для фильтрации соответствующих сервисов и источников данных. Однако следует отметить, что на время выполнения могут влиять и другие параметры, такие как изменение конфигураций параллелизма по умолчанию.

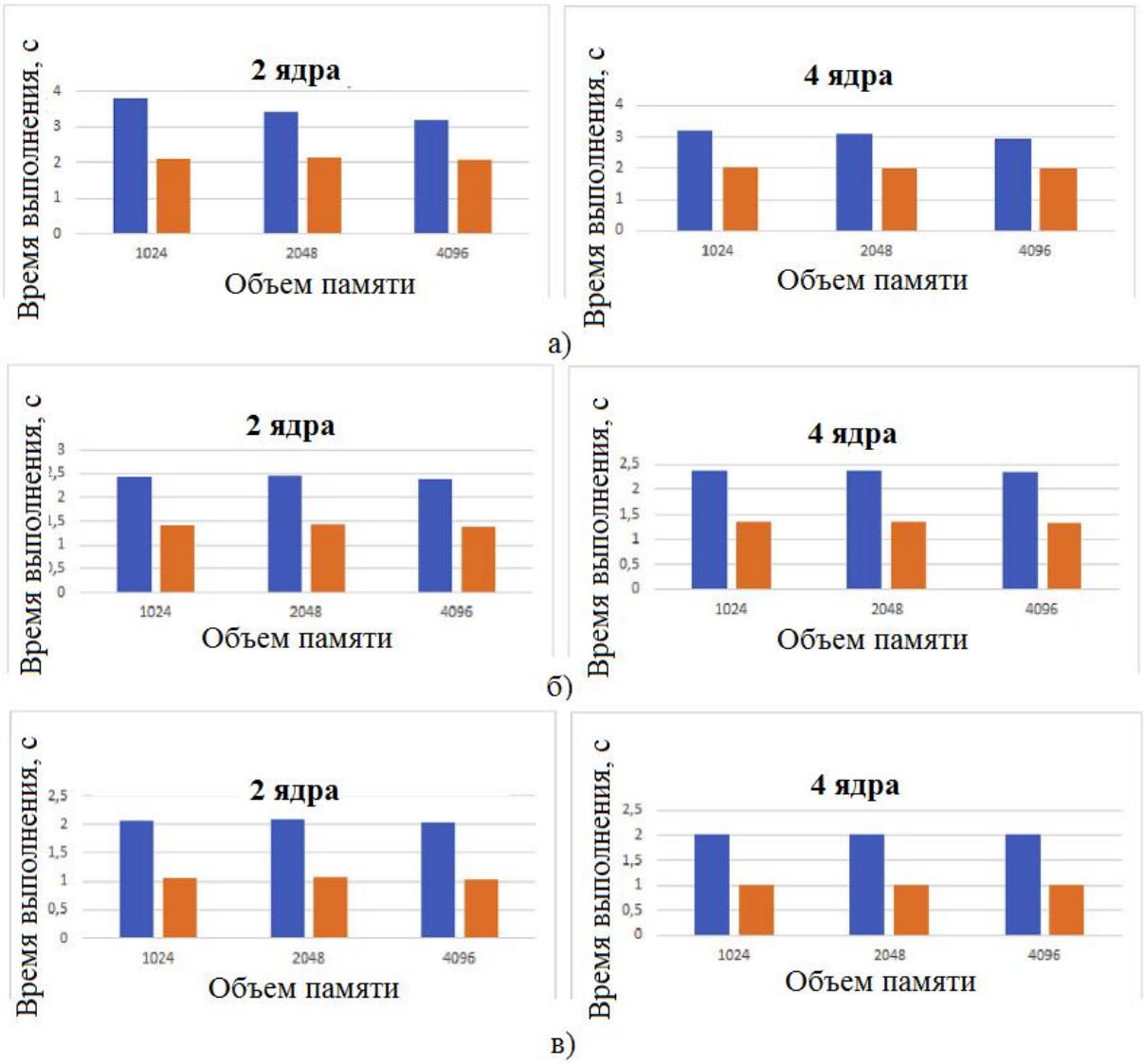


Рис. 2.14. Среднее время работы при изменении количества памяти на одного исполнителя и различных размерах абстрактных сервисов: а) 1 ведущий и 3 ведомых; б) 1 ведущий и 5 ведомых; в) 1 ведущий и 7 ведомых; ■ - MR-EA/G; ■ - наш подход

Влияние распределения на количество потерянных источников данных

Поскольку исходными данными нашего подхода является расширенная решетка, состоящая из нечетких понятий с перекрывающимися отношениями (отношения субпонятие-суперпонятие), целью этой серии тестов

является изучение влияния распределения и секционирования данных на качество BSCo.

Для этого размер кластера и степень разреженности (плотность контекста) были изменены, чтобы измерить уровень потерь при группировании сервисов и источников данных, а также влияние на процент выбранных источников данных для составленной BS (см. рис. 2.15).

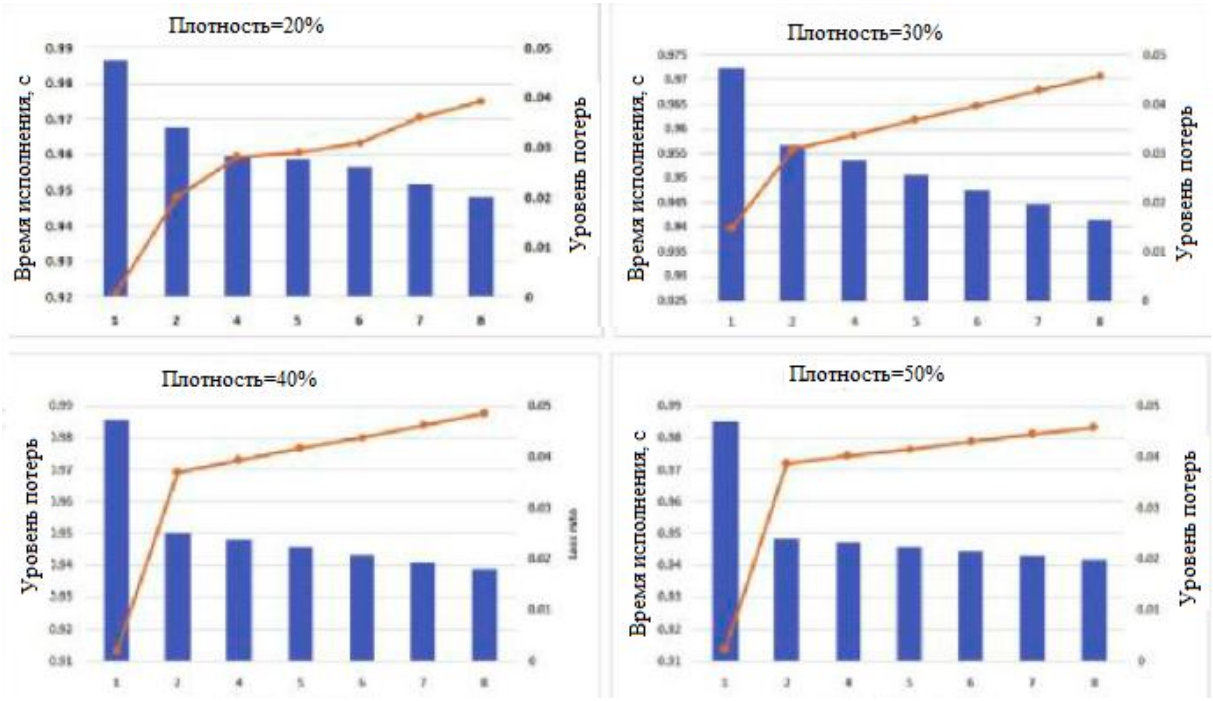


Рис. 2.15. Среднее время работы при изменении количества памяти на одного исполнителя и различных размерах абстрактных сервисов: ▬ - средняя оценка, ●— - уровень потерь

Из рис. 2.15 очевидно, что использование одноузлового кластера позволяет снизить уровень потерь независимо от плотности нечеткого контекста. Однако коэффициент потерь возрастает с увеличением количества виртуальных машин в кластере и достигает 0,0485 в случае 8 виртуальных машин и высокой плотности (плотность = 40%).

На рис. 2.15 показано снижение коэффициента потерь в случае разреженного нечеткого контекста (плотность = 20%). Это понятно из-за уменьшения количества нечетких понятий в этом случае. Напротив, нечет-

кий контекст с высокой плотностью подразумевает большое количество перекрывающихся отношений. Таким образом, разделение расширенной решетки приводит к потере информации о зависимостях и наследовании между концепциями.

На рис. 2.15 также показано прямое влияние уровня потерь на оценку составленной BS и, следовательно, на процент выбранных источников данных. На самом деле, как объясняется в разделе 2.3.2 (см. Алгоритм 2.3), оптимальный BS - это тот, который использует минимальное количество источников данных, удовлетворяя при этом ограничениям QoS, QoD и L-severity. Следовательно, поиск такой комбинации связан с определением набора нечетких понятий, которые группируют требуемые сервисы в их разделах Extent и минимального количества источников данных в их разделах Intent. Это может оказаться неосуществимым, если расширенная решетка секционирована и распределена по узлам кластера. Но, в конечном счете, разница в процентном соотношении источников данных невелика, как мы можем видеть в случаях со 2 и 8 узлами. В худшем случае этот разрыв не превышал 1,93% и 1,15% для значений низкой и высокой плотности (20% и 40%) соответственно. Потери зависимостей между нечеткими понятиями можно избежать, принимая во внимание отношения между понятиями при разбиении на решетки.

2.4.5. Перспективы исследования

Существующий подход может быть улучшен на разных уровнях. В ближайшее время мы проведем дополнительные эксперименты по оценке производительности и качества решения алгоритмов, основанных на нечетком RCA. Мы также применим наш композиционный подход в реальных случаях, таких как IoT BS [2.34, 2.41], производственные сервисы [2.42] и крупные медицинские сервисы [2.15].

Ниже мы кратко описываем будущие улучшения, связанные с про-

блемой BSCo.

- *Совершенствование модели качества BS:* Как упоминалось в разделе 1.2.3, существующие подходы к BS учитывают только QoS, поскольку они игнорируют уровни качества используемых источников данных BS. Модель расширенного качества BS (QoBS) должна включать не только традиционные атрибуты QoS, но и свойства QoD. В рамках нынешнего подхода мы сосредоточились на трех характеристиках качества данных (QoD): полноте, точности и своевременности. Поскольку качество больших данных - это широкое понятие, охватывающее несколько аспектов качества (например, согласованность, доступность, достоверность, репутация, ценность, количество) [2.35], нашей насущной необходимостью является обогащение предлагаемой модели QoBS дополнительными атрибутами QoD и контекстуальными функциями. Улучшенная модель QoBS должна быть достаточно богатой, чтобы охватывать различные домены и функции BS. Мы также намерены улучшить функцию оценки качества данных (см. раздел 2.3.3), включив новые показатели оценки на уровне экземпляра, такие как недостающие данные, нерелевантные или устаревшие данные, ошибки ввода данных, пропущенные данные и т.д. [2.36].

- *Поэтапное управление решеткой BS:* В реальных сценариях BS обеспечивается за счет объединения различных типов (виртуализированных или физических) сервисов (например, транзакционных сервисов, сервисов передачи данных, облачных сервисов и интеллектуальных сервисов Интернета вещей и т.д.), которые потребляют и генерируют огромный объем данных. Возьмем, к примеру, приложения для обработки больших медицинских данных [2.15]. Разнообразие этих сервисов и источников данных приведет к созданию большого решетчатого семейства с перекрывающимися связями. Кроме того, BS, работающие в режиме реального времени и в потоковом режиме, характеризуются высокой степенью изменений, что может значительно увеличить сложность обновления и обра-

ботки структуры BS. Чтобы избежать перестраивания семейства lattice каждый раз, можно применять онлайновые инкрементальные алгоритмы [2.37], такие как AddIntent, Godin и их производные, которые соответствуют эволюции lattice BS и облегчают задачи управления. В этом случае появление новых сервисов/источников данных или обновление существующих запустит поэтапное управление путем идентификации всех измененных концепций в сетке BS и всех канонических генераторов новых концепций.

- *Ориентированный на поток BSCo:* Мы также уделим больше внимания характеристикам big data 5V, главным образом скорости, поскольку они оказывают большое влияние на качество составляемых данных. Фактически, массовое внедрение потоковых сервисов, таких как социальные приложения (например, социальные и бизнес-сети) и городские сервисы (например, сервисы, основанные на местоположении), привело к появлению очень большого объема данных, зависящих от времени. Следовательно, для решения проблем потоковой обработки в BS, возможным решением является учет временных ограничений и скорости передачи данных в процессе BSCo. Основываясь на успешном применении анализа временных нечетких концепций в задачах потоковой обработки данных в умных городах [2.9], в настоящее время разрабатывается реализация, объединяющая временное расширение fuzzy RCA и Spark Framework. В этом расширении хранилище BS рассматривается как временная нечеткая решетка, в которой временные границы (отношения между понятиями) обозначают аспекты эволюции (например, изменения в уровнях QoS и QoD) сервисов и их источников данных.

2.5. Выводы к главе 2

В исследовании мы рассмотрели проблему повторного использования сервисов в эпоху больших данных. BS, рассматриваемые как сложные

экосистемы, создаются и предоставляются путем повторного использования разнородных сервисов (веб, мобильных, облачных, данных и т.д.) из разных доменов в нескольких облачных зонах доступности. Чтобы справиться с проблемами BSCo, в основном с проблемами QoS и безопасности, мы начали с понимания свойств BS и определения модели качества для BS. Предлагаемая модель расширяет традиционную модель QoS веб-сервисов, используя характеристики, связанные с “большими данными” (атрибуты QoD).

На втором этапе мы использовали нечеткое расширение анализа реляционных концепций (нечеткий RCA) для моделирования среды BS в виде семейства решеток. Известный как мощное средство представления данных, кластеризации и анализа, нечеткий RCA использовался для представления взаимосвязей между компонентами BS не только с точки зрения возможностей QoS и QoD, но и в зависимости от их доменов и облачных сред размещения.

Также, учитывая сложный и распределенный характер среды BS, которая рассматривается как большой распределенный контейнер для различных типов сервисов и источников данных, мы внедрили наш подход BSCo поверх хорошо известной платформы Spark big data. Это позволило параллельно обрабатывать информацию о BS из нескольких облаков. Экспериментальные исследования подтвердили способность нашего подхода предоставлять безопасные и высококачественные BS в сжатые сроки.

Итак, разработан 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.

Создан алгоритм компоновки больших сервисов, отличающийся уче-

том 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%.

Литература к главе 2

2.1. Ardagna, D., Cappiello, C., Sam, W., Vitali, M., 2018. Context-aware data quality assessment for big data. Future Generat. Comput. Syst. 89, 548–562.

2.2. Atencia, M., David, J., Euzenat, J., Napoli, A., Vizzini, J., 2020. Link key candidate extraction with relational concept analysis. Discrete Appl. Math., https://doi.org/10.1016/j.dam.2019.02.012.

2.3. Ballou, D.P., Pazer, H.L., 2003. Modeling completeness versus consistency tradeoffs in information decision contexts. IEEE Trans. Knowl. Data Eng. 15 (1), 240–243.

2.4. Barhamgi, M., Benslimane, D., Amghar, Y., Cuppens-Boulahia, N., Cuppens, F., 2013. Privcomp: a privacy-aware data service composition system. EDBT/ICDT 55 (6), 86–97.

2.5. Bertino, E., Ferrari, E., 2018. Big data security and privacy. In: A Comprehensive Guide through the Italian Database Research over the Last 25 Years. Springer, pp. 425–439.

2.6. Boulakbech, M., Messai, N., Sam, Y., Devogele, T., Hammoudeh, M., 2017. IoT mashups: from iot big data to iot big service. In: ICFNDS.

2.7. Cai, L., Zhu, Y., 2015. The challenges of data quality and data quality assessment in the big data era. Data Sci. J. 14, https://doi.org/10.5334/dsj-2015-002.

2.8. De Maio, C., Fenza, G., Gallo, M., Loia, V., Senatore, S., 2014. Formal and relational concept analysis for fuzzy-based automatic semantic annotation. Appl. Intell. 40 (1), 154–177.

2.9. De Maio, C., Fenza, G., Loia, V., Orciuoli, F., 2017. Distributed online temporal fuzzy concept analysis for stream processing in smart cities. J. Parallel Distr. Comput. 110, 31–41.

2.10. Ferchichi, H., Akaichi, J., 2016. Using mapreduce for efficient parallel processing of continuous k nearest neighbors in road networks. J. Software Syst. Dev., https://doi.org/10.5171/2016.356668.

2.11. Gabrel, V., Manouvrier, M., Murat, C., 2015. Web services composition: complexity and models. Discrete Appl. Math. 196, 100–114.

2.12. Gai, K., Qiu, M., Zhao, H., Tao, L., Zong, Z., 2016. Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. 59, 46–54.

2.13. Gai, K., Qiu, M., Zhao, H., 2018. Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distr. Comput. 111, 126–135.

2.14. Gai, K., Xu, K., Lu, Z., Qiu, M., Zhu, L., 2019. Fusion of cognitive wireless networks and edge computing. IEEE Wirel. Commun. 26 (3), 69–75.

2.15. Hao, F., Park, D.-S., Min, S.D., Park, S., 2016. Modeling a big medical data cognitive system with n-ary formal concept analysis. In: Advanced

Multimedia and Ubiquitous Engineering. Springer, pp. 721–727.

2.16. Hazen, B.T., Boone, C.A., Ezell, J.D., Jones-Farmer, L.A., 2014. Data quality for data science, predictive analytics, and big data in supply chain management: an introduction to the problem and suggestions for research and applications. Int. J. Prod. Econ. 154, 72–80.

2.17. Hossain, M.S., Moniruzzaman, M., Muhammad, G., Ghoneim, A., Alamri, A., 2016. Big data-driven service composition using parallel clustered particle swarm optimization in mobile environment. IEEE Trans. Serv. Comput. 9 (5), 806–817.

2.18. Huang, L., Zhao, Q., Li, Y., Wang, S., Sun, L., Chou, W., 2017. Reliable and efficient big service selection. Inf. Syst. Front 19 (6), 1273–1282.

2.19. Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., Nguifo, E.M., 2018. An experimental survey on big data frameworks. Future Generat. Comput. Syst. 86, 546–564.

2.20. Jamil, H.M., Rivero, C.R., 2017. A novel model for distributed big data service composition using stratified functional graph matching. In: Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics. ACM, p. 34.

2.21. Jatoth, C., Gangadharan, G., Fiore, U., Buyya, R., 2018. Qos-aware big service composition using mapreduce based evolutionary algorithm with guided mutation. Future Generat. Comput. Syst. 86, 1008–1018.

2.22. Kathiravelu, P., 2017. Software-defined Inter-cloud Composition of Big Services. EMJD-DC, pp. 1–2.

2.23. Kumar, C.A., Singh, P.K., 2014. Knowledge representation using formal concept analysis: a study on concept generation. In: Global Trends in Intelligent Computing Research and Development. IGI Global, pp. 306–336.

2.24. Kuznetsov, S.O., Makhalova, T., 2018. On interestingness measures of formal concepts. Inf. Sci. 442, 202–219.

2.25. Lahmar, F., Mezni, H., 2020. Security-aware multi-cloud service composition by exploiting rough sets and fuzzy fca. Soft Comput. 1–20.

2.26. Li, D., Wu, J., Deng, Z., Chen, Z., Xu, Y., 2017. Qos-based service selection method for big data service composition. In: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1. IEEE, pp. 436–443.

2.27. Malki, A., Barhamgi, M., Benslimane, S.-M., Benslimane, D., Malki, M., 2014. Composing data services with uncertain semantics. IEEE Trans. Knowl. Data Eng. 27 (4), 936–949.

2.28. Mezni, H., Kbekbi, M., 2019. Reusing process fragments for fast service composition: a clustering-based approach. Enterprise Inf. Syst. 13 (1), 34–62.

2.29. Mezni, H., Sellami, M., 2017. Multi-cloud service composition using formal concept analysis. J. Syst. Software 134, 138–152.

2.30. Mezni, H., Sellami, M., 2018. A negotiation-based service selection approach using swarm intelligence and kernel density estimation. Software Pract. Ex. 48 (6), 1285–1311.

2.31. Rouane-Hacene, M., Huchard, M., Napoli, A., Valtchev, P., 2013. Relational concept analysis: mining concept lattices from multi-relational data. Ann. Math. Artif. Intell. 67 (1), 81–108.

2.32. Sellami, M., Hacid, M.-S., Gammoudi, M.M., 2018. A fca framework for inference control in data integration systems. Distributed Parallel Databases 1–44.

2.33. Shehu U., Safdar G., Epiphaniou G., 2015. Towards network-aware composition of big data services in the cloud, Int. J. Adv. Comput. Sci. Appl. 6 (10).

2.34. Taherkordi, A., Eliassen, F., Horn, G., 2017. From iot big data to iot big services. In: Proceedings of the Symposium on Applied Computing. ACM, pp. 485–491.

2.35. Taleb, I., Dssouli, R., Serhani, M.A., 2015. Big data pre-processing: a quality framework. In: 2015 IEEE International Congress on Big Data. IEEE, pp. 191–198.

2.36. Taleb, I., El Kassabi, H.T., Serhani, M.A., Dssouli, R., Bouhaddioui, C., 2016. Big data quality: a quality dimensions evaluation. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld). IEEE, pp. 759–765.

2.37. Valtchev, P., Missaoui, R., Godin, R., 2004. Formal concept analysis for knowledge discovery and data mining: the new challenges. In: International Conference on Formal Concept Analysis. Springer, pp. 352–371.

2.38. Vavilis, S., Petkovi, M., Zannone, N., 2014. Data leakage quantification. In: IFIP Annual Conference on Data and Applications Security and Privacy. Springer, pp. 98–113.

2.39. Vavilis, S., Petkovi, M., Zannone, N., 2016. A severity-based quantification of data leakages in database systems. J. Comput. Secur. 24 (3), 321–345.

2.40. Wang, R.Y., Strong, D.M., 1996. Beyond accuracy: what data quality means to data consumers. J. Manag. Inf. Syst. 12 (4), 5–33.

2.41. Wang, X., Yang, L.T., Feng, J., Chen, X., Deen, M.J., 2016. A tensor-based big service framework for enhanced living environments. IEEE Cloud Comput. 3 (6), 36–43.

2.42. Wei, L., Zhao, Q., Shu, H., 2018. Design of manufacturing big data access platform based on soa. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC). IEEE, pp. 1841–1845.

2.43. Xu, X., Sheng, Q.Z., Zhang, L.-J., Fan, Y., Dustdar, S., 2015. From big data to big service. Computer (7), 80–83.

2.44. Xu, X., Motta, G., Wang, X., Tu, Z., Xu, H., 2018. A new paradigm of software service engineering in the era of big data and big service. Computing 100, 353–368.

2.45. Zheng, Z., Zhang, Y., Lyu, M.R., 2010. Distributed QoS evaluation for real-world web services. In: 2010 IEEE International Conference on Web Services. IEEE, pp. 83–90.

2.46. Zhou, L., Chen, H., Yu, T., Ma, J., Wu, Z., 2008. Ontology-based scientific data service composition: a query rewriting-based approach. In: AAAI Spring Symposium: Semantic Scientific Knowledge Integration, pp. 116–121.

3. Управление распределением больших данных интернета вещей

Являясь новой, востребованной технологией в стране и за рубежом, Интернет вещей сочетает в себе характеристики данных с преимуществами управления распределенными базами данных в режиме реального времени, а стратегия распределения данных как ключевая технология схемы хранения данных находится в центре внимания исследований. В соответствии с массовостью, пространственно–временной корреляцией, дисбалансом доступа и постоянной изменчивостью информации в Интернете вещей, для адаптации к ней необходима модель распределения данных во временной области, чтобы разработать стратегию динамического распределения данных, основанную на адаптивной обратной связи по нагрузке во временной области. В соответствии с характеристиками данных задача распределения статических данных сводится к простой задаче линейного программирования, а для обратной связи с информацией о нагрузке используется адаптивная временная область. Наконец, для реализации динамического распределения данных настраивается функция порога динамической нагрузки. Для моделирования распределения глобальных скалярных данных используется целочисленное линейное программирование и предлагается алгоритм распределения глобальных данных (GDP), основанный на алгоритме RODP. Алгоритм GDP может быстро решить проблему распределения скалярных данных во всей программе за полиномиальное время. Наконец, численные эксперименты в основной части программного цикла показывают, что предложенная стратегия обладает лучшей производительностью с точки зрения балансировки нагрузки в краткосрочной области и миграции системных данных, чем аналогичные алгоритмы. Имитационные эксперименты проводились на двух наборах тестовых программ соответственно. Результаты экспериментов показывают, что глобальный алгоритм

распределения данных и итеративный алгоритм оптимального распределения данных, предложенные в работе, превосходят алгоритм распределения данных на основе жадной стратегии с точки зрения задержки доступа и энергопотребления для всех тестовых программ.

3.1. Интернет вещей как компонент технологии больших данных

Интернет вещей стал популярной технологией в области академических исследований и промышленного производства и в будущем будет играть важную роль в новом мире Интернета. Интернет вещей использует различные сенсорные устройства для связи человека с другими людьми и объектами в любое время и в любом месте в сфере деятельности через любую сеть или сервис, предоставляя масштабные межотраслевые услуги передачи данных. Сегодня Интернет вещей широко используется в транспорте, здравоохранении и общественных службах. В обширной и разнообразной сетевой среде Интернета вещей надежность распределения данных становится основой информационной службы Интернета вещей, а оптимизация энергоэффективности играет важную роль в содействии распределению данных в Интернете вещей с ограниченным потреблением энергии. Спустя немногим более десяти лет после того, как был предложен Интернет вещей в качестве новой технологической точки доступа, он вызвал широкий интерес исследователей и представителей промышленности по всему миру.

Что касается системы Интернета вещей, то исследования в стране и за рубежом в основном сосредоточены на ‘вещах’ или ‘сетях’. Исследования, ориентированные на ‘вещи’, проводятся вокруг датчиков, а исследования, ориентированные на ‘сеть’, - вокруг обработки информации. В области обработки информации в Интернете вещей способы хранения, анализа и обработки собранной с датчиков информации являются одной из ключевых технологий [3.2, 3.12] Информационная система должна сохра-

нять информацию с датчиков, получать значение в реальном времени или историческое значение требуемой информации с датчиков, а затем получать тренд изменения реальной физической величины, отражаемой датчиком, что закладывает основу для интеллектуального анализа в реальном времени, интеллектуального восприятия объектов и поиска и интеллектуального анализа данных на основе объектов. Сенсорная информация Интернета вещей обладает следующими характеристиками [3.4, 2,32].

1. Данные сенсоров Интернета вещей являются типичным представителем больших данных. Датчики Интернета вещей будут генерировать большое количество сенсорной информации, описывающей физическое тело, и эти данные будут непрерывно поступать в центр обработки данных, формируя огромный поток данных. Объем информации отражается в объеме информации в реальном времени и объеме исторической информации [3.5, 3.6].

2. Временная и пространственная корреляция данных датчиков. По сравнению с традиционными интернет-данными, сенсорные данные Интернета вещей, как правило, обладают пространственно-временными характеристиками физического мира. Перерыв в анализе и хранении информации приведет к искажению отраженной физической величины, поэтому информация с датчиков будет передаваться в режиме реального времени. В то же время датчик существует в реальном физическом мире, и данные датчика будут агрегироваться под влиянием пространственного положения датчика, то есть датчика, расположенного рядом с пространством [3.14, 3.24], и тенденция изменения его данных похожа.

3. Несбалансированность доступа к данным датчиков. В ходе реального производственного цикла данные массового датчика изменяются относительно конкретного значения данных с точки зрения их практической значимости, поэтому фактические операции запроса, как правило, связаны со статистической информацией и сложной логикой [3.3, 3.18], и общая

статистика часто может быть получена с помощью каждого из них.

Узел распределенных вычислений, не требующий больших затрат, становится узким местом в системе ресурсов. Операция обновления данных выполняется для каждой массивной точки данных. Чтобы соответствовать требованиям реального времени, узлы хранения должны постоянно принимать информацию о массовом обновлении данных [3.10, 3.28], что часто требует большого объема памяти и ресурсов ввода-вывода. Большинство сенсорных данных, собираемых Интернетом вещей, включают в себя постоянные изменения температуры, влажности, давления, географического положения и других физических величин. Таким образом, постоянные изменения физических величин приводят к постоянным изменениям в процессе обновления данных. Можно сделать вывод о диапазоне тенденций изменения данных в следующем временном интервале по изменениям в обновлении данных в текущем временном интервале [3.16, 3.21]. Упомянутая выше природа сенсорных данных в Интернете вещей создает серьезные проблемы при хранении информации и управлении ею. В [3.8, 3.10] отмечается, что хранение информации в Интернете вещей должно быть централизованным с использованием распределенной базы данных, а стратегия распределения данных является одной из ключевых технологий распределенного хранения данных [3.19, 3.26]. Предыдущие исследования, посвященные хранению больших данных в Интернете вещей, в основном были посвящены описанию целей и требований к информационным центрам. В [3.9, 3.11] основное внимание уделялось целям и требованиям к хранению информации в Интернете вещей, но не было предложено практических и осуществимых схем хранения данных и стратегий распространения данных. В большинстве существующих исследований, посвященных стратегиям распределения данных, были предложены решения для общего хранения данных. В [3.13, 3.27] изучается общая проблема обработки данных на основе фреймворка или алгоритма, не проводя соответствующей

оптимизации для характеристик данных Интернета вещей. Другой вид мобильного Интернета вещей использует подвижные распределительные узлы для перемещения по сетевому пространству и сбора информации, генерируемой узлами данных. В большинстве сценариев применения Интернета вещей расстояние между узлами обычно превышает максимальный радиус связи между узлами, поэтому однократная передача данных часто невозможна [3.13, 3.23]. Хотя многоступенчатая передача данных делает возможным крупномасштабное статическое распределение данных Интернета вещей, ‘эффект воронки’ или ‘эффект горячей точки’ возникает только при использовании многоступенчатого режима для агрегации данных узлов, что приводит к снижению производительности распределения данных и энергоэффективности [3.20, 3.22].

Чтобы эффективно решить эту проблему, в соответствии с требованиями времени появилась стратегия распределения данных с помощью мобильных распределительных узлов, которая, как было доказано, эффективно оптимизирует энергетические показатели Интернета вещей или WSN, что называется мобильным распределением данных [3.1, 3.26]. В соответствии с характеристиками траекторий в назначенных узлах существующие исследования можно разделить на две категории: траектории со свободной мобильностью и траектории с ограниченной мобильностью. В [3.17, 3.30, 3.31] также изучался метод беспроводной передачи энергии с использованием мобильного распределительного узла в качестве узла передачи радиочастотной энергии. Был рассмотрен мобильный распределительный узел с полной мобильностью, и было проведено математическое моделирование для планирования маршрута, контроля скорости и других аспектов. Совместная оптимизация была проведена в соответствии с методом случайного развертывания узла, и была разработана разумная стратегия беспроводной передачи энергии. Стратегия минимизирует длину пути перемещения узла, агрегацию данных и время беспроводной передачи

энергии в качестве целевой функции и оптимизирует задачу комбинаторной оптимизации с помощью алгоритма имитационного отжига.

В существующих исследованиях для реализации беспроводных систем ретрансляции обычно используется метод SWIPT для одновременной передачи беспроводной информации и энергии. Однако метод SWIPT в беспроводных системах ретрансляции не может быть непосредственно применен для оптимизации энергоэффективности агрегирования данных в Интернете вещей. В существующей литературе недостаточно исследований о влиянии метода SWIPT на энергоэффективность крупномасштабного Интернета вещей или WSN. Метод SWIPT может передавать избыточную энергию от узлов к другим узлам при передаче информации. Однако технология радиочастотной передачи энергии обеспечивает низкую долю сбора энергии, поэтому необходимо интегрировать избыточные энергетические ресурсы большого числа сетевых узлов с помощью эффективных стратегий передачи энергии, повысить энергоэффективность узлов в горячих точках и оптимизировать общий показатель энергоэффективности сети.

3.2. Характеристики данных датчиков Интернета вещей

Предложена разновидность динамических данных, основанных на адаптивной схеме обратной связи по нагрузке во временной области (ATDA). Стратегия ATDA IoT sensor направлена на устранение дисбаланса между постоянной изменчивостью и доступом к информации, предлагая адаптивную стратегию обратной связи с нагрузкой во временной области, стратегию динамической обратной связи с людьми во временной области, позволяющую в режиме реального времени контролировать влияние изменения нагрузки на систему. Стратегия ATDA предлагает своего рода статическую схему распределения данных для гетерогенных физических узлов, при этом информационное пространство датчиков рассматривается в

синтезированном решении. По сравнению с алгоритмами Random и Bubbaa, эта стратегия лучше подходит для балансировки нагрузки в краткосрочной области и имеет самые низкие затраты на связь в полной области. Математическое моделирование скалярного распределения данных осуществляется на основе многослойной сети; предложен алгоритм сегментного распределения данных на основе динамического программирования, который позволяет получить оптимальное решение при полиномиальной временной сложности. Затем предлагается модель целочисленного линейного программирования для решения задачи глобального распределения данных и эвристический алгоритм глобального распределения данных. Далее предлагается модель целочисленного линейного программирования для решения глобальной задачи распределения данных.

3.3. Архитектура и стратегия распределения данных для датчиков Интернета вещей

3.3.1. Архитектура распределения данных

Для исследования распределения данных датчиков по хранилищам в Интернете вещей необходимо понять соответствующую структуру хранения и разработать более разумную схему распределения данных на этой основе. Традиционный метод обработки данных датчиков в Интернете вещей, как правило, использует метод распределенного хранения. Схема хранения данных представляет собой распределенную схему хранения баз данных реального времени, которая сочетает в себе технологию облачных вычислений для хранения данных с ‘подбором ключей’ и параллельную технологию распределенной базы данных. Схема хранения, основанная на ‘key gas hundred’, может гарантировать обновление данных в режиме реального времени, в то время как платформа параллельного распределенного хранилища может гарантировать хранение больших объемов данных.

Структура распределенной системы баз данных реального времени

состоит из четырех частей: узла управления, клиентского интерфейса узла хранения и порта сбора данных. Каждая часть подключена к высокоскоростной сети, и на каждом узле хранения работает база данных в режиме реального времени. При традиционной стратегии распределения данных точки данных распределяются случайным и статическим образом, а узлы хранения возвращают свою загрузку в течение фиксированного периода. Такая стратегия не учитывает характеристики сенсорных данных Интернета вещей, и легко назначить точки, относящиеся к пространственной информации, одному и тому же узлу хранения при статическом распределении. В процессе обратной связи по нагрузке изменение данных не может быть передано динамически, что приведет к неравномерной загрузке в течение короткого промежутка времени и повлияет на производительность обновления данных в режиме реального времени.

На рис. 3.1 показана структура системы динамического распределения данных, принятая политикой ATDA. 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.

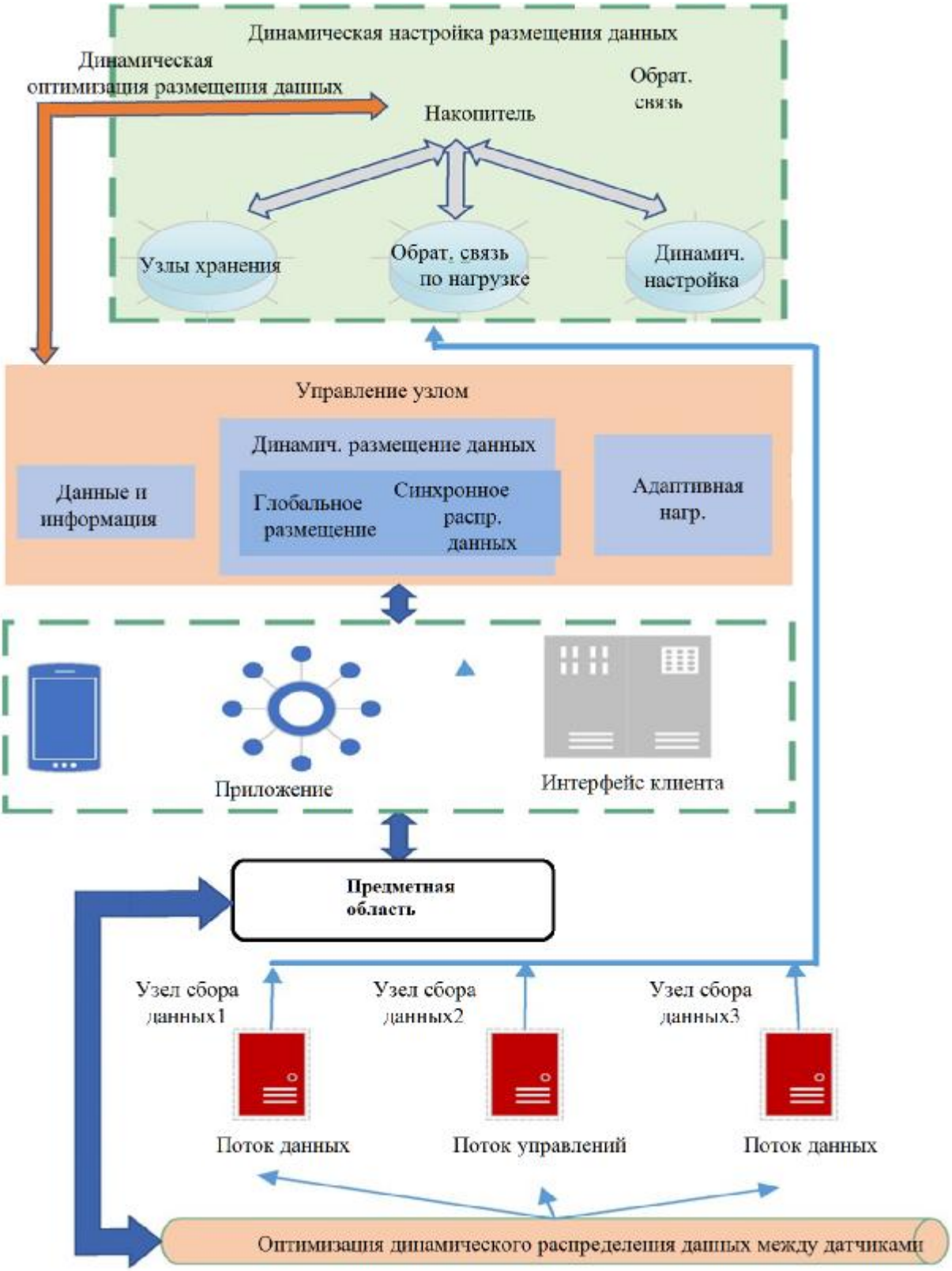


Рис. 3.1. Структурная схема динамической системы распределения данных

3.3.2. Стратегия распределения данных

Схема динамического распределения данных основана на модели

распределения данных для сенсорной информации Интернета вещей. Во-первых, распределение данных абстрагируется в математическую модель. Во-вторых, предложен алгоритм статического распределения, который расширяет математическую модель распределения данных во временной области и оптимизирует пространственную корреляцию для набора сенсорной информации Интернета вещей в соответствии с предложенной гипотезой и математической моделью сокращения предварительной информации гетерогенных узлов хранения. Затем предлагается адаптивный алгоритм обратной связи во временной области, который использует непрерывное изменение информации датчиков для определения следующего цикла обратной связи по нагрузке на основе текущего состояния нагрузки и соответствующим образом сохраняет информацию о нагрузке узла. Наконец, предложен алгоритм динамического распределения, который находит перегруженные узлы хранения в соответствии с эвристической функцией порога динамической нагрузки и регулирует нагрузку на перегруженные узлы хранения с помощью жадной идеи.

Распределение данных является важной частью процесса оптимизации компилятора. Он определяет, какие данные хранятся в каких регистрах и какие данные хранятся в памяти, путем анализа данных, используемых в программе, и доступных ресурсов системы, чтобы обеспечить бесперебойную работу программы при ограниченных ресурсах хранилища.

Для разных целей оптимизации требуется различная схема распределения данных, например, прерывание, когда более встроенное приложение хочет использовать как можно меньше регистров, чтобы уменьшить накладные расходы: прерывания для защиты и восстановления производительности, оптимизация приоритета, чтобы полностью использовать все доступные регистры, чтобы как можно меньше использовать регистры, получить доступ к памяти, повысить эффективность выполнения. Ориентируясь на два показателя в модели оценки экологичности, а именно на ин-

декс энергопотребления и равновесном индексе использования ресурсов, различные методы оптимизации распределения данных будут оказывать следующее влияние на индекс экологичности системы.

1. Различные схемы распределения данных будут создавать различные последовательности команд, а различные последовательности выполнения команд будут влиять на частоту переключения и степень балансировки шины данных команд и, таким образом, на энергопотребление шины и нагрузку на отдельную шину. Как настроить схему распределения данных, уменьшить частоту переключения шины, сбалансировать нагрузку на каждую шину, улучшить индекс шины - эту схему распределения данных нельзя игнорировать.

2. Из-за различных режимов доступа к данным в программе, различных схем распределения данных частота доступа и энергопотребление различных устройств хранения данных будут сильно различаться. Даже для одной и той же операции считывания данных модуль хранения с более высокой частотой доступа потребует более высокого энергопотребления в процессе считывания из-за влияния температурного фактора. Следовательно, как разумно настроить доступ к блоку хранения в соответствии с конкретным режимом доступа к данным, чтобы сбалансировать частоту доступа к каждому блоку хранения и снизить дополнительное потребление энергии и потери оборудования, вызванные чрезмерно интенсивным доступом к блоку хранения. Выразим модель оценки индекса в процессе распределения данных следующим образом:

$$Q=d(W_{BUS}+W_M)+g(S_{BUS}+S_M) \quad (3.1)$$

Здесь W_{BUS} представляет индекс энергопотребления шины, и модель энергопотребления шины может быть использована для расчета в соответствии с режимом переключения шины. W_M представляет собой показатель энергопотребления накопителя. Поскольку динамическое энергопотребление накопителя в основном обусловлено операциями чтения и записи, чем

чаще выполняются операции чтения и записи, тем выше будет потребление энергии.

Узел управления получает информацию о загрузке узлов хранения в режиме реального времени, поэтому ему необходимо динамически корректировать распределение данных по каждому узлу хранения, чтобы сбалансировать нагрузку. Динамическое распределение данных состоит из двух частей: определение того, перегружен ли узел хранения; регулировка нагрузки на перегруженный узел хранения. Чтобы определить, перегружен ли узел хранения, необходимо установить пороговое значение перегрузки. Стоимость связи, вызванная более сбалансированной нагрузкой, выше, в то время как стоимость связи, вызванная относительно несбалансированной нагрузкой, ниже. Однако установка порога перегрузки узла хранения требует балансировки нагрузки и затрат на связь.

Основная идея алгоритма для установки порога перегрузки узлов хранения 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:

$$K_M = \begin{cases} b + k\ln(e + e(K_{mean} - a)) \\ b + K_{mean} \end{cases} \qquad (3.2)$$

K_M - порог перегрузки узлов хранения, K_{mean} - средняя скорость загрузки каждого узла хранения. Критическим считается порог серьезной перегрузки, когда уровень загрузки памяти превышает 75%, это серьезно влияет на производительность, поэтому обычно он составляет от 0,6 до 0,8.

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.

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

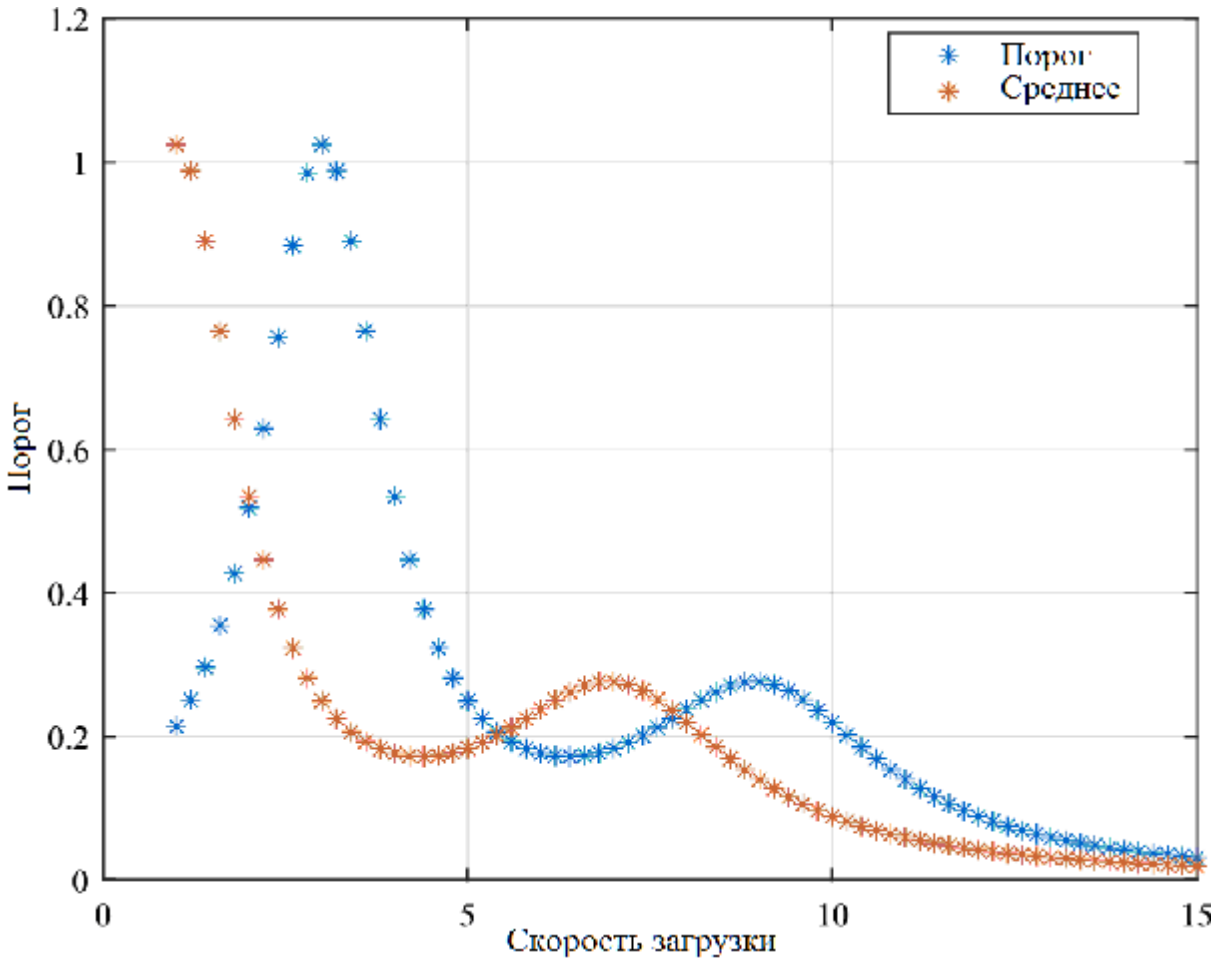


Рис. 3.2. Диаграмма порога нагрузки

Ключом к регулированию нагрузки на перегруженный узел хранения является сокращение миграции данных. В системе стоимость миграции каждой 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 алгоритма приведена на рис. 3.3 и в табл. 3.1.

Алгоритм Dynamic использует жадное правило для передачи данных узлу с малой загрузкой.

В любом итерационном процессе алгоритма, описанного в этой главе, количество операций, необходимых для алгоритма ручного вмешательства и алгоритма обновления популяции, больше, чем для предыдущих операций, таких как пересечение и мутация. Таким образом, временная сложность вмешательства человека и обновления популяции может быть использована для отражения сложности всего алгоритма. Каждое вмешательство человека состоит из трех этапов поиска с временной сложностью около $O(n^8)$.

В процессе обновления популяции алгоритм пузырьковой сортировки используется для сортировки индивидуального адаптивного значения и неадаптивного значения, а временная сложность в худшем случае равна $O(n_{ga}^2)$. Таким образом, уровень временной сложности эвристического алгоритма, разработанного в этой главе, равен уровню $O(N^2)$, что лучше, чем уровень $O(N^3)$.

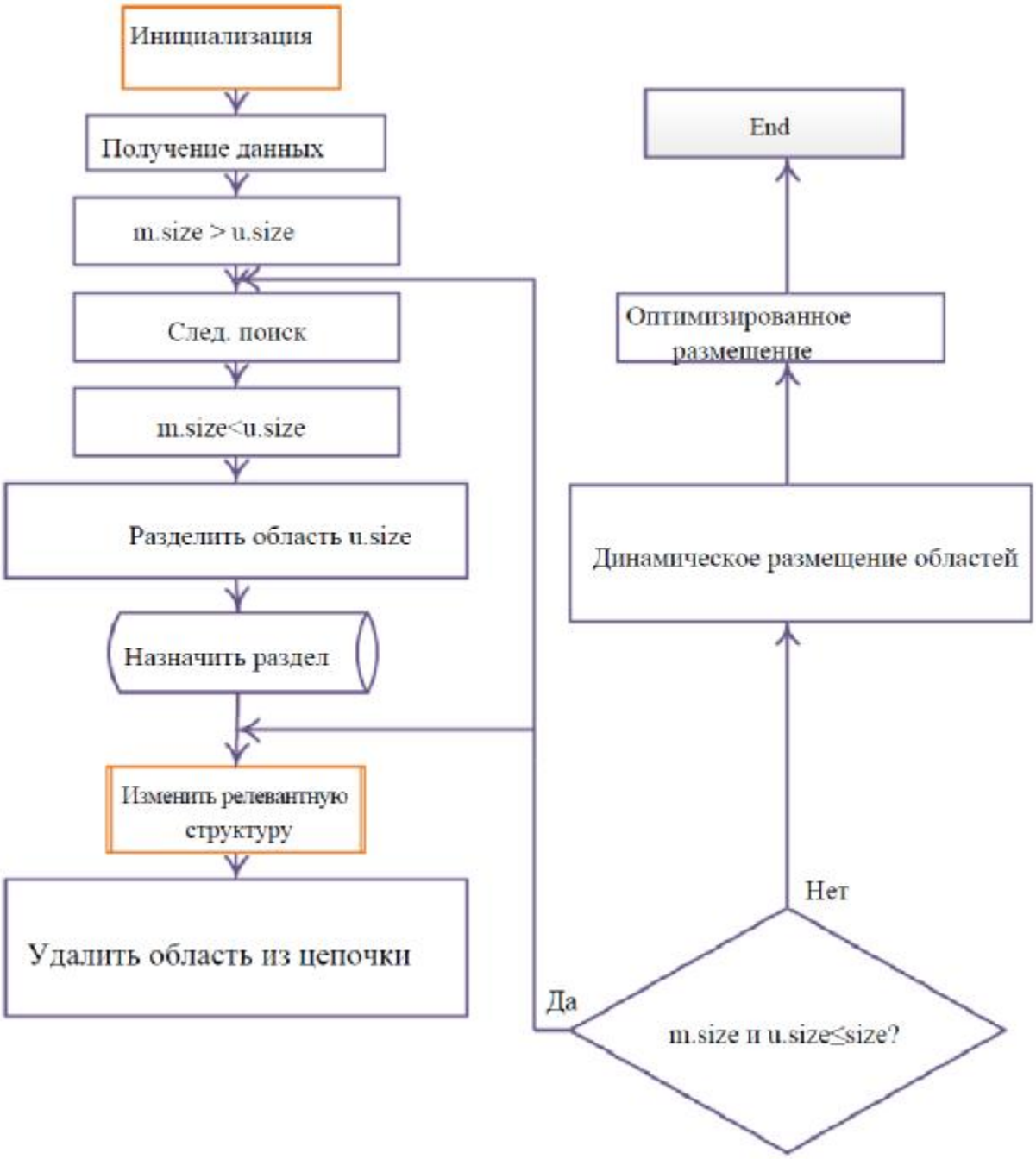


Рис. 3.3. Структурная схема алгоритма Dynamic динамического распределения данных

3.4. Разработка алгоритма оптимизации распределения больших данных для датчиков в Интернете вещей

Для задачи размещения данных во встроенной системе многоуровневого хранения на основе SPM существующая жадная стратегия не смогла получить оптимальную схему размещения данных. На основе метода ди-

намического программирования разработан алгоритм оптимального распределения данных по разделам с полиномиальной временной сложностью.

Таблица 3.1

(1) Алгоритм динамического распределения данных

BEGIN
Init $Node_i$; /* Инициализация загрузки для узлов хранения с 1 по N */
While True
Wait $Node_i$ /* Ожидание поступления информации о загрузке каждого узла */
/*Узел возвращает информацию о нагрузке, обновляет значение нагрузки L; */
 	
If $K>K_{mean}$ /* превышает ли скорость загрузки пороговое значение */
Call the Dynamic (Node) ;
End If
Возвращаем обновленный K_{mean}
End While
END

Таблица 3.2

(2) Алгоритм Dynamic

BEGIN
While $K>K_{mean}$
Выбрать (точка данных Ta_i), чтобы найти наиболее часто обновляемую точку данных $Node_i$
Удалить $Node_i$ из хранилища Node;
Выбрать узел хранения $Node_i$ с наименьшей нагрузкой из узлов хранения;
Добавить точку данных Ta_i в хранилище $Node_i$;
Пересчитать загрузку $Node_i$;
End While
END

Для моделирования проблемы глобального распределения данных используется модель целочисленного линейного программирования (ILP), и предлагается эвристический алгоритм глобального распределения данных, основанный на оптимальном результате распределения данных по

сегментам программы, что еще больше снижает накладные расходы на доступ ко всей программе. Предположим, что программа состоит из двух программных сегментов, и количество обращений к данным в каждом сегменте показано в табл. 3.3.

Табл. 3.3 показывает три различные схемы распределения данных и общую нагрузку на два сегмента программы. Можно видеть, что три схемы распределения в табл. 3.3, используют оптимальную схему выделения из примера в предыдущем разделе, в то время как распределение данных в блок 2 то же. Различные схемы распределения данных, влияют на общую нагрузку на доступ к конечной программе. Это связано с тем, что схема распределения данных для уровня 1 будет использоваться в качестве первоначального распределения для уровня 2, что влияет на общую нагрузку на доступ к программе. Для всей программы ‘распределение 2’ и ‘распределение 3’ обеспечивают более эффективное распределение данных. Таким образом, выбор оптимального распределения данных также влияет на общие затраты на выполнение всей программы.

Таблица 3.3

Время доступа к данным, распределяемое по глобальным данным

Данные	Блок 1	Блок 2
V1	11	4
V2	6	7
V3	8	7
V4	5	3
V5	5	4
V6	6	8

Затем, учитывая режим доступа к регистру (т.е. частоту доступа к самому регистру и частоту доступа к регистру, близкую к частоте доступа к регистру), в некоторых случаях перестановка также может привести к замене набора инструкций класса процедурами перераспределения регистров, что не гарантирует увеличения данных о переполнении с одной сто-

роны, при условии максимально сбалансированного доступа к каждому регистру, с другой стороны, команда между переключениями по шине выполняется как можно реже и сбалансировано.

3.5. Эксперимент

Эксперимент основан на распределенной системе баз данных реального времени, в которой узлы управления, узлы хранения и станции сбора данных расположены на разных физических устройствах соответственно. Реальные данные с датчиков используются в качестве тестовых образцов, чтобы максимально соответствовать процессу распространения данных в реальной рабочей среде. Экспериментальными алгоритмами сравнения были алгоритм Bubba и алгоритм Random.

Экспериментальные данные взяты из реальных образцов данных датчиков системы сотовой связи. Физические величины, отраженные в этих данных, включают локальный трафик, стартовый поток (интенсивность и частота), пропускную способность, потоковые ограничения и другие физические величины в реальной сети. Экспериментальная среда: Количество узлов хранения данных равно 10, емкость диска каждого узла хранения составляет от 100 ГБ до 1 ТБ, объем оперативной памяти - 1-4 ГБ, процессор - Inter Xeon CPU 9 3430 с частотой 240 ГГц, 2,39 ГГц или Inter Core I7-2600 с частотой 3,4 ГГц, а точки передачи данных с максимальной нагрузкой могут быть установлены в диапазоне от 100000 до 400000.

На рис. 3.4 показаны результаты сравнения стратегии ATDA, алгоритма Bubba и алгоритма Random, а также поочередно используются индексы DM, LEST и LBOT. По оси абсцисс на рисунке показано количество точек данных, использованных в эксперименте.

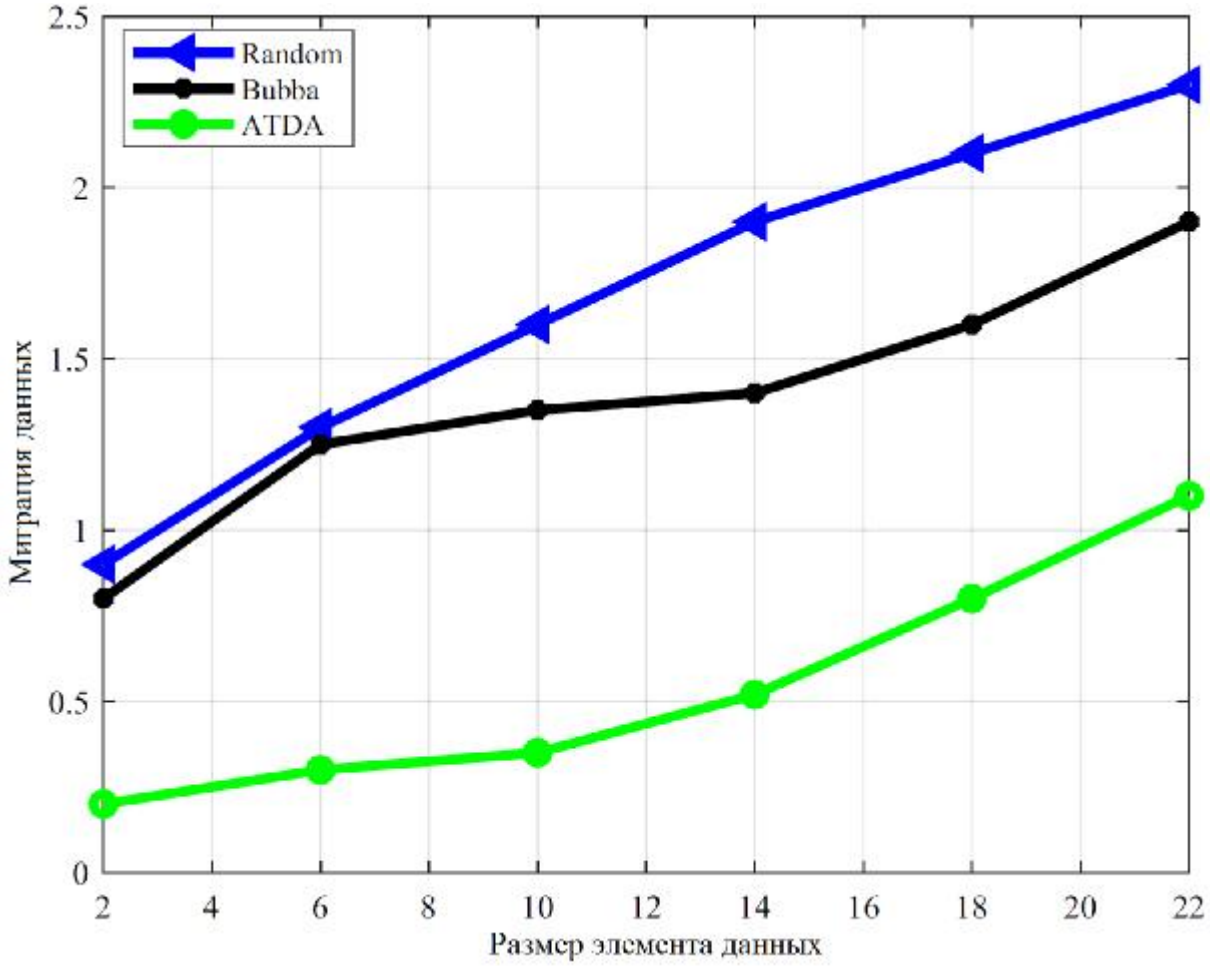


Рис. 3.4. Результаты сравнения стратегии ATDA, алгоритма Bubba и алгоритма Random

Как показано на рис. 3.4, 3.5 и 3.6, стратегия ATDA превосходит алгоритм Bubba и случайный алгоритм как в индексах LBST, так и в индексах DM. В индексе LBOT стратегия ATDA дает худший эффект, чем алгоритм Bubba и случайный алгоритм, при низкой нагрузке. Индекс DM - это объем переноса нагрузки на каждый узел хранения в системе, который отражает стоимость связи с системой и, следовательно, эффективность использования системных ресурсов. Среди индексов DM стратегия алгоритма, описанная в работе, превосходит алгоритм Bubba и алгоритм Random. При низкой нагрузке объем миграции, вызванной стратегией ATDA, очень мал, в то время как при высокой нагрузке объем миграции между страте-

гией ATDA и алгоритмами Bubba и Random, как правило, близок. При низкой нагрузке алгоритм ATDA выдерживает большие колебания нагрузки, в то же время адаптивный механизм обратной связи во временной области делает информационный цикл обратной связи относительно коротким.

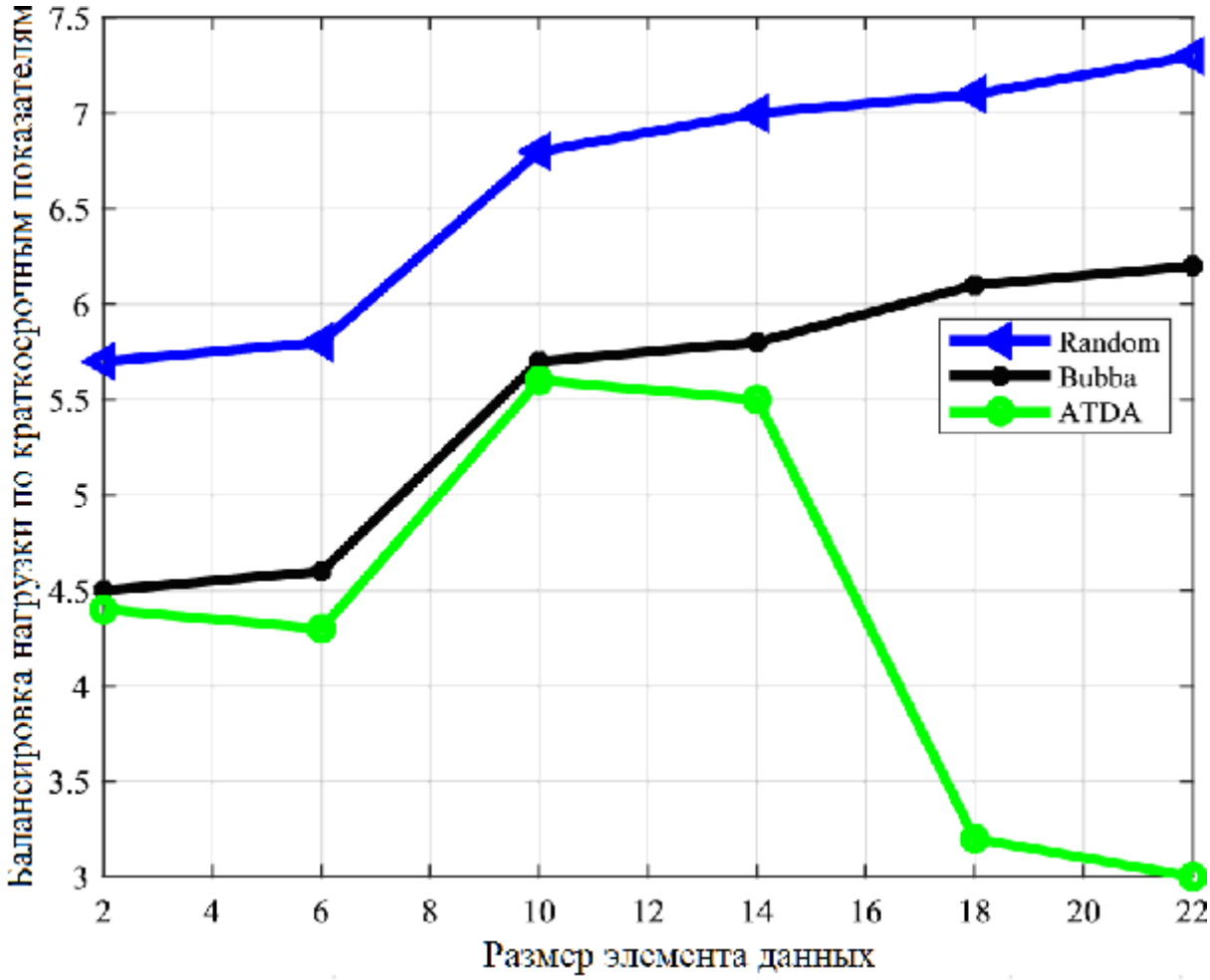


Рис. 3.5. Экспериментальное сравнение алгоритмических стратегий для балансировки нагрузки по краткосрочным показателям

С увеличением нагрузки каждый узел хранения находится в состоянии высокой нагрузки, цикл обратной связи по нагрузке узла хранения уменьшается, а разница между порогом перегрузки и средней нагрузкой уменьшается, поэтому часто начинается перенос нагрузки.

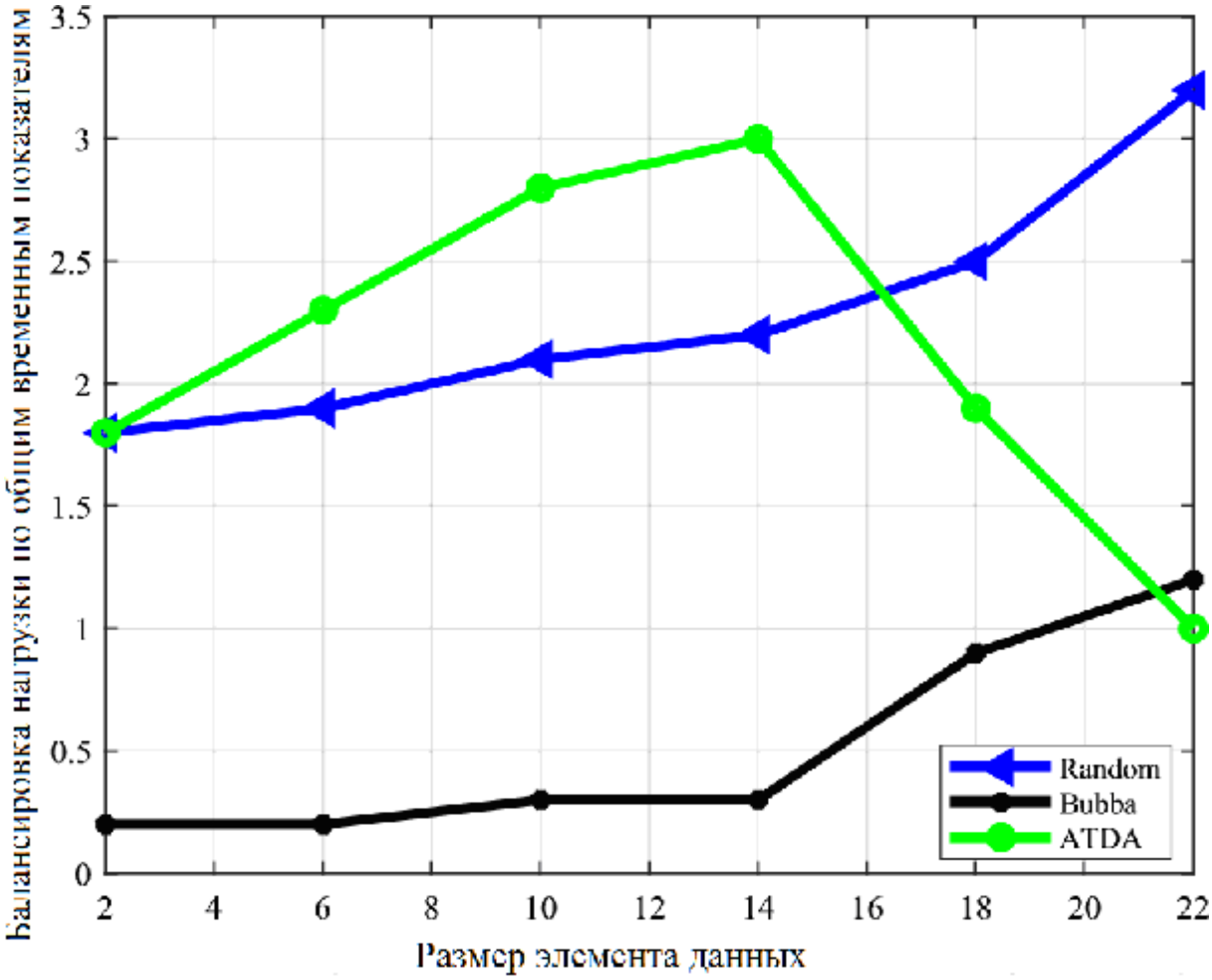


Рис. 3.6. Экспериментальное сравнение стратегий алгоритма балансировки нагрузки по общим временным показателям

При низкой нагрузке результаты стратегии ATDA могут быть аналогичны алгоритму Bubba и случайному алгоритму, в то время как при высокой нагрузке стратегия ATDA превосходит алгоритм Bubba и случайный алгоритм. Обратная связь - используется адаптивный алгоритм во временной области в соответствии с изменением нагрузки на систему, обратная связь по изменению временного интервала загрузки в режиме реального времени, стратегия ATDA при низкой нагрузке, допускающая большие колебания нагрузки, но строго ограничивающая колебания нагрузки при высокой нагрузке, для достижения балансировки нагрузки, адаптивная обратная связь по масштабированию узлов хранения. в то же время обес-

печьте балансировку нагрузки во временной области за короткое время. В процессе перехода от низкой к средней нагрузке показатели, описывающие стратегию ATDA, теоретически снижаются с увеличением разницы между пороговым значением нагрузки и средней нагрузкой, но экспериментальная тенденция противоположна. В результате анализа было установлено, что стратегия статического распределения ATDA заключается в равномерном распределении данных датчиков в соответствии с географической информацией, что позволяет эффективно избежать кластеризации данных в соответствии с их пространственными атрибутами. Таким образом, статическое распределение оказывает большое влияние на балансировку нагрузки при условии очень низкой нагрузки. При низкой нагрузке стратегия ATDA уступала алгоритму Bubba и случайному алгоритму в индексах LBOT, в то время как при высокой нагрузке стратегия ATDA превосходила случайный алгоритм в индексах LBOT и в основном была равна алгоритму Bubba. В случае низкой нагрузки стратегия ATDA предполагает, что значительные колебания нагрузки не повлияют на эффективное использование всей системы в целом, что позволяет в определенной степени сбалансировать нагрузку. В случае высокой нагрузки стратегия ATDA строго контролирует колебания нагрузки, поэтому нагрузка очень сбалансирована.

В Mibench были проведены эксперименты с 11 вариантами использования. На рис. 3.7 показаны статистические результаты предложенного алгоритма по системному индексу в условиях отсутствия итераций, 10 итераций и 20 итераций.

В то же время, в работе проводятся эксперименты по каждой тестовой программе с точки зрения коэффициента затрат на энергопотребление системы, и экспериментальные данные показаны на рис. 3.8. Из экспериментальных данных можно сделать вывод, что использование стратегии жадного распределения данных U_{day} значительно повышает энергопотреб-

ление системы экономия по сравнению с использованием стратегии случайного распределения данных.

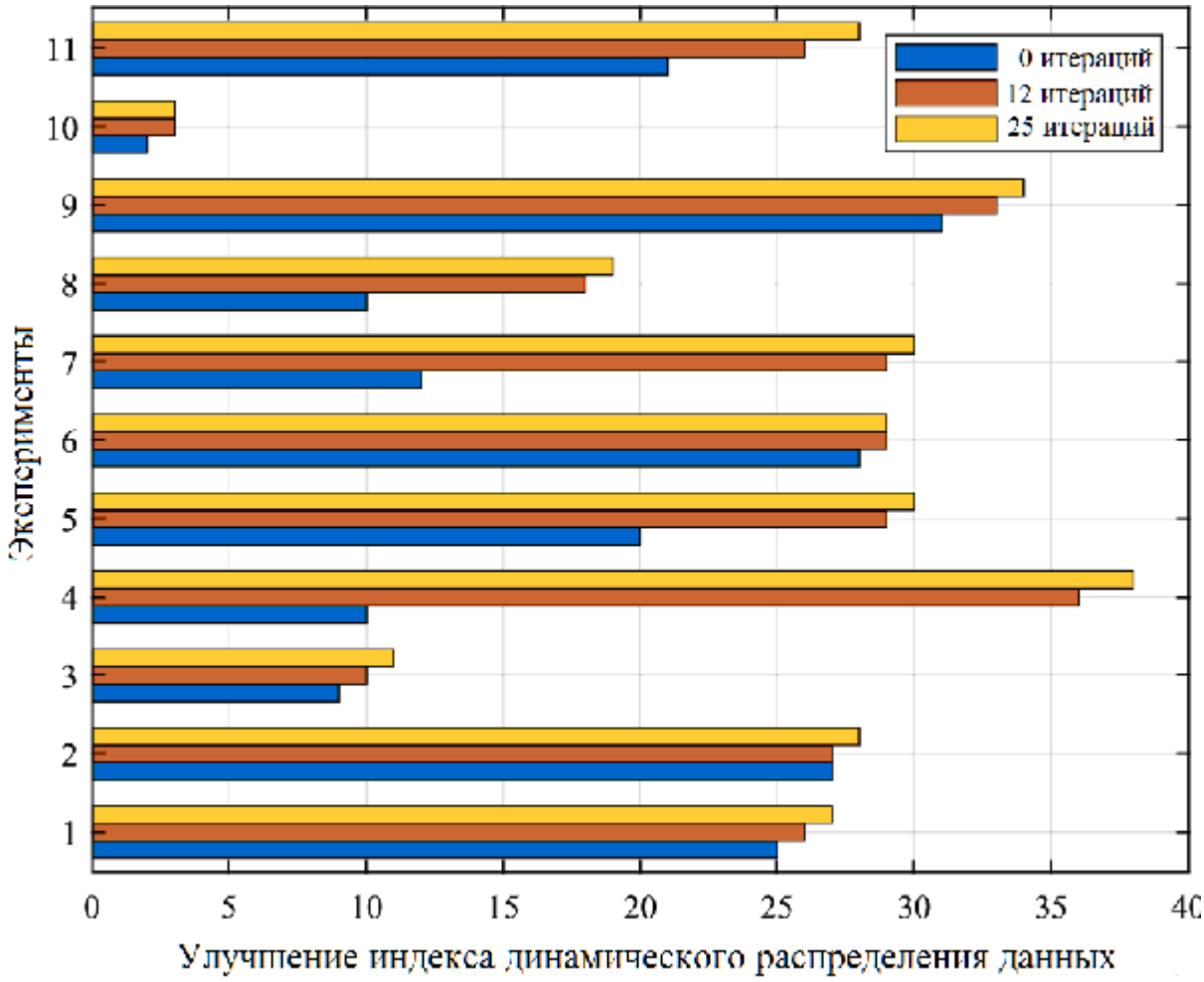


Рис. 3.7. Значение индекса динамического распределения данных увеличивается

Таблица 3.3

Три типа распределения данных для двух программных сегментов (блоков)

Распределение данных	Блок	Основная память	Накладные расходы блока	Стоимость
1	Блок 1	V2,V4,V6	1128	1899
	Блок 2	V1,V3,V5	778	
2	Блок 1	V2,V4,V5	1114	1798
	Блок 2	V1,V3,V5	674	
3	Блок 1	V2,V4,V5	1144	1797
	Блок 2	V2,V5,V6	676	

В среднем алгоритм U_{day} на 25,84% эффективнее стратегии случай-

ного распределения данных в плане экономии энергопотребления системы. Используя глобальный алгоритм оптимального распределения данных, алгоритм случайного распределения данных позволяет снизить общее энергопотребление системы в среднем на 28,74%. Таким образом, для всех тестовых программ глобальный алгоритм оптимального распределения данных, предложенный в работе, также может обеспечить наилучшие результаты оптимизации при снижении общего энергопотребления системы.

На рис. 3.9 представлена гистограмма, соответствующая задержке доступа к системе и энергопотреблению каждой тестовой программы в этом разделе. На гистограмме можно наглядно увидеть сравнение трех различных стратегий распределения данных. Очевидно, что алгоритм оптимального распределения данных динамического программирования превосходит два других алгоритма.

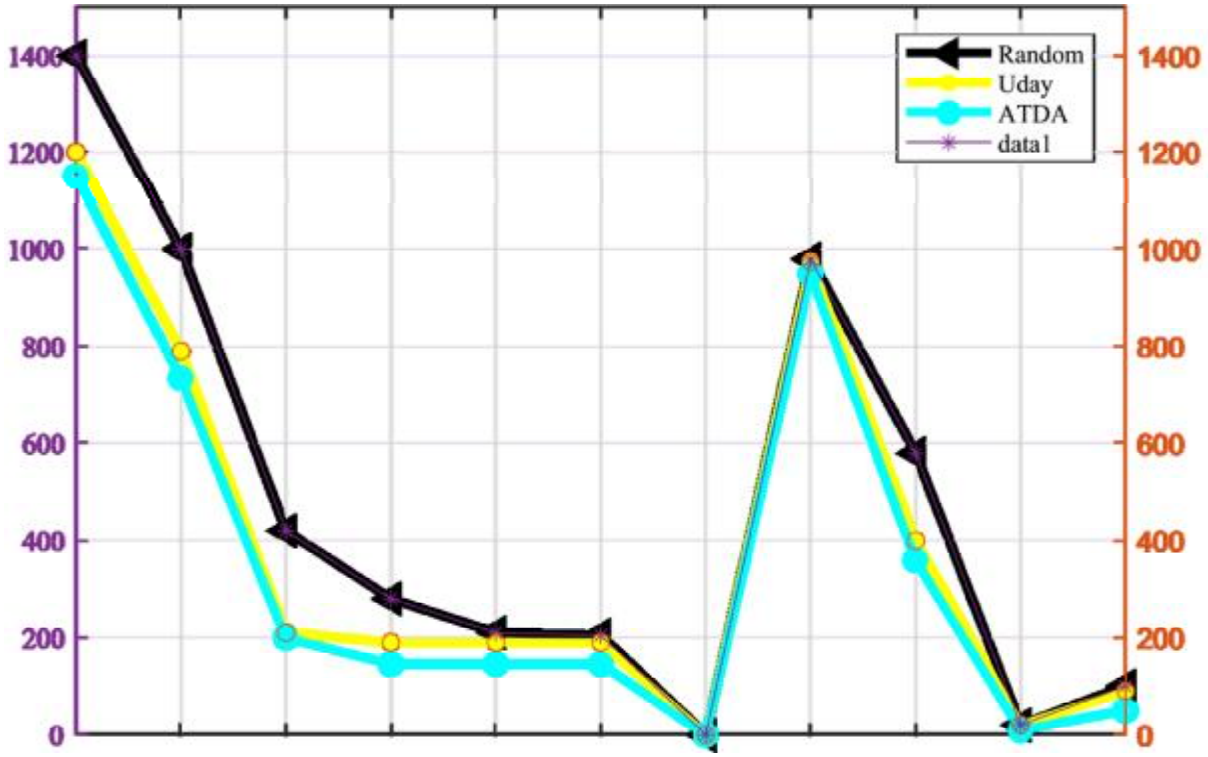


Рис. 3.8. Диаграмма распределения данных и энергопотребления для доступа к датчикам Интернета вещей

После проверки информация о скорости загрузки будет проведена инициализация новых узлов хранения, и система будет считать, что скорость загрузки новых узлов хранения самая низкая. Между тем, добавление новых узлов хранения в систему снизит среднюю нагрузку на систему, что упростит перенос нагрузки на новые узлы и увеличит количество узлов хранения без каких-либо изменений.

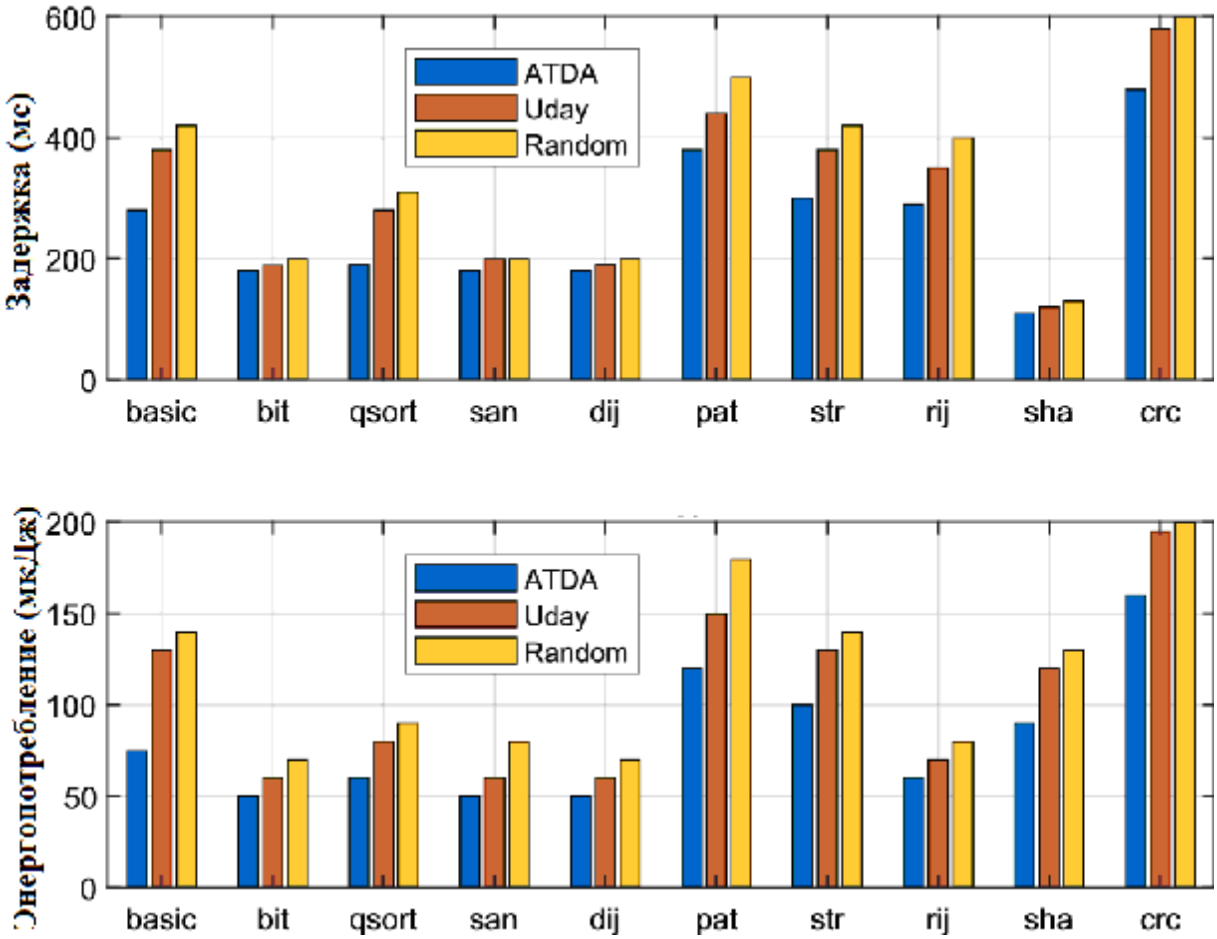


Рис. 3.9. Гистограмма задержки доступа и энергопотребления

3.6. Выводы к главе 3

В главе представлена стратегия распределения данных для хранения больших объемов сенсорной информации в Интернете вещей. Представлена модель распределения данных во временной области, а статическое распределение данных сводится к задаче линейного программирования с использованием характеристик корреляции информационного пространст-

ва и дискретной выборки информации в Интернете вещей. На основе непрерывного изменения информации, поступающей от датчиков, разработана адаптивная стратегия обратной связи во временной области. Наконец, эвристическая функция порога загрузки используется для динамической регулировки нагрузки и оптимизации стратегии распределения данных в целом. Эксперименты показывают, что эта стратегия эффективна при балансировке нагрузки в реальном времени и использовании системы.

Предложен итерационный алгоритм оптимального распределения данных, основанный на динамическом программировании, который нацелен на переменные массива во встроенной системе многоуровневого хранения и обеспечивает оптимальное распределение цикла таким образом, чтобы минимизировать накладные расходы на доступ ко всему объему цикла.

Далее будет продолжено изучение проблемы оптимизации, рассматривая как операции запроса, так и обновления сенсорной информации Интернета вещей, и объединим характеристики сенсорной информации и преимущества платформы распределенных вычислений, чтобы более всесторонне изучить проблему оптимизации распределения данных сенсорной информации Интернета вещей. и систематическим образом при условии частых запросов. В большинстве существующих исследований используется единый механизм обнаружения и защиты от определенного типа вредоносного поведения. Однако из-за ограниченности сферы применения сетевого протокола единый уровень обнаружения и механизма защиты не может справиться с большим количеством видов вредоносного поведения. В сочетании с данными Интернета вещей, используемыми для сбора данных о сетевых протоколах на всех уровнях, будет разработан легкий комплексный метод обнаружения вторжений и оптимизации энергоэффективности, который имеет более широкую область применения.

Литература к главе 3

3.1. Al-Turjman F (2018) Information-centric framework for the internet of things (IoT): traffic modeling& optimization. Futur Gener Comput Syst 80:63–75

3.2. Bijarbooneh FH, Du W, Ngai CH (2017) Cloud-assisted data fusion and sensor selection for internet-of-things. IEEE Internet Things J 3(3):257–268

3.3. Bi K, An K, Li X (2020) A resource optimization allocation strategy for China's shipbuilding industry green innovation system. Int J Innov Technol Manag 17(4):2050029–2050042

3.4. Choi K, Chung SH (2017) Enhanced time-slotted channel hopping scheduling with quick setup time for industrial Internet of Things networks. Int J Distrib Sens Netw 13(6):1362–1377

3.5. Ding K, Zhao H, Hu X (2017) Distributed channel allocation and time slot optimization for green internet of things. Sensors 17(11):2479–2491

3.6. Galinina O, Tabassum H, Mikhaylov K (2016) On feasibility of 5Ggrade dedicated RF charging technology for wireless-powered wearables. IEEE Wirel Commun 23(2):28–37

3.7. He Y, Zhang S, Tang L (2020) Large scale resource allocation for the internet of things network based on ADMM. IEEE Access 8:57192–57203

3.8. Jiang W, Wang H, Li B (2020) A multi-user multi-operator computing pricing method for Internet of things based on bilevel optimization. Int J Distrib Sens Netw 16(1):155014–155032

3.9. Jiao J, Sun Y, Wu S (2020) Network utility maximization resource allocation for NOMA in satellite-based internet of things. IEEE Internet Things J 7(4):3230–3242

3.10. Ke H, Wang J, Wang H (2019) Joint optimization of data offloading and resource allocation with renewable energy aware for IoT devices: a deep reinforcement learning approach. IEEE Access 7:179349–179363

3.11. Liu XM (2020) Uplink resource allocation for multicarrier grouping cognitive internet of things based on K-means Learning. Ad Hoc Netw 96:102002–102002

3.12. Liu M, Li D, Zeng Y (2020a) Combinatorial-oriented feedback for sensor data search in internet of things. IEEE Internet Things J 7(1):284–297

3.13. Liu X, Zhang X (2020) NOMA-based resource allocation for clusterbased cognitive industrial internet of things. IEEE Trans Industr Inf 16(8):5379–5388

3.14. Liu X, Ding H, Zhang X (2020b) Rate satisfaction-based power allocation for NOMA-based cognitive Internet of Things. Ad hoc Netw 98(Mar):102063.1-102063.8

3.15. Liu X, Jia M, Ding H (2020c) Uplink resource allocation for multicarrier grouping cognitive internet of things based on K-means Learning. Ad hoc Netw 96(Jan):1020021–1020029

3.16. Li Q, Ma X, Peng H (2015) Data fusion optimization model of elastic wave in wireless sensor networks. J Comput Inf Syst 11(3):815–822

3.17. Li X, Tan L, Li F (2019) Optimal cloud resource allocation with cost performance tradeoff based on internet of things. Internet Things J IEEE 6(4):6876–6886

3.18. Luong NC, Hoang DT, Wang P (2017) Data collection and wireless communication in internet of things (IoT) using economic analysis and pricing models: a survey. IEEE Commun Surv Tutor 18(4):2546–2590

3.19. Mcnarie T, Quist G, Lewinger K (2017) Into the information age: application of internet of things for sewer maintenance optimization. Proc Water Environ Fed 2017(2):499–516

3.20. Ning Z, Wang X, Kong X (2017) A social-aware group formation framework for information diffusion in narrowband internet of things. IEEE Internet Things J 5:1527–1538

3.21. Rehman M, Liew C, Wah T (2015) Mining personal data using smartphones and wearable devices: a survey. Sensors 15(2):4430–4469

3.22. Rullo A, Midi D, Serra E (2017) Pareto optimal security resource allocation for internet of things. ACM Trans Inf Syst Secur 20(4):15.1-15.30

3.23. Safia A, Aghbari Z, Kamel I (2017) Efficient data collection by mobile sink to detect phenomena in internet of things. Information 8(4):123–143

3.24. Salam A, Javaid Q, Ahmad M (2020) Bioinspired mobility-aware clustering optimization in flying ad hoc sensor network for internet of things: BIMAC-FASNET. Complexity 20:1–20

3.25. Stella K, Giorgos A, Symeon P (2015) On the optimization of a probabilistic data aggregation framework for energy efficiency in wireless sensor networks. Sensors 15(8):19597–19617

3.26. Sun Z, Xing X, Wang T (2019) An optimized clustering communication protocol based on intelligent computing in informationcentric internet of things. IEEE Access 7(99):28238–28249

3.27. Tao H, Miaowang Z, Lijuan Z (2018) A channel-aware duty cycle optimization for node-to-node communications in the internet of medical things. Int J Parallel Prog 48:264–279

3.28. Wang M, Xu C, Chen X (2019) Design of multipath transmission control for information-centric internet of things: a distributed stochastic optimization framework. IEEE Internet Things J 6(6):9475–9488

3.29. Xiang M, Wang D (2019) Performance analysis and optimization for coverage enhancement strategy of Narrow-band Internet of Things. Future Gener Comput Syst 101(C):434–443

3.30. Xu S, Wang X, Yang G (2020) Routing optimization for cloud services in SDN-based internet of things with TCAM capacity constraint. J Commun Netw 22(2):145–158

3.31. Ya´nez W, Mahmud R, Bahsoon R, Zhang Y, Buyya R (2020) Data allocation mechanism for internet-of-things systems with blockchain. IEEE

Internet Things J 7(4):3509–3522

3.32. Yin X, Li S, Lin Y (2019) A novel hierarchical data aggregation with particle swarm optimization for internet of things. Mobile Netw Appl 24(6):1994–2001

4. Интеграция больших данных в системы принятия решений

Большие данные - важная тема для обсуждения и исследований. Эта тема приобрела такое значение благодаря значимой ценности, которую можно извлечь из этих данных. Применение больших данных в современном бизнесе позволяет предприятиям принимать более быстрые и разумные решения, обеспечивая реальное конкурентное преимущество. Однако многие проекты с использованием больших данных дают разочаровывающие результаты, которые по многим причинам не отвечают потребностям лиц, принимающих решения. Основная причина этого провала заключается в пренебрежении изучением аспектов принятия решений в этих проектах. В свете этой проблемы, в качестве решения в исследовании предлагается интеграция аспекта принятия решений в Big Data. Таким образом, представлены три основных вклада:

1) Разъясняется определение больших данных;

2) Представлена модель BD-Da, концептуальная модель, описывающая уровни, которые следует учитывать при разработке проекта больших данных, для имплементации в систему принятия решения корпоративного программного обеспечения;

3) Представлен подход к разработке проекта, связанного с исследование Big Data. Цель подхода – поддержка принятия решений в организации. Особенность подхода – базирование на требованиях.

4.1. Большие данные и проблема принятия решений

Мы живем в эпоху, когда наблюдаем массовое и непрерывное производство данных в различных форматах (видео, изображения, текст и т.д.) пользователями в социальных сетях, устройствах Интернета вещей, смарт-устройствах и других источниках. Можно сказать, что мы вступили в эпоху больших данных, когда большие данные все чаще становятся важной

областью для дискуссий и исследований. Большое количество ученых и исследователей из многих дисциплин написали на эту важную тему. Большие данные приобрели такое значение из-за значительной ценности, которую можно извлечь из обработки и анализа этих данных [4.42].

Большие данные могут играть ведущую роль в принятии решений современной организацией. В настоящее время популярна идея о том, что большие данные позволяют компаниям создавать мощную основу для принятия более качественных, быстрых, подкрепленных фактическими данными и надежных решений [4.26]. В [4.47] указано, что заявили, что применение больших данных в современном бизнесе обеспечивает понимание и бизнес-аналитику в режиме реального времени, например, тенденции и характеристики их клиентов, что позволяет компаниям быстро реагировать и оптимизировать процессы принятия решений, что, в свою очередь, может привести к повышению эффективности бизнеса и конкурентному преимуществу [4.47]. Аналогичным образом, в [4.30] отмечено, что если организациям удастся внедрить инструменты и методы работы с большими данными в свой бизнес для извлечения правильной и полезной информации из данных, они смогут поддерживать процесс принятия более быстрых и адекватных решений, ведущих к снижению затрат, разработке новых продуктов, созданию оптимизированных тендеров и появлению новых клиентов. о тенденциях рынка, таким образом, они могут создать устойчивое конкурентное преимущество [4.30]. Например, в [4.4] установлено, что компании, принимающие решения на основе данных, могут повысить производительность на 5-6%.

Данные являются стратегическим активом, но они бесполезны, если их не использовать конструктивно и надлежащим образом для получения ценных результатов [4.34]. Все современные организации стремятся использовать большие данные, в то время как об эффективном использовании аналитики больших данных при решении бизнес-задач или принятии

решений известно очень мало [4.2]. В результате многие проекты с использованием больших данных, разработанные организациями, потерпели неудачу или не достигли поставленных целей. Например, в [4.25] указано, что 55% проектов с использованием больших данных дают разочаровывающие результаты, в то время как в [4.20] указано, что 60% проектов в области больших данных потерпели неудачу и были свернуты в течение 2017 г.

В литературе показано, что причина, по которой эти проекты не дают желаемых результатов, заключается в том, как были разработаны эти проекты, поскольку организации, как правило, фокусируются на самих данных и на данных аналитики без акцента на принятии решений, что является фактическим использованием больших данных. В этом контексте [4.54] указано, что сосредоточение внимания на определении требуемых данных, используемых аналитических технологий и рабочего процесса без сосредоточения внимания на принятии решений как таковом препятствует проектам интеллектуального анализа данных и прогнозного анализа для получения желаемого результата. В другом исследовании [4.25] объясняется, что основной причиной провала проектов с использованием больших данных было отсутствие связи между менеджерами, которые представили глобальное видение проекта – глобальное видение решения, которое необходимо принять, и желаемую информацию, полученную в результате анализа данных, – и командой аналитиков больших данных, ответственной за фактическую реализацию. Кроме того, отсутствие бизнес-контекста, связанного с данными, и недостаточный опыт ведения бизнеса препятствуют общему пониманию бизнеса, бизнес-целей и их использованию в задачах подсистем поддержки принятия решений [4.25].

Несмотря на признание жизненно важной роли больших данных в поддержке принятия решений в организации, большинство исследований сосредоточены на технологических аспектах больших данных, игнорируя

изучение аспекта принятия решений, который заключается в фактическом использовании больших данных. Исходя из этой задачи, основной исследовательский 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?

Чтобы ответить на исследовательский вопрос, сначала нужно ответить на иной вопрос: какие концепции, связанные с большими данными, встречаются в литературе? И чтобы ответить на него, необходимо провести библиографический поиск, чтобы найти статьи, которые определяют эти понятия и обеспечивают теоретическую поддержку для построения интегрированной модели, предложенной в этом исследовании.

Таким образом, методология систематического обзора литературы (SLR) является подходящим и полезным подходом, позволяющим сделать процесс исследования более точным и менее предвзятым, поскольку SLR - это средство оценки и интерпретации всех доступных исследований, относящихся к конкретному исследовательскому вопросу, тематической области или интересующему нас явлению, с использованием надежной, строгой и проверяемой методологии [4.27]. Соответственно, применялся зеркальный подход, чтобы обеспечить всестороннее понимание концепции больших данных. В результате было установлено, что большие данные нельзя рассматривать только в терминах данных, но есть и другие концепции, которые следует учитывать при определении, а именно: наборы данных с новыми характеристиками, жизненный цикл аналитики данных, технология, аналитический методы, понимание и принятие решений.

В работе предлагается концептуальная модель (BD-Da) для Big Data для имплементации интегрированных подходов в подсистему принятия решений организации. Модель BD-Da разделяет эти концепции на три уровня, которые необходимо учитывать при разработке проекта с исполь-

зованием больших данных, направленного на решение проблемы принятия решений в организации. Эти уровни - уровень данных, анализа данных и принятия решений.

4.2. Концепция больших данных в исследованиях

Количество публикаций, посвященных большим данным, постоянно растет. Например, поиск в Google Scholar по ключевому слову «большие данные» позволил получить 50 миллионов научных работ по многим дисциплинам за 0,04 секунды. Повторное использование термина «большие данные» в разных контекстах сопровождалось увеличением количества существующих определений больших данных. В результате в литературе нет единого определения термина «большие данные», но существуют разнообразные и даже часто противоречивые определения, описывающие этот термин [4.9]. Это противоречие привело к двусмысленности и путанице среди исследователей и практиков, что, в свою очередь, могло помешать эффективному развитию данной темы [4.9]. Например, в исследовании [4.15] объясняется, что отсутствие комплексного определения больших данных представляет собой проблему для их исследований по разработке показателей качества больших данных из-за двусмысленности концепции больших данных, а также путаницы характеристик больших данных и характеристик качества данных. В некоторых определениях больших данных характеристики качества данных определяются как характеристики, которые отличают большие данные, что усложняет процесс определения концепций и инструментов качества больших данных.

В литературе существует множество исследований, в которых предпринимались попытки проанализировать различные определения больших данных, чтобы дать четкое и краткое общее определение этого понятия, которое устранило бы двусмысленность и уменьшило бы путаницу, связанную с его использованием. Среди них:

• различные определения больших данных в литературе охватывают по крайней мере один из следующих аспектов [4.58]: размер (огромный объем данных), сложность данных и технологии, используемые для обработки больших или сложных наборов данных (инструменты и методы).

• в [4.23] выделено три категории определений, которые важны для определения различных аспектов больших данных. Это атрибутивные определения, сравнительные определения и архитектурные определения.

• существующие определения больших данных можно разделить на шесть категорий [4.31]. Первая категория фокусируется на характеристиках данных, вторая – на технологии анализа данных, третья – на коммерческой ценности данных, четвертая – на структуре больших данных, пятая – на объеме и источнике данных, а в некоторых определениях большие данные рассматриваются как явление – последняя категория определений не является преобладающей.

• существует [4.9] четыре основные темы, связанные с определением больших данных в современной литературе. Это (1) информация: создаваемая информация характеризуется «Объемом», «Скоростью» и «Разнообразием». (2) Для анализа больших данных необходимы специальные технологии и (3) методы. (4) Влияние: необходимо извлекать ценную информацию из больших данных, которая влияет на компании и общество.

• В [4.15] рассмотрены различные определения больших данных, чтобы предложить альтернативное определение. В результате определено три взаимодополняющих элемента, определяющих термин «большие данные»: характеристики данных, архитектура и обработка и применение больших данных.

4.2.1. Процесс принятия решений в организации как приложение больших данных

В [4.17] определено принятие решений как процесс выбора среди

альтернативных вариантов действий для достижения целей и задач. Весь процесс принятия управленческих решений является синонимом управления [4.51].

Роль менеджеров в организации – принимать решения. В [4.51] классифицированы различные решения, принимаемые менеджерами, на два основных типа управленческих решений относительно структуры этих проблем, а именно: запрограммированные и непрограммированные решения. В дополнение к этим двум категориям в [4.19] определен еще один класс решений, который находится между запрограммированными и непрограммированными решениями, а именно полуструктурированные решения.

• **Программированные решения:** такого рода решения принимаются для решения рутинных проблем, которые хорошо структурированы, хорошо понятны и обычно повторяются. Подобные проблемы имеют стандартные методы решения. Когда организация впервые сталкивается с этой проблемой, ей могут потребоваться большие усилия, чтобы принять решение и решить проблему. Чтобы решить проблему впервые, требуется системный подход, но в результате такого подхода проблема будет иметь алгоритмическое решение, которое можно будет применять для поиска приемлемого решения каждый раз, когда возникает одна и та же проблема [4.37].

• **Незапрограммированные решения:** будут направлены на решение тех проблем, которые являются нечеткими, сложными, не очень структурированными, возникают нечасто, для их решения не существует рутинных или систематических процедур. Эти решения могут быть поддержаны передовыми инструментами поддержки принятия решений, включая большие данные, но не могут быть автоматизированы, поскольку способности менеджеров и человеческая интуиция часто лежат в основе принятия решений [4.37].

• **Полуструктурированноые решения:** полуструктурированные проблемы имеют некоторые структурированные элементы и некоторые другие неструктурированные элементы. Решение полуструктурированной проблемы предполагает сочетание как стандартных процедур решения, так и человеческого суждения [4.12].

Существует еще одна классификация управленческих решений, основанная на организационных уровнях, на которых эти решения принимаются, предложенная в [4.12]. Определены три категории решений: стратегические, тактические и оперативные решения.

• **Стратегические решения:** это решения, принимаемые на самом высоком уровне управления. Это сложные, нечастые и очень влиятельные решения, которые определяют цели, предназначение и направление организации бизнеса, а также его отношение к внешней среде. Эти решения требуют большого количества людей, времени и денег. Обычно они не запрограммированы по своей природе [4.37].

• **Тактические или административные решения:** эти решения касаются реализации стратегических решений. Их составляют менеджеры среднего звена, такие как руководители подразделений или отделов. Они менее эффективны, более конкретны и конкретны и более ориентированы на действия, чем стратегические решения, поскольку стратегические решения применяются ко всем отделам внутри организации, а тактические решения формулируют цели предприятия в конкретной ведомственной манере [4.37].

• **Оперативные решения:** эти решения, связанные с ходом повседневной деятельности предприятия, принимаются на низшем уровне управления. Эти решения носят административный характер, принимаются неоднократно и менее рискованны. Они предназначены для доработки тактических решений [4.37]. Принятие решений является сердцем всех управленческих функций, и что богатый процесс принятия решений является

основой успеха предприятия, поскольку принятие решений абсолютно необходимо для получения и поддержания конкурентного преимущества [4.17]. В литературе предлагается ряд моделей процесса принятия решений; эти модели определяют путь к правильному решению через ряд этапов. Так, исследование [4.51] представляет процесс принятия решений как трехфазную систему, включающую три фазы: фазу разведки, проектирования и выбора (модель IDC): (I) фаза разведки относится к поиску в окружающей среде условий, проблем или возможностей, вызову для решения; (D) Фаза проектирования относится к разработке и анализу альтернативных решений проблемы или возможности; (C) Фаза выбора относится к выбору одной или нескольких доступных альтернатив. Позже был добавлен четвертый этап: внедрение и мониторинг можно считать пятым этапом - формой обратной связи [4.12].

Проанализировав множество моделей процесса принятия решений, авторы заметили, что эти модели разделены на три основных этапа: определение решения, создание и оценка альтернатив и выбор альтернативы или нескольких. Некоторые модели включают реализацию и оценку вариантов. Эти этапы соответствуют этапам процесса принятия решений [4.51], а именно: анализ, проектирование, выбор, реализация и анализ. Связь этих моделей с моделью [4.51] представлена в табл. 4.1.

Таблица 4.1

Связь модели IDC с предыдущими моделями

Источник	**Проблематика**	**Дизайн**	**Выбор**	**Выполнение**	**Обзор**
[4.36]	Этап идентификации	Этап разработки	Этап выбора		
[4.8]	• Распознавание проблем • Определение проблемы	• Альтернативное поколение • Разработка модели	Выбор	выполнение	

Источник	Проблематика	Дизайн	Выбор	Выполнение	Обзор
[4.44]	• Изложите решение • Разработать цели • Разделите цели на «должны» и «хочу». • Взвесьте самые важные желания.	• Генерировать альтернативы • Экран Альтернативы • Сравните альтернативы с желаниями • Определить неблагоприятные последствия	Сделать лучший сбалансированный Выбор		Просмотр данных
[4.32]	Выявление проблемы	• Создание альтернатив • Оценка альтернатив	Выбор альтернативы	Реализация решения	Оценка эффективности решений
[4.39]	• Определение проблемы. • Собирая информацию	Определение альтернатив.	• Нахождение консенсуса и выбор альтернативы. • Предвидение последствий решения.	Реализация альтернативы	
[4.35]	• Ситуационные изменения, требующие перемен Анализ (определить ситуацию, требующую решения) • Формулировка проблемы и причинно-следственная связь	Генерация идей решения	Выбор набора решений	Реализация и планирование последствий	

Источник	Проблематика	Дизайн	Выбор	Выполнение	Обзор
	Анализ				
[4.32]	• Определение проблемы • Выявление и ограничение факторов	• Разработка потенциальных решений • Анализ альтернатив	Выбор лучшей альтернативы	Реализация решения	Создание системы контроля и оценки
[4.3]	Выявление проблемы	• Разработка альтернатив • Анализ альтернатив	Выбор альтернативы	Реализация альтернативы	Оценка эффективности решений

4.2.2. Стандарт модели принятия решений и обозначений (DMN)

Большие данные предоставляют организациям беспрецедентную возможность принимать более быстрые и разумные решения, основанные на знаниях. Чтобы использовать эту возможность, организации необходимо глубокое понимание этих решений и их требований, чтобы иметь возможность связать знания, извлеченные из больших данных, со своими решениями [4.10]. Моделирование решений - это мощный метод представления решений и требований к ним в ясном, кратком и понятном формате [4.16]. В этом контексте OMG (Object Management Group) разработал новый стандарт - стандарт модели и нотации решений (DMN) для моделирования бизнес-решений [4.10]. Основная цель DMN - предоставить всем бизнес-пользователям понятную общую систему обозначений [4.40].

Моделирование решений с использованием стандарта DMN включает два уровня, диаграммы требований к принятию решений (DRD) и логику принятия решений, которые можно использовать независимо или вместе в модели принятия решений [4.40]. DRD включает в себя набор элементов и взаимосвязь между ними. Эти элементы определяют решение,

которое будет принято, и то, как оно зависит от других решений, политики или регулирования (источник знаний), бизнес-знаний (модель знаний) и входных данных [4.40].

Для интеграции логики принятия решений принято использовать логику отдельных компонент задачи принятия решений. Это:

- реализуемые в реальном времени аналитические модели;
- таблицы решений;
- бизнес-правила.

Применение логики принятия решений необходимо для верификации и автоматизации соответствующих процессов [4.40].

Стандарт DMN целесообразно использовать следующим образом:

- реализация системы автоматизации принятия решений;
- моделирование требований к системе автоматизации;
- моделирование решений ЛПР с применением DRD.

Моделирование требований к принятию решений можно использовать для моделирования любого решения, будь то стратегическое, тактическое или оперативное решение, при условии, что это решение заслуживает моделирования. Если решение является динамичным, способ принятия этого решения часто меняется, существует широкий набор альтернативных решений на выбор, применяется множество политик или правил, или, если оно основано на большом количестве данных, то это решение стоит смоделировать [4.54].

4.2.3.Модели принятия решений на основе больших данных

Известно [4.2], что применение Big Data делает процесс принятия решений более эффективным, что положительно сказывается на качестве принимаемых решений. Вместе с тем разнозненность данных в организации мешет принятию эффективных решений, что отрицательно сказывается на рабочем процессе. К сожалению, задача интеграции больших данных

для описываемых специфических задас до сих пор является малоисследованной.

Например, исследования [4.45, 4.46] содержат пример такой интеграции. Описаны элементы, существенные для испльзования Big Data в динамике принтия решений. Эти сущестенные элементы – большие данные, бизнес-аналитика (BI) и принятие решений.

Система поддержки (DSS) с моделью IDC основана на исследовании Big Data вместе с BI для формирования данных, которые будут способствовать ЛПР для улучшения качества принимаемых решений. ЛПР также может использовать DSS в качестве подсистемы прогнозирования и выбора альтернатив.

В другом исследовании [4.13] предложена структура под названием «B-DAD Framework».

Платформа B-DAD объединяет инструменты, архитектуру и анализ больших данных в модель IDC. Чтобы оценить и продемонстрировать свою структуру, проведено исследование в сфере розничной торговли. Цель состояла в том, чтобы определить рекламные продукты, когда их следует предлагать, и изучить влияние социальных сетей с использованием объективных данных. Источники данных – POS-терминалы, ERP-системы, Социальные сети.

Модель для поддержки принятия решений в области общественной безопасности и снижения количества преступлений в регионе описана в [4.55] предлагает сбор и анализ большого количества данных, установления критериев для оценки альтернатив, использования мультикритериальных методов для оценки альтернатив и выбора наиболее подходящего для реализации.

На основе SLR и качественных исследований в [4.2] определено шесть взаимосвязанных и повторяющихся ключевых шагов, которые обеспечивают четкое и полезное руководство по использованию BDA при при-

нятии решений:

определить проблему или решение, которое, как ожидается, будет принято с помощью анализа больших данных,

проанализировать соответствующие прошлые результаты и контекст, чтобы избежать повторения и ошибок,

выбрать переменные и разработать модель, которая представляет проблему,

собрать все соответствующие данные из разных источников для измерения и тестирования модели,

анализировать данные, чтобы получить ценную информацию,

принимать меры по решению проблемы на основе информации, полученной из больших данных.

В другом исследовании была предложена новая модель процесса принятия решений [4.6], которая интегрирует большие данные на этапе модели IDC.

Нстоящее исследование - еще одна попытка предложить решение, которое поможет менеджерам воспользоваться мощью больших данных для создания возможностей для решения конкретной организационной проблемы. В этом исследовании предлагается модель, определяющая три аспекта, а именно: аспекты данных, аспект анализа данных и аспекты принятия решений. Наше исследование сосредоточено на определении этих аспектов, включая аспект принятия решений, который был исключен из проектов по работе с большими данными. Чтобы определить аспект принятия решения в нашей модели, авторы прибегли к двум ключевым элементам, а именно: процессу принятия решений и моделированию решений с использованием расширенной модели и обозначения oDMN+. Некоторые исследования предоставляют убедительные иллюстрации вклада моделирования решений в правильную постановку проблемы и определение правильных моделей анализа данных, которые будут построены на уровне

анализа данных.

Например, в [4.52, 4.54] использовано моделирование решений с использованием стандарта DMN и методологии CRISP-DM для усиления своих проектов по анализу данных. Они предложили разработать модель требований к принятию решений на первом этапе методологии CRISP-DM, то есть на этапе понимания бизнеса, чтобы сформулировать свои аналитические требования. Таким образом, они обеспечивают четкое понимание бизнеса и эффективный запуск проекта. Новый подход на основе DMN помог аналитической команде оживить ранее неразрешимые проекты из-за ошибочного начала этих проектов, которое привело к неправильному пониманию их целей. Использование DMN обеспечило общий язык между бизнес-клиентом и командой аналитиков. Это позволяет командам понять контекст принятия решения, например, цели или показатели, на которые влияет это решение, входные данные, источники знаний и другие решения, необходимые для принятия решения.

Другое исследование [4.22] показало, что моделирования решений с использованием стандарта DMN недостаточно для моделирования решений в контексте Big Data. В [4.22] показано, что DMN удалось отобразить связь между решениями и требуемой информацией, однако ей не удалось отобразить связь между информацией и источниками данных, которые предоставляют эту информацию. Данные, создаваемые в настоящее время, предоставляются из разных источников, и в любое время могут появиться новые источники данных, поэтому организация полагается на большие данные для принятия своих решений, необходимо идентифицировать различные возможные источники данных, которые могут предоставить информацию, необходимую для этих целей. Чтобы удовлетворить эту потребность, разработана расширенная модель и обозначение oDMN+ на основе стандарта DMN. Расширенная модель и обозначение oDMN+ позволяют описать связь между множеством сущностей. Среди них:

- принятие решений,

- объективные данные,

- необходимая информация.

В [4.22] показано, что использование oDMN+ позволило образом, ускорять принимаемые решения. Это также помогло найти альтернативные источники данных, которые можно использовать в соответствии с требованиями лиц, принимающих решения.

Основное различие между этим исследованием и предыдущими исследованиями заключается в том, что это исследование является единственным среди этих исследований, целью которого было определение концепции больших данных с точки зрения принятия решений. В этом контексте данное исследование определяет и систематизирует различные аспекты больших данных, которые следует учитывать для разработки проекта больших данных, направленного на решение проблемы, требующей решения. В этом исследовании была предложена модель, описывающая аспекты больших данных, целые концепции, составляющие каждый аспект, а также взаимосвязь между ними. Помимо концепций, связанных с большими данными, в литературе предлагаются новые концепции о больших данных с аспектом принятия решения (BD-DA), в частности, понятие аспекта решения: решение, модель решения, альтернатива и выбор. Определение аспекта решения позволяет правильно сформулировать решение, которое будет принято, и бизнес-информацию, необходимую для анализа данных.

Кроме того, в этом исследовании описывается конкретный, логический, основанный на требованиях подход к разработке проекта анализа больших данных для поддержки принятия решений. В отличие от предыдущих исследований, в которых подчеркивалась только важность признания проблемы и правильной формулировки проблемы для принятия разумных и достаточных решений, в нашем исследовании предложено моделирование решений с использованием oDMN+ в качестве метода, обеспе-

чивающего надежную поддержку этого шага, поскольку оно будет поддерживать сотрудничество и общение между лицами, принимающими решения, и экспертами по большим данным.

4.3. Методология исследования

Чтобы предоставить обзор существующих определений, связанных с термином «большие данные» в литературе, в исследовании использовался подход SLR, следующий процессу, предложенному в [4.27] и примененному в [4.2]. В рамках процесса разработан протокол, в котором были указаны вопросы исследования, стратегия поиска, критерии включения и исключения, извлечение данных и синтез исследований.

Чтобы обсудить и уточнить определение больших данных в литературе и дать четкое определение различных аспектов больших данных. В этой работе исследуется следующий исследовательский вопрос: каковы концепции, связанные с большими данными, в литературе?

Процесс поиска, который состоит из построения критериев поиска и процесса ручного выбора, направлен на выявление литературы, которая включает в себя различные определения больших данных, а также существующих исследований, в которых пытались изучить эти определения.

4.3.1. Анализ и интерпретация

В этом разделе анализируются и интерпретируются результаты, полученные в процессе исследования, чтобы уточнить определение больших данных и определить различные аспекты больших данных.

Опираясь на анализ неисчерпывающего списка существующих определений больших данных и результатов предыдущих работ, в которых пытались объяснить определение термина «большие данные», авторы заметили, что термин «большие данные» определялся с разных точек зрения. Некоторые определения были предложены академическими исследователями,

а другие - соответствующими компаниями, такими как IBM, Microsoft, Oracle, Hadoop и т. д. С разных точек зрения существовали различия между этими определениями, которые были схожи в одних элементах и различались в других. Иногда в одной и той же статье приходилось давать разные определения термину «большие данные», чтобы дать четкое и всеобъемлющее определение. В этом контексте в [4.23] заявилено, что достичь консенсуса по определению больших данных практически невозможно, поэтому использование множества определений для определения термина «большие данные» был логичным выбор, при котором каждое определение фокусируется на определенном аспекте.

4.3.2. Понятия термина «Большие данные»

Результат исследования показывает, что термин «большие данные» часто ассоциируется с шестью понятиями, каждое из которых включает хотя бы одно из них. Это наборы данных с новыми характеристиками, жизненный цикл анализа данных, технологии, аналитические методы, понимание (ценность) и влияние (принятие решений).

- **Датасеты с новыми характеристиками:** категория определений подчеркнула новые характеристики данных по сравнению с традиционными данными. Три основных измерения больших данных: объем, скорость и разнообразие [4.59]. Модель 3Vs была расширена до моделей 4Vs, 5Vs, 6Vs и 7Vs. IBM предложила модель 4Vs, добавив достоверность в качестве четвертой V больших данных; модель 3V была расширена до модели 5Vs за счет повышения ценности и достоверности; другое исследование расширило модель 3V, добавив достоверность, изменчивость и визуализацию для определения модели 6Bc; хотя существуют две альтернативы модели 7V: 3V с достоверностью, достоверностью, волатильностью и ценностью; а вторая модель 7V включает модели 3V с ценностью, достоверностью, изменчивостью и сложностью [4.15]. Модель 3V была широко при-

нята в качестве определения больших данных, тогда как расширенные модели вызвали споры, поскольку они вызывают путаницу между атрибутами больших данных и другими элементами данных, которые связаны с качеством данных и управлением ими и не являются элементами, однозначно описывающими большие данные. 5Vs, которые включают в себя правдивость и ценность с объемом, разнообразием и скоростью, являются наиболее часто используемым расширением в литературе [4.15] В этом исследовании используется подход 4 Vs, предложенный IBM, который включает в себя: объем, разнообразие, скорость и достоверность, при этом ценность не рассматривается как характеристика больших данных. Ниже мы даем определение этим элементам.

◦ **Объем:** Характеристика, которая больше всего ассоциируется с большими данными и отличает большие данные от традиционных данных, связана с большим объемом данных, генерируемых каждый день машинными и человеческими ресурсами [4.49]. По сообщению IBM [4.24] каждый день создается 2,5 квинтиллиона данных.

◦ **Скорость:** учитывает скорость производства, обработки и анализа данных [4.49]. Кроме того, обработка и анализ данных требуют получения результатов в реальном времени.

◦ **Разнообразие:** основное внимание уделяется разнообразию источников данных и разнообразию создаваемых типов данных [4.49]. Организациям необходимо обрабатывать различные типы данных, включая структурированные данные, такие как традиционные системы баз данных; полуструктурированные, такие как данные XML; и неструктурированные данные, такие как текст, изображения, веб-данные, данные датчиков, аудио, видео и т. д. В этом контексте IBM сообщила, что 90% сгенерированных данных представляют собой неструктурированные данные и что 80% данных поступают в формате видео, изображений и документов [4.24].

◦ **Правдивость:** основное внимание уделяется качеству данных, свя-

занных с определенными типами источников. Качество данных связано со степенью достоверности этих данных. Организации принимают свои решения на основе результатов анализа некоторых неопределенных и неточных данных, таких как настроения и правдивость людей, погодные условия и экономические факторы [4.49]. Несмотря на неопределенность данных, их нельзя игнорировать, поскольку они по-прежнему содержат ценную информацию.

• **Жизненный цикл аналитики данных:** некоторые определения связывают большие данные с процессом сбора, хранения, обработки и анализа данных с целью извлечения пользы из этих данных. В этом смысле в исследовании [4.58] говорится, что большие данные в основном связаны с двумя идеями: хранением данных и анализом данных.

• **Технологии:** другая категория определений подчеркивала необходимость в новых технологических инструментах, которые позволяют собирать, хранить, обрабатывать и анализировать эти объемы данных, поскольку традиционные технологии управления данными не могут справиться с характеристиками больших данных.

• **Аналитические методы**: существует категория определений, в которых основное внимание уделяется тому, что анализ этих наборов данных требует применения мощных аналитических методов для извлечения информации из структурированных и неструктурированных данных.

• **Понимание (значение):** некоторые определения сосредоточены на скрытой ценности этих больших объемов самых разнообразных данных, которая в некоторых исследованиях рассматривается как пятая часть больших данных. Ценность указывает на то, что в таком огромном объеме данных скрыто много потенциальной и очень полезной информации. По [4.49] это значение связано с использованием данных и не является одной из характеристик, характеризующих эти данные, поскольку это значение необходимо извлечь из этих данных с помощью анализа. Поэтому предло-

жено разделить данные и их использование, чтобы устранить двусмысленность и непоследовательность определения. Также предложено использовать термин «понимание больших данных» для описания ценности больших данных. В исследовании принята идея, согласно которой ценность не является характеристикой данных. Ценность является фундаментальной особенностью, которая придает большим данным все большее значение [4.43]. Таким образом, ценность или понимание являются неотъемлемой частью концепций, определяющих термин «большие данные».

• **Влияние (принятие решений):** Исследование [4.56] в качестве цели работы с Big Data считает использование знаний из данных для научных целей или для работы с Big Data в реальном времени для ыормирования близких к оптимальным решений.

4.4. Модель BD-DA: большие данные с моделью решений

4.4.1. Уровни модели BD-Da

Построенная модель BD-Da представляет собой концептуальную модель, которая иллюстрирует интеграцию аспекта принятия решений в большие данные. Опираясь на шесть концепций, связанных с определениями больших данных, модель BD-Da выделяет три уровня больших данных, которые необходимо учитывать при разработке проекта больших данных, направленного на поддержку принятия решений в организациях. Это уровень данных, уровень анализа данных и уровень принятия решений, как они изображены на 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.

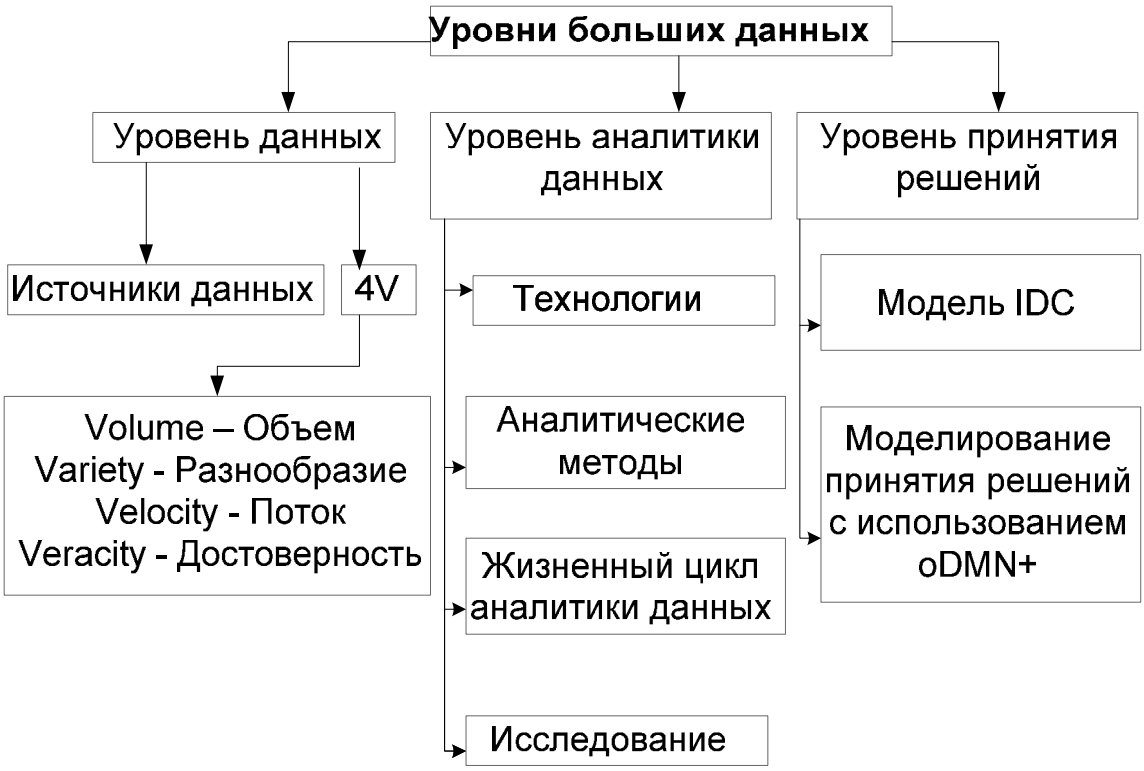


Рис. 4.1. Уровни модели BD-Da

• **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.

• **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+.

4.4.2. Модель BD-Da: детализация

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. BD-Da делит эти концепции на три уровня, которые следует учитывать для разработки успешных проектов больших данных, а именно: уровень данных, уровень анализа данных и уровень принятия решений. Ниже объясняется каждый компонент предлагаемой нами модели.

Проблема решения

В [4.11] отмечено, что запуск проекта по аналитике больших данных, который начинается со сбора и анализа данных, не уделяя достаточного времени для понимания целей и требований проекта, а также формулирования правильной бизнес-задачи, является распространенной ошибкой, которую совершают в проектах аналитики больших данных. Эта ошибка приводит к созданию ценности без каких-либо последствий и недостижению своих целей из-за несоответствия цели и имеющихся данных или неправильного понимания целей проекта. Успешный проект по работе с большими данными начинается с понимания предметной области бизнеса и формулирования бизнес-проблемы (что соответствует предлагаемому этапу открытия их жизненного цикла). Формулирование бизнес-задачи проекта означает определение основных целей, потребностей, которые должны быть достигнуты с точки зрения бизнеса, и требований для удовлетворения этих потребностей [4.11].

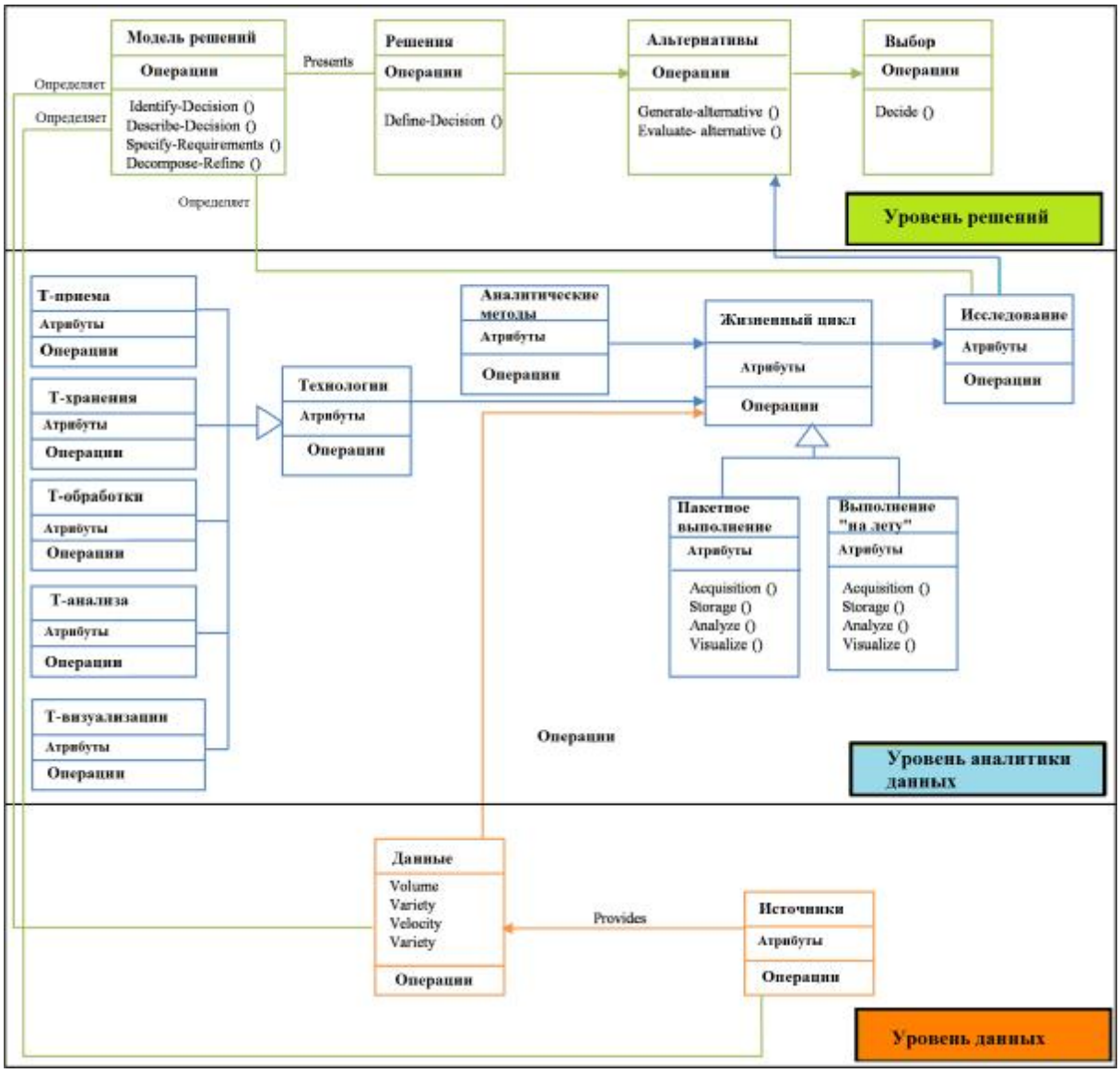


Рис. 4.2. Детализация модели BD-Da: T-* - технологии *

Для проекта аналитики больших данных, целью которого является поддержка принятия решений внутри организации, нужно связать постановку проблемы проекта с определением решений, которые будут приняты, и их требований (соответствует фазе разведки модели IDC). Процесс интеграции из задач, в которых использование Big Data осуществляется фактически, и есть в результате проект поддержки принятия решений.

Стартовые компоненты для начала процесса принятия решений есть дефиниция необходимых знаний и данных. Эти знания и данные в процессе анализа данных порождают данные для модели. В этом контексте пред-

лагается моделирование этих решений, чтобы лучше понять эти решения и их требования на основе данных и анализа данных.

Модель принятия решений

Чтобы четко определить, какое решение будет принято, и указать его требования, можно использовать стандарт DMN для моделирования решений и их требований. Цель использования DMN на этом этапе - представить решения ясным, простым и недвусмысленным образом, а также облегчить общение и сотрудничество между лицами, принимающими решения, и командой аналитиков.

Напомним, что моделирование требований к принятию решений может использоваться для моделирования стратегических, тактических и оперативных решений с целью описания принятия решений человеком, полуавтоматизации или автоматизации решений. В этом контексте для запрограммированного или полузапрограммированного решения, цель которого состоит в полуавтоматизации или автоматизации этого решения, можно использовать логику принятия решения для описания DRD. Для непрограммируемых решений, таких как стратегические решения, которые обычно не запрограммированы по своей природе, можно разработать DRD и описать его, используя естественный язык или семантику бизнес-словаря и правил (SBVR), а не логику принятия решений.

SVBR - это стандарт, опубликованный OMG. Его цель состоит в том, чтобы сформулировать бизнес-словарь, позволяющий преодолеть двусмысленность бизнес-терминов, который затем можно будет применить в бизнес-правилах, чтобы они были однозначно и ясно сформулированы. Формулировка правил в SBVR соответствует шаблону естественного языка и предназначена больше для бизнеса, чем для ИТ, что способствует трансформации бизнеса независимо от конструкции ИС [4.21].

Моделирование процесса принятия решений с использованием DMN

позволяет поддерживать сотрудничество между лицами, принимающими решения, и аналитической командой больших данных, а также благодаря понятным общим обозначениям, которые облегчают общение и сотрудничество между лицами, принимающими решения, и командой аналитиков, а также позволяют хорошо понимать данные и желаемые знания в области аналитики больших данных. Однако раньше не удалось представить связь между информацией и источниками данных, которые ее предоставили. А дополнение, предоставляемое oDMN+, позволяет восполнить этот пробел. Расширенная модель oDMN+, основанная на стандарте DMN, позволяет моделировать источники данных. В этом контексте авторы предлагают использовать расширенную модель и обозначение oDMN+ для моделирования решения, а не модель DMN, поскольку, помимо преимуществ, предоставляемых DMN, дополнение, обеспечиваемое oDMN+, позволяет определять и представлять потенциальные альтернативы источников, предоставляющих данные.

Для разработки модели принятия решений с использованием oDMN+ определим четыре шага. Эти шаги аналогичны шагам, предложенным в [4.53] для построения модели принятия решений с использованием стандарта DMN. Использование oDMN+ требует добавления некоторых необходимых модификаций для адаптации к изменениям oDMN+.

В [4.53] объясняются четыре итерационных шага по разработке эффективной модели требований к принятию решений с использованием стандарта DMN. Эти шаги:

- **Идентифицировать решения**, которые являются центром интереса этого проекта.
- **Описать решения**: название, краткое описание каждого решения и то, как улучшение этих решений повлияет на бизнес.
- **Указать требования к решению**: Указать требования решения (информация и знания) и объединить их в диаграмму требований решения.

На этом этапе также необходимо определить различные источники данных, которые могут предоставить необходимую входную информацию, поскольку эти источники данных будут моделироваться с использованием oDMN+.

- **Декомпозиция и уточнение модели:** Если для принятия решения требуется информация, полученная из других решений, необходимо определить необходимые дополнительные решения, описать их и указать их требования.

Источники данных

В настоящее время мы живем в эпоху, когда данные интенсивно и непрерывно проникают во все сферы нашей деятельности. Эти данные могут генерироваться людьми, например, электронные письма, исследования, документы, файлы журналов и разнообразные данные, генерируемые внешеней средой [4.1].

Таким образом, современные организации имеют многочисленные внешние источники, которые потенциально могут предоставить новый вид информации. Модель разработчика, использующая oDMN+, определяет другой источник, который может предоставить данные, необходимые для проекта.

Данные с новыми характеристиками

Данные являются ключевым элементом, связанным с концепцией больших данных. Это сырье, используемое при производстве деловой информации. Эти данные постоянно создаются из нескольких источников. Он отличается от традиционных данных своими характеристиками, известными как 4V: объем, разнообразие, скорость и достоверность. Модель принятия решений, разработанная с использованием oDMN+, определяет необходимые данные, которые необходимо собрать и проанализировать в

нашем проекте.

Жизненный цикл аналитики данных

Организациям необходимо использовать информацию, полученную из больших данных, для улучшения процесса принятия решений. Чтобы преобразовать необработанные данные, определенные моделью принятия решений на уровне принятия решений, в информацию, которую могут использовать лица, принимающие решения, эти данные, находящиеся в нескольких источниках, должны пройти через последовательность этапов, известную как жизненный цикл больших данных. Эти данные необходимо собирать, хранить, обрабатывать и анализировать, чтобы превратить их в полезную и ценную информацию. Затем эту информацию необходимо представить в структурированном виде и показать лицам, принимающим решения, чтобы они могли использовать их для принятия более эффективных решений. В целом существует две модели обработки больших данных, а именно пакетная обработка и потоковая обработка [4.29]. Выбор между ними зависит от требуемой задержки.

- **Пакетная обработка:** предназначен для решения проблемы больших данных. Он работает с данными, которые уже хранились в системе хранения в течение определенного периода времени, игнорируя новые данные, созданные после начала пакетной обработки. Это зависит от параллельной распределенной системы обработки. Он обеспечивает стабильность и надежность, однако имеет высокую задержку, поэтому не подходит для приложений реального времени [4.7].
- **Потоковая обработка или обработка в реальном времени:** ориентирован на обработку потоковых данных с высокой скоростью, используя бездисковый подход для достижения низкой задержки. Потоковая обработка позволяет собирать и анализировать данные по мере их создания Она хорошо работает с приложениями, которым требуется обработка

потоков данных из разнородных источников в реальном времени [4.7].

Приведем этапы жизненного цикла процессов анадитики Big Data [4.18]:

- Сбор данных (загрузка);
- Хранение данных;
- Анализ данных;
- Использование и визуализация данных;

Технологии

Характеристики больших данных перевешивают способность классических аппаратных сред и программных инструментов обрабатывать такие большие объемы, скорость и разнообразие данных. Действительно, традиционным технологиям и платформам часто данных не хватает емкости, масштабируемости, гибкости и производительности, потребных для работы Big Data [4.41]. Для хранения, обработки и анализа такого объема структурированных и неструктурированных данных было разработано множество инструментов и технологий, обеспечивающих большую гибкость, масштабируемость и производительность [4.41]. Ниже приведены примеры доступных технологий больших данных. Они разделены на пять категорий: инструменты хранения данных, инструменты обработки данных, инструменты приема данных, инструменты анализа данных и инструменты визуализации данных.

- **Инструменты хранения данных:** файловая система, такая как Hadoop, системы баз данных NoSQL с широкими столбцами, такие как Hbase и Cassandra, системы NoSQL для хранения документов, такие как CouchDB и MongoDB, системы NoSQL для хранения ключей и значений, такие как Redis, и системы NoSQL Graph СУБД, такие как Neo4j.
- **Инструменты обработки данных:** инструменты пакетной обработки, такие как Hadoop MapReduce, Spark, Pig, Hive и т. д. Инструменты потоковой обработки, такие как Flink, потоковая передача Spark, Strom,

Samza и т. д.

- **Инструменты приема данных:** Инструменты пакетной обработки, такие как Sqoop и Chukwa. И инструменты потоковой обработки, такие как Flume, Nifi и Kafka.
- **Инструменты анализа данных:** инструменты пакетной обработки, такие как Mahout, H2O и Spark MLlib. И инструменты потоковой обработки, такие как SAMOA.
- **Инструменты визуализации данных:** инструменты пакетной обработки, такие как Tableau и Pentaho. И инструменты потоковой обработки, такие как Zoomdata.

Аналитические методы

Ценность больших данных невозможно получить, применяя традиционные аналитические методы, используемые с небольшими наборами реляционных данных. Чтобы преодолеть ограничения традиционных методов и эффективно анализировать большие данные, необходимо разработать новые аналитические методы для анализа новых типов данных, а традиционные методы необходимо адаптировать к объему, скорости и разнообразию данных.

Эти методы анализа данных включают в себя множество проблем, которые обычно мешают друг другу [4.56]. Согласно [4.1], основными методами анализа больших данных являются методы:

1) машинного обучения (классификация, кластеризация, регрессия, анализ ассоциаций, анализ графов и дерево решений);

2) статистического анализа;

3) интеллектуального анализа данных. Эти методы применяются для анализа данных с новыми характеристиками. Они применимы к текстовой аналитике, аудиоаналитике, видеоаналитике, аналитике социальных сетей и т.д. [4.1].

Понимание (ценность)

Организации изо всех сил пытаются обработать большие объемы, обеспечить скорость и учесть разнообразие больших данных, чтобы получить скрытую значимую ценность больших данных [4.43]. Эта ценность заключается в знаниях, извлеченных из больших данных путем обработки и анализа необработанных данных. Это понимание потенциально может улучшить процесс принятия решений в организации, чтобы принимать более быстрые и разумные решения. Требуемая от проекта больших данных информация определяется на уровне принятия модели решения, в то время как уровень анализа данных отвечает за сбор этой информации из необработанных данных на уровне данных.

Так, в [4.48] установлено вероятное число больных астмой, которые потенциально обратятся за неотложной помощью в территориальном ограничении, спомощью анализа данных отделения неотложной помощи, социальных сетей, Google и данных датчиков окружающей среды. Другое исследование [4.60] позволило выявить потенциальных покупателей роскошных автомобилей с использованием данных владельцев автомобилей и пользователей телекоммуникаций. В то время как исследование [4.38] позволило предсказать тенденции на индийском фондовом рынке ежедневно и ежемесячно посредством обработки новостей, данных социальных сетей и исторических цен. В другом исследовании [4.50] использовались данные социальных сетей, собранные из Твиттера, чтобы определить подлинность и полярность настроений по отношению к бренду «Starbucks». В исследовании [4.7] также использовались данные Твиттера, чтобы классифицировать пользователей в соответствии с их политической ориентацией (демократы или республиканцы) на основе политического содержания в твитах.

Альтернативы и выбор

Создание, разработка и анализ альтернативных вариантов действий для решения проблемы принятия решения являются важным шагом в решении проблемы принятия решения (соответствует этапу проектирования модели Саймона). Результаты, полученные на уровне аналитики больших данных, обеспечивают решение этой задачи. Лица, принимающие решения, полагаются на новую информацию, свои знания и опыт, чтобы предложить альтернативные варианты действий. Кроме того, определены критерии, которые будут использоваться при оценке каждой альтернативы.

После определения альтернатив и критериев оценки эти альтернативы оцениваются, чтобы определить влияние, которое каждая из них окажет на бизнес. Для оценки альтернатив в справочнике [4.45] предлагается внедрение DSS для прогнозирования наиболее адекватных решений среди предложенных альтернатив. Исследование [4.13]показало, что для оценки этих альтернатив можно использовать визуализацию, отчетность, информационные доски, сценарии «что, если» и исследовательские методы. А исследование [4.55] предложило использовать многокритериальные методы для оценки альтернатив и выбора наиболее подходящей из них.

Выбор состоит в том, чтобы принять фактическое решение; на основе результатов оценки лица, принимающие решения, выбирают альтернативу или набор альтернатив, которые решат проблему, подлежащую реализации.

4.5. Анализ и особенности применимости

Значение исследовательской работы обусловлено важностью влияния больших данных на принятие решений в организациях. Принятие адекватных и разумных решений является жизненно важной частью успеха любой организации. Поскольку большие данные обладают потенциалом для улучшения качества принятия решений внутри организации, необхо-

димо признать концепцию больших данных и правильно применять большие данные внутри организации для улучшения качества принимаемых решений. Таким образом, исследование важно для практики и практической работы в области принятия решений на основе Big Data в организациях. В частности:

1. Рассмотрен набор определений Big Data и концепций, которые определяют концепцию Big Data с точки зрения принятия решений.

2. Устновлено, что для получения хорошего эффекта от Big Data, необходимо, четкое понимание не только принимаемых решений, но и требований к данным и знаниям, позволяет сформулировать правильную бизнес-задачу. Это понимание дает проекту Big Data большие преимущества, в решении задачи обеспечении положительного эффекта для ЛПР.

3. Описан логический подход, основанный на требованиях, который объясняет, как компания разрабатывает проект анализа больших данных для поддержки принятия решений. Предлагаемый нами подход начинается с поиска, идентификации и формулирования проблемы, по которой необходимо принять решение, затем выявляются различные требования к данным и знаниям, которые могут поддержать эти решения, и моделируются с использованием oDMN+. После этого все необходимые данные собираются из внутренних и внешних источников, сохраняются и управляются в соответствующей системе хранения больших данных (например, HDFS или NoSQL), обрабатываются и анализируются для получения новой информации. Затем на основе полученной информации разрабатываются и оцениваются альтернативные решения проблемы. Этот подход имеет важные практические последствия.

Во-первых, необходимо наладить сотрудничество между лицами, принимающими решения, и аналитиками в рамках процессов принятия организационных решений, чтобы предотвратить недопонимание и поддержать эффективное использование больших данных при принятии решений.

Предлагаемая модель, основанная на моделировании решений, является возможным решением в этом отношении. Применение предложенного подхода к разработке проекта больших данных, направленного на поддержку принятия решений, может устранить непонимание между экспертами в предметной области (менеджерами) и аналитиками. Моделирование решений поддерживает сотрудничество между лицами, принимающими решения, и командой аналитиков благодаря понятным общим обозначениям, которые облегчают общение и сотрудничество между ними.

Кроме того, существует еще один подход, радикально отличающийся от предложенного.

Он основан на простом сборе большого количества данных и последующем поиске закономерностей, которые могли бы дать полезную информацию, которая может поддержать решения - эти идеи ранее были неизвестны. Например, предлагаемая структура «B-DAD» зависит от этого подхода [4.13]. Платформа B-DAD основана на сборе всех видов доступных данных, их обработке, организации и анализе для извлечения любой информации, которую можно извлечь из них. Затем лица, принимающие решения, видят, как эта информация может поддержать их решения, которые могли быть неизвестны заранее [4.13]. С одной стороны, некоторые считают этот подход большим обещанием и инновацией в области анализа больших данных. С другой стороны, сбор большого количества данных из нескольких доступных источников данных, которые, как правило, знают все, не сосредотачиваясь на выявлении полезных данных, которые действительно необходимы для обеспечения необходимой бизнес-информации для достижения их целей, крайне нежелателен, поскольку эта стратегия скорее всего, потерпит неудачу [4.34]. Важно начинать проект больших данных с поиска, идентификации и формулирования правильного проблемного вопроса, на который проект больших данных пытался ответить. В этом контексте наш подход, основанный на требованиях, основанный на

oDMN+, позволяет компаниям лучше использовать большие данные для предоставления бизнес-информации в режиме реального времени, чтобы помочь принимать адекватные решения с оптимизацией времени и стоимости применения проекта больших данных. Разработанная модель четко определяет данные и источники данных, которые будут использоваться, вопросы, на которые следует ответить, политику и правила, которые будут применяться, а также аналитику данных, которая будет разработана для ответа на эти вопросы. В результате компании могут гораздо легче определить необходимые данные и оставшиеся без ответа вопросы в своем бизнесе, чтобы решить их с помощью больших данных, вместо того, чтобы сосредотачиваться на всех данных. Сбор, управление и обработка большого количества данных становятся все более сложной проблемой в эпоху больших данных и могут дать множество ответов на вопросы, которые на самом деле не имеют значения [4.34].

Все предыдущие исследования, которые интересовались интеграцией больших данных и принятия решений, включая предложенную нами модель, начинаются с выявления проблемы и формулирования проблемы, вызывающей принятие решений. На этом этапе важно различать симптомы проблемы и саму проблему. Иногда то, что считается проблемой (например, снижение продаж в прошлом месяце), может быть не проблемой, а симптомами проблемы, которые не всегда легко отличить от реальной проблемы [4.12]. Определение проблемы можно определить путем мониторинга и анализа внутренних и внешних данных. В этой области выгодно следовать подходу, который основан на сборе всех соответствующих данных для выявления любой информации, которая могла бы определить реальную проблему, для выявления проблемы, требующей решения. Затем начинаем применять предложенный подход, основанный на требованиях, которые включают моделирование решений с использованием oDMN+.

Малые данные будут по-прежнему оставаться важной частью иссле-

довательской среды. Они имеют долгую историю развития в частных и государственных организациях и предприятиях, с устоявшимися методологиями и способами анализа, а также с опытом получения содержательных ответов [4.28]. В этом контексте, хотя исследование сосредоточено на больших данных, подход, описанный выше, также применим и к небольшим данным, поскольку он предоставляет компоненты, которые полезны для общего подхода, который определяет данные как основу для принятия решений, включая небольшие данные, которые также могут извлечь выгоду из технологической эволюции, возникшей в результате появления больших данных.

4.6. Выводы к главе 4

Большие данные стали горячей темой, которая все больше привлекает внимание исследователей во многих дисциплинах благодаря ценным знаниям и практическим идеям, которые могут быть получены. Эта ценность потенциально может улучшить процесс принятия решений в организации, чтобы принимать более быстрые и разумные решения. Поэтому сегодня организации считают, что им необходимо использовать большие данные, чтобы сохранить свое конкурентное преимущество.

Несмотря на важность больших данных в этих организациях, согласно многим исследованиям, существует большое количество проектов по большим данным, которые не достигли желаемых целей. Причин провала этих проектов много. Мы можем суммировать основные причины неудачи этих проектов в том, как эти проекты разрабатывались, поскольку они игнорировали аспект принятия решений. Основная цель этой работы - разработать концептуальную модель, которая интегрирует аспект принятия решений в большие данные, что позволяет связать извлеченную ценность и ее фактическое использование. Для разработки этой модели:

• Для понимания концепции больших данных применен подход SLR.

В результате определены шесть основных понятий, связанных с определениями больших данных, а именно: наборы данных с новыми характеристиками, жизненный цикл анализа данных, технологии, аналитические методы, понимание и принятие решений.

• Определено три уровня Big Data, которые необходимо применять при создании проекта Big Data в области принятия решений: данных, анализа данных и принятия решений.

• Разработана концептуальная модель BD-Da. Она описывает три уровня больших данных, составляющие каждого уровня и отношения между ними.

Исследование дает два основных вклада: первый разъясняет и определяет концепцию больших данных, а второй заключается в изучении возможности интеграции больших данных и процесса принятия решений для поддержки лиц, принимающих решения, для принятия более эффективных и быстрых решений на основе данных.

В конечном итоге следует отметить ограничения данной работы. Модель представляет собой теоретическую модель, основанную на теоретическом исследовании без применения предложенной модели для решения реальной проблемы принятия решений в среде больших данных. Однако в исследовании выделены три точки зрения, которые следует проанализировать для разработки проекта больших данных.

Литература к главе 4

4.1. Aggi, M. K., & Jain, S. (2018). Survey towards an integration of big data analytics to big insights for valuecreation// Information Processing & Management, 54(5), 758–790. doi:10.1016/j.ipm.2018.01.010

4.2. Akter, S., Bandara, R., Hani, U., Wamba, S. F., Foropon, C., & Papadopoulos, T. (2019). Analytics-based decision-making for service systems: A qualitative study and agenda for future research// International Journal of Information Management, 48, 85–95. doi:10.1016/j.ijinfomgt.2019.01.020

4.3. Arndt, H. (2018). Knowledge discovery and anomalies - towards a dynamic decision-making model for medical informatics [Doctoral dissertation]. Stellenbosch University.

4.4. Brynjolfsson, E., Hitt, L. M., & Kim, H. H. (2011). Strength in numbers: How does data-driven decisionmaking affect firm performance?

4.5. Casado, R., & Younas, M. (2015). Emerging trends and technologies in big data processing// Concurrency and Computation, 27(8), 2078–2091.

4.6. Chiheb, F., Boumahdi, F., & Bouarfa, H. (2019). A New Model for Integrating Big Data into Phases of Decision-Making Process// Procedia Computer Science, 151, 636–642. doi:10.1016/j.procs.2019.04.085.

4.7. Colleoni, E., Rozza, A., & Arvidsson, A. (2014). Echo chamber or public sphere? Predicting political orientation and measuring political homophily in Twitter using big data// Journal of Communication, 64(2), 317–332. doi:10.1111/jcom.12084.

4.8. Courtney, J. F. (2001). Decision making and knowledge management in inquiring organizations: Toward a new decision-making paradigm for DSS// Decision Support Systems, 31(1), 17–38. doi:10.1016/S0167-9236(00)00117-2.

4.9. De Mauro, A., Greco, M., & Grimaldi, M. (2015). What is big data? A consensual definition and a review of key research topics// A. I. P. Dans (Ed.), AIP conference proceedings (pp. 97–104). Academic Press. doi:10.1063/1.4907823

4.10. Decker, G., & Debevoise, T. (2015, April). Quick Guide to Decision Modeling using DMN 1.0. Signavio, Inc.

4.11. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. John Wiley & Sons, Inc.

4.12. Efraim, T. (2011). Decision decision-making systems, modeling, and support. Pearson Education India.

4.13. Elgendy, N., & Elragal, A. (2016). Big data analytics in support of the decision making process// Procedia Computer Science, 100, 1071–1084. doi:10.1016/j.procs.2016.09.251.

4.14. Elsevier. (2017). Scopus Content Coverage Guide.

4.15. Emmanuel, I., & Stanier, C. (2016). Defining big data. Proceedings of the International Conference on Big Data and Advanced Wireless Technologies. Academic Press. doi:10.1145/3010089.3010090.

4.16. Forest, H., Foo, E., Rose, D., & Berenzon, D. (2014). Big Data, how it can become a differentiator. Deutsche Bank.

4.17. Forman, E. H., & Selly, M. A. (2001). Introduction: Management Decision-Making Today// Decision by objectives: how to convince others that you are right (p. 1). Scientific, World.

4.18. Google Cloud. (2018, October 17). Data Lifecycle. Retrieved from https://cloud.google.com/solutions/datalifecycle-cloud-platform.

4.19. Gorry, G. A., & Scott Morton, M. S. (1971). A framework for management information systems.

4.20. Grover, V., Chiang, R. H., Liang, T.-P., & Zhang, D. (2018). Creating Strategic Business Value from Big Data Analytics: A Research Framework. Journal of Management Information Systems, 35(2), 388–423. doi:10.1080/07421222.2018.1451951.

4.21. Hall, O., & Odd, S. (2019). Business Decisions or Rules – Why not Both? The Views of Three Decision Modelling Experts// International Journal of Information System Modeling and Design. Volume 10. Issue 4.

4.22. Horita, F. E., Albuquerque, J. P., Marchezini, V., & Mendiondo, E. M. (2017). Bridging the gap between decisionmaking and emerging big data sources: An application of a model-based framework to disaster management in Brazil// Decision Support Systems, 97.

4.23. Hu, H., Wen, Y., Chua, T.-S., & Li, X. (2014). Toward scalable systems for big data analytics: A technology tutorial. IEEE Access : Practical Innovations, Open Solutions.

4.24. IBM. Extracting business value from the 4 V's of big data. Retrieved from https://www.ibmbigdatahub.com/infographic/extracting-business-value-4-vs-big-data.

4.25. Infochimps. (2012). CIOs & Big Data: What Your IT Team Wants You to Know. http://www.infochimps.com/resources/report-cios-big-data-what-your-it-team-wants-you-to-know-6/.

4.26. Janeiro, J., & Eduardsen, J. S. (2018). How can big data affect uncertainty in strategic decision-making? Aalborg University.

4.27. Kitchenham, B. (2004). Procedures for performing systematic reviews. Keele, UK: Keele University.

4.28. Kitchin, R., & Lauriault, T. P. (2015). Small data in the era of big data// GeoJournal, 80(4), 463–475. doi:10.1007/s10708-014-9601-7

4.29. Kolomvatsos, K., & Hadjiefthymiades, S. (2015). An efficient time optimized scheme for progressive analytics in big data// Big Data Research, 2(4), 155–165. doi:10.1016/j.bdr.2015.02.001.

4.30. Kościelniak, H., & Puto, A. (2015). BIG DATA in decision making processes of enterprises// Procedia Computer Science, 65, 1052–1058. doi:10.1016/j.procs.2015.09.053

4.31. Li, T. Z., Wang, S. H., & Ma, J. (2014). Study on Fair Definitions and Application Modes of Big Data// Applied Mechanics and Materials.

4.32. Litherland, N. (2017, September 26). Decision-Making Process of Managers. Recupere sur bizfluent: https://bizfluent.com/how-does-5280248-decisionmaking-process-managers.html.

4.33. Lunenburg, F. C. (2010). The decision making process// National Forum of Educational Administration and Supervision Journal, 27(4), 12.

4.34. Marr, B. (2015). Big Data: Too Many Answers, Not Enough Questions. Forbes.

4.35. Martin, T. N. (2016). Smart Decisions: The Art of Strategic Thinking for the Decision Making Process. Springer. doi:10.1057/9781137537003.

4.36. Mintzberg, H., Raisinghani, D., & Theoret, A. (1976). The structure of" unstructured" decision processes// Administrative Science Quarterly, 21(2), 246–275. doi:10.2307/2392045.

4.37. Montana, P. J., & Charnov, B. H. (2000). Management Decision-Making: Types and Styles. In P. J. Montana, & B. H. Charnov (Eds.), Business Review Books Management Third Edition (pp. 86-105).

4.38. Nayak, A., Pai, M. M., & Pai, R. M. (2016). Prediction models for Indian stock market// Procedia Computer Science, 89, 441–449. doi:10.1016/j.procs.2016.06.096

4.39. Negulescu, O. (2014). Using a decision-making process model in strategic management. Review of General Management.

4.40. OMG. (2016, May). Decision Model and Notation (DMN) V1.1 with change bars.

4.41. Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., & Belfkih, S. (2017). Big Data technologies: A survey// Journal of King Saud University-Computer and Information Sciences, 30(4), 431-448.

4.42. Panneerselvam, J., Liu, L., & Hill, R. (2015). An Introduction to Big Data// B. Akhgar, G. B. Saathoff, H. R. Arabnia et al. (Eds.), Application of Big Data for National Security (pp. 3-13). Elsevier. doi:10.1016/B978-0-12-801967-2.00001-X.

4.43. Panneerselvam, J., Liu, L., & Hill, R. (2015). An Introduction to Big Data// Application of Big Data for National Security (pp. 3–13). Elsevier. doi:10.1016/B978-0-12-801967-2.00001-X

4.44. Parker, J. S., & Moseley, J. D. (2008). Kepner-Tregoe decision analysis as a tool to aid route selection Part 1// Organic Process Research & Development, 12(6), 1041-1043.

4.45. Poleto, T., de Carvalho, V. D., & Costa, A. P. (2017). The Full Knowledge of Big Data in the Integration of InterOrganizational Information: An Approach Focused on Decision Making// International Journal of Decision Support System Technology, 9(1), 16–31. doi:10.4018/IJDSST.2017010102.

4.46. Poleto, T., de Carvalho, V. D. H., & Costa, A. P. C. S. (2015, May). The roles of big data in the decision-support process: an empirical investigation// Proceedings of the International conference on decision support system technology (pp. 10-21). Cham: Springer.

4.47. Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making// Big Data, 1(1), 51–59. doi:10.1089/big.2013.1508 PMID:27447038.

4.48. Ram, S., Zhang, W., Williams, M., & Pengetnze, Y. (2015). Predicting asthma-related emergency department visits using big data// IEEE Journal of Biomedical and Health Informatics, 19(4), 1216–1223.

4.49. Schroeck, M., Shockley, R., Smart, J., Romero-Morales, D., & Tufano, P. (2012). Analytics: The real-world use of big data// IBM Global Business Services, 12, 1–20.

4.50. Shirdastian, H., Laroche, M., & Richard, M.-O. (2017). Using big data analytics to study brand authenticity sentiments: The case of Starbucks on Twitter// International Journal of Information Management.

4.51. Simon, H. A. (1960). The executive as decision maker// The new science of management decision, 1-7.

4.52. Taylor, J. (2016). Bringing Clarity to Data Science Projects with Decision Modeling: A Case Study. International Institute for Analytics.

4.53. Taylor, J. (2016). Decision Modeling with DMN. Decision Management Solutions.

4.54. Taylor, J. (2017). Framing Analytic Requirements. Decision Management Solutions.

4.55. Turet, J. G., & Costa, A. P. (2018). Big Data Analytics to Improve the Decision-Making Process in Public Safety: A Case Study in Northeast Brazil// Proceedings of the International Conference on Decision Support System Technology. Academic Press. doi:10.1007/978-3-319-90315-6_7.

4.56. Wang, H., Xu, Z., Fujita, H., & Liu, S. (2016). Towards felicitous decision making: An overview on challenges and trends of Big Data// Information Sciences, 367, 747–765. doi:10.1016/j.ins.2016.07.007.

4.57. Ward, J. S. (2013). Undefined by data: a survey of big data definitions.

4.58. Ward, J. S., & Barker, A. (2013). Undefined by data: a survey of big data definitions.

4.59. Ylijoki, O., & Porras, J. (2016). Perspectives to definition of big data: A mapping study and discussion// Journal of Innovation Management, 4(1), 69–91. doi:10.24840/2183-0606_004.001_0006.

4.60. Zhang, H., Zhang, L., Cheng, X., & Chen, W. (2016). A novel precision marketing model based on telecom big data analysis for luxury cars// Proceedings of the 2016 16th International Symposium on Communications and Information Technologies (ISCIT) (pp. 307-311). IEEE.

4.61. Zhou, X., Jin, Y., Zhang, H., Li, S., & Huang, X. (2016). A map of threats to validity of systematic literature reviews in software engineering// Proceedings of the 2016 23rd Asia-Pacific Software Engineering Conference (APSEC) (pp. 153-160). Academic Press.

5. Программные проекты управления большими данными: кластеризация и база знаний

5.1. Интеллектуальный алгоритм кластеризации разнородных больших данных в среде со сложными атрибутами

5.1.1. Повышение стабильности операций интеллектуального анализа гетерогенных больших данных в среде сложных атрибутов

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 классификации 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.

В последние годы, в связи с растущим использованием сетевых ресурсов, различные отрасли промышленности уделяют все больше внимания анализу гетерогенных больших данных, особенно в среде со сложными атрибутами, большие данные имеют множество характерных параметров. Это повлияло на степень использования больших данных пользователем. По этой причине людям необходимо использовать базу данных для осуществления разумного планирования и эффективного анализа разнородных больших данных. Научно-исследовательские учреждения предло-

жили несколько методов интеллектуального анализа разнородных больших данных в сложных атрибутивных средах, но интеллектуальный анализ в сложных атрибутивных средах требует выполнения ряда операций, таких как анализ, очистка, преобразование и интеграция данных. В результате предложенный ранее метод не может обладать высокой точностью, стабильностью и практичностью одновременно при проведении работ [5.1].

Интеллектуальный кластерный анализ гетерогенных больших данных использует технологию моделирования данных для моделирования и анализа внутренней структуры и распределения данных. С точки зрения интеллектуального анализа данных, интеллектуальная кластеризация гетерогенных больших данных - это неконтролируемый алгоритм. При отсутствии предварительных знаний для разделения данных и формирования маркерных кластеров используется алгоритм кластеризации. Исследовательские направления теории кластерного анализа включают в себя следующие аспекты: Во-первых, способность обрабатывать различные типы данных. Большинство существующих алгоритмов применяются для анализа числовых данных, но при практическом применении приходится сталкиваться со многими типами данных. Таким образом, ограниченность алгоритма в возможностях обработки данных препятствует его популяризации и применению. Во-вторых, возможность идентифицировать кластеры данных произвольной формы. Большинство существующих алгоритмов кластеризации используют стандартное евклидово расстояние для выполнения задач измерения сходства, поэтому этот алгоритм, как правило, идентифицирует сферические кластеры. Форма кластеров фактических данных средней и высокой размерности в основном не сферическая, поэтому улучшение способности алгоритма распознавать кластеры произвольной формы является ключом к улучшению эффекта кластеризации.

В сложной атрибутивной среде метода интеллектуального анализа больших данных выбор гетерогенной базы данных особенно важен. По-

этому предлагается метод интеллектуального анализа больших данных СУБД в среде сложных атрибутов. Благодаря очистке пространства параметров среды сложных атрибутов и внедрению распределенной идеи для повышения практичности метода, точность и стабильность интеллектуального анализа разнородных больших данных значительно повышаются.

5.1.2. Разработка алгоритма интеллектуальной кластеризации разнородных больших данных на основе среды со сложными атрибутами

5.1.2.1. Очистка пространств параметров для среды со сложными атрибутами

Целью очистки пространства параметров является соблюдение требований к качеству анализа данных, чтобы полностью гарантировать корректность анализа данных [5.2]. Очистка данных - это обнаружение и исправление поврежденных или ошибочных записей в наборе записей, таблице или базе данных, а затем замена, исправление или удаление выявленных искаженных данных, которые являются неполными, неверными, неточными или не относящимися к делу. Процесс достижения согласованности данных.

На рис. 5.1 показан метод очистки пространства параметров. Первый шаг - это очистка от неточных атрибутов записей в наборах данных пространства параметров и распознавание атрибутов исключений в наборах данных пространства параметров [5.3]. Основная идея заключается в том, чтобы сначала присвоить вес каждому атрибуту, затем подсчитать среднее значение и стандартное отклонение каждого значения поля атрибута, а затем установить доверительный интервал для каждого атрибута. Исходя из того, находится ли значение атрибута в пределах доверительного интервала, можно определить, является ли атрибут ненормальным. Алгоритм кла-

стеризации может судить о том, является ли атрибут ненормальным, по расстоянию между значением атрибута и центром кластера и использовать знания о распознавании образов для поиска аномального атрибута [5.4].

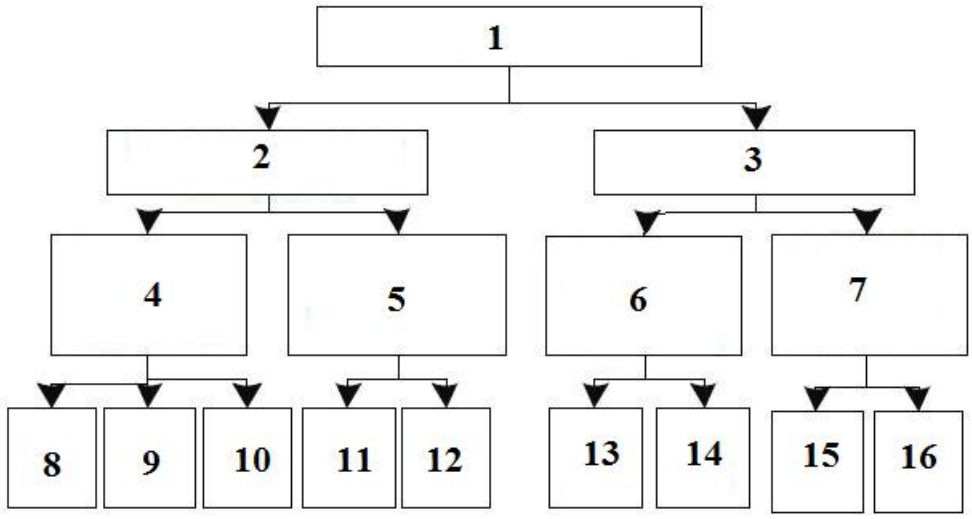


Рис. 5.1. Классификация методов очистки пространства параметров: 1 – очистка пространства параметров; 2 – очистка пространства неточных параметров; 3 – повторяющаяся запись данных; 4 – способ обнаружения ошибок в данных; 5 – очистка неточных данных; 6 – алгоритмы обнаружения повторяющихся данных; 7 – метод очистки повторяющихся данных; 8 – статистические подходы; 9 – метод кластеризации; 10 – метод ассоциативных правил; 11 - метод очистки данных на основе незанятых значений; 12 - очистка данных от шума; 13 - очистка несогласованных данных; 14 - алгоритм последовательного сбора полей данных; 15 - рекурсивный алгоритм сопоставления полей; 16 - сортировка по множеству ближайших соседей

Задача очистки данных рассматривается как задача статистического вывода структурированных текстовых данных в среде сложных атрибутов. Это классический инструмент для представления и обоснования противоречивых знаний. Прежде чем дать определение байесовских сетей, сначала приводятся соответствующие формулы в качестве теоретической основы. Пусть Ω - выборочное пространство эксперимента E, A - событие из E, Ω - разбиение из $p(a)>0$, $b_1, b_2, \ldots, b_n$, $p(b_i)>0$, $i=1, 2, \ldots, n$.

Тогда

$$\overline{p}(b_i \mid a) = \frac{p(b_i \mid a)}{\sum_{j=1}^{n} p(b_j \mid a)} \qquad (5.1)$$

D = T_1, T_2, ..., T_n представляет входные данные структурированных данных, которые содержат искаженные данные. T_IÎ D представляет один или несколько кортежей с искаженными данными для значения атрибута m [2.5]. Учитывая возможный набор для замены C. Кортеж T для возможных "грязных" данных в D, он может очистить базу данных, заменив T_I Î D подходящим кортежем для очистки T с $P_{R(T^*|T)}$.

Используя байесовские правила в сложных атрибутивных средах, при очистке из нескольких источников необходимо учитывать, что каждый источник данных может включать разные поля данных и существуют разные формы данных, поэтому причины получения неточных данных различны. Проблемы с неточными данными в средах с несколькими источниками гетерогенных данных можно резюмировать следующим образом: во-первых, ошибки в данных: ошибки в данных могут быть вызваны неправильным сбором данных или неправильным вводом данных, что приводит к ошибкам различной степени в данных [5.6]. Во-вторых, конфликт имен: конфликт имен возникает, когда одно и то же имя используется для другого объекта или когда для одного и того же объекта используется другое имя. В-третьих, неоднородность данных: разные представления одних и тех же объектов из разных источников, например, разные структуры компонентов, разные типы данных и разные ограничения целостности. В-четвертых, избыточность данных: разные представления данных из разных источников содержат разные ошибки версий. В-пятых, в гетерогенной среде с несколькими источниками, даже если существуют одно и то же имя атрибута и тип данных, в источнике данных могут быть разные представления значений или разные интерпретации.

5.1.2.2. Введение терминов для кластеризации разреженных подпространств в средах со сложными атрибутами

При работе с наборами данных малой размерности традиционные алгоритмы кластеризации пытаются найти кластеры во всех измерениях наборов данных. Но в средах со сложными атрибутами обычно существует множество независимых измерений. Эти независимые измерения будут скрывать существующие кластеры в зашумленных данных и влиять на результаты традиционного алгоритма кластеризации, в то время как реальные корреляционные данные будут распределены по низкоразмерной структуре, которая может представлять характеристики алгоритма кластеризации [5.7]. Кроме того, в наборе данных с очень высокой размерностью объекты данных распределены неравномерно, и все объекты данных практически равны друг другу. Это делает бессмысленным использование одного измерения расстояния и может привести к сбою измерений. Таким образом, в этом проекте введен обычный термин кластеризации разреженных подпространств в среде сложных атрибутов, чтобы исключить нерелевантную и избыточную информацию в наборе данных, и кластеризация выполняется только по связанным измерениям. Поскольку размерность наблюдаемых данных обычно выше, чем их существенная корреляционная размерность, теоретически возможно уменьшить размерность исходного пространства без потери какой-либо информации. На практике метод уменьшения размерности чаще всего используется для уменьшения размерности высокоразмерных данных перед кластеризацией [5.8]. Существует два широко используемых метода уменьшения размерности данных: извлечение признаков и выбор признаков.

На рис. 5.2 показана структурная схема уменьшения размерности высокоразмерных данных.

Рис. 5.2. Структурная схема уменьшения размерности высокоразмерных данных

На первом этапе выделение объектов представляет собой метод предварительной обработки в проекционном пространстве, который позволяет алгоритму кластеризации использовать только небольшое количество вновь выбранных объектов для кластеризации. Извлечение признаков

путем введения обычного термина кластеризации разреженных подпространств в среде сложных атрибутов, создания набора данных линейной комбинации из атрибутов данных, обнаружения потенциальной структуры, генерации и выбора новых векторов признаков, чтобы добиться уменьшения размерности [5.9].

Второй этап заключается в сохранении относительного расстояния между исходными объектами данных без удаления каких-либо исходных атрибутов данных при введении обычного термина кластеризации разреженных подпространств в среде сложных атрибутов, что позволяет сохранить влияние независимых измерений. Поэтому, когда в наборе данных имеется большое количество независимых атрибутов, маскирующих кластеры, извлечение объектов не сможет дать желаемого эффекта.

На третьем этапе выбор объектов - это метод устранения избыточной информации путем анализа всего набора данных. Он выбирает оптимальное подмножество из исходного набора данных путем поиска различных подмножеств признаков и использования некоторых критериев для оценки этих подмножеств. Распространенные стратегии поиска включают случайный поиск, поиск по выборке и жадный последовательный поиск.

На этапе 4 критерии оценки основаны на двух основных моделях: модели оболочки и модели фильтра. Наконец, пятый шаг, в соответствии с большей частью работы по обучению под наблюдением, заключается в выборе меры точности и классификационной метки, чтобы завершить внедрение правил кластеризации разреженных подпространств в среде сложных атрибутов.

5.1.2.3. Настройка индекса интеллектуальной кластеризации разнородных больших данных

Валидность кластеризации гетерогенных больших данных определя-

ется тем, подходит ли данное нечеткое разбиение для всех данных. Индекс валидности кластера гетерогенных больших данных может использоваться для непосредственного измерения качества результатов кластеризации. Хорошие результаты кластеризации должны быть как можно более компактными и располагаться как можно дальше между кластерами [5.10]. В различных источниках предлагаются различные показатели скалярной достоверности, но ни один из них не применим полностью для оценки всех результатов кластеризации. Для оценки результатов кластеризации выбраны пять показателей достоверности.

Первый индикаторный коэффициент, используемый для измерения количества “совпадений” между кластерами и определения его по формуле (5.2):

$$PC = \frac{1}{N} \sum_{I=1}^{C} \sum_{J=1}^{N} U_{IJ} \tag{5.2}$$

В формуле (5.2) U_{IJ} указывает степень, в которой точка данных j относится к категории i, C и N указывают количество кластеров и общее количество выборок данных соответственно.

Второй показатель, классификационная энтропия (CE): измеряет неоднозначность разбиения кластера и определяет ее по формуле (5.3):

$$CE = \frac{1}{N} \sum_{I=1}^{C} \sum_{J=1}^{N} \left(U_{IJ} \log\left(U_{IJ} \right) \right) \tag{5.3}$$

В формулах (5.2), (5.3) показатели PC и CE определяют, насколько понятны результаты кластеризации. Чем больше значение PC, тем компактнее класс, чем меньше значение CE, тем лучше эффект кластеризации [5.11, 5.12].

Третий показатель - коэффициент разделения (SC): это отношение суммы компактности внутри кластера и разделения между кластерами. Это сумма показателей валидности отдельных кластеров, нормализованная путем деления на нечеткую мощность каждого кластера, которая определяет-

ся по формуле (5.4):

$$SC = \sum_{I=1}^{e} \frac{\sum_{J=1}^{n} U_{IJ}}{N_I \sum_{\substack{K=1 \\ K \neq I}}^{C} |V_K - V_I|} \qquad (5.4)$$

В формуле (5.4) N_I представляет выборку j из набора данных, V_K и V_I являются центрами кластеров K и I соответственно. SC может использоваться для измерения качества различных разделов с одинаковым количеством кластеров. Чем меньше значение SC, тем лучше результаты кластеризации.

Четвертый показатель - индекс разделения: в отличие от индекса разделения SC, индекс разделения использует минимальное расстояние между разделами для достижения эффективности разделения. Чем меньше значение S, тем больше расстояние между классами и тем лучше результаты кластеризации.

Пятый показатель – коэффициент XB, предназначенный для количественной оценки отношения общих изменений внутри кластера к разделению кластеров. Размер XB может измерять степень компактности и разделения между кластерами. Чем меньше соответствующее значение, тем компактнее кластер и чем дальше расстояние между кластерами, тем лучше результат кластеризации.

5.1.2.4. Реализация интеллектуальных вычислений для кластеризации разнородных больших данных

Для того чтобы обнаружить и устранить дублирующиеся записи в наборах данных, необходимо решить проблему того, как определить, дублируются ли две записи или нет, и оценить сходство данных, то есть проблему сопоставления данных. Простейший набор атрибутов может быть

получен путем сокращения вышеуказанных атрибутов. В соответствии с простейшим набором атрибутов извлекаются данные из связанных атрибутов, синтезируется таблица данных, а затем таблица данных очищается от аналогичных повторяющихся данных. Таким образом, соответствующие записи из записей сравниваются, и вычисляется сходство.

Идея базового алгоритма о близости сортировки может быть сведена к трем шагам:

Шаг 1 - создать ключ сортировки: вычислить ключ сортировки для каждой записи в наборе данных, извлекая соответствующее поле или часть поля.

Шаг 2 - сортировка данных: сортируется весь набор данных или его часть в соответствии с ключами, созданными на этом шаге.

Шаг 3 - идентификация дублирования данных: перемещайте окно фиксированного размера в соответствии с порядком записей, сравнивая каждую запись с другими записями в окне. Если размер окна W, при каждой новой записи, попадающей в окно, сравнивается с предыдущей записью, то запись была признана “подходящей” для W *1. На самом деле, точность обнаружения повторяющихся записей во многом зависит от созданного ключевого слова sort, которое напрямую влияет на эффективность и точность сопоставления. Если неправильно выбрать ключевые слова, можно пропустить большое количество повторяющихся записей.

Во-первых, поскольку две повторяющиеся записи после сортировки могут находиться далеко от физического местоположения, они могут никогда не оказаться одновременно в одном и том же скользящем окне и не могут быть идентифицированы как повторяющиеся записи.

Во-вторых, трудно определить размер скользящего окна W. Если значение W слишком велико, время сравнения увеличится, что приведет к ненужности некоторых сравнений; если значение W слишком мало, некоторые повторяющиеся записи не могут быть обнаружены. Когда размер

всех дублированных кластеров в наборе данных сильно различается, независимо от того, как выбран размер W, это нецелесообразно [5.13, 5.14]. Кроме того, для всего процесса сопоставления временная сложность алгоритма равна O(n), где n – общее число записей набора данных, а алгоритм интеллектуальной кластеризации гетерогенных больших данных показан ниже (рис. 5.3).

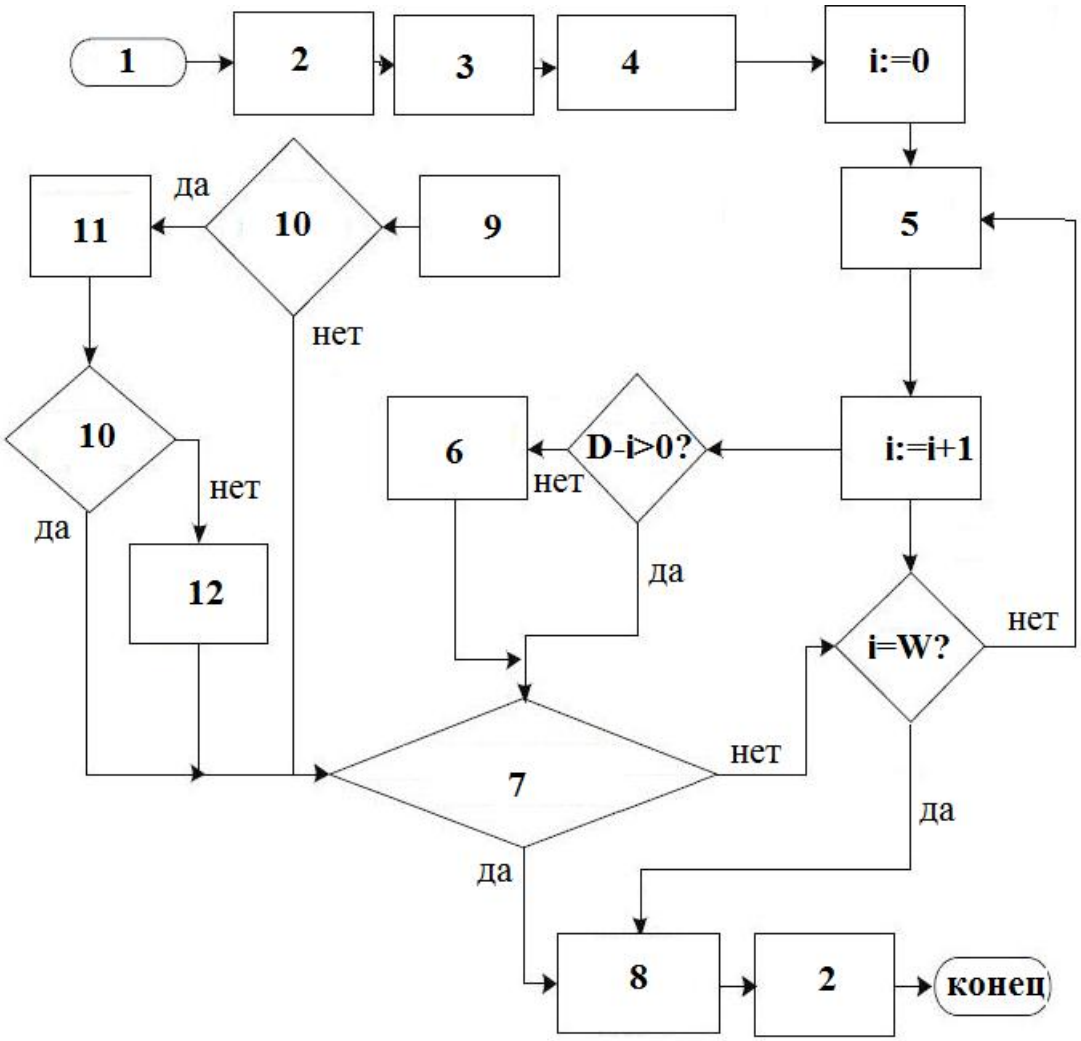


Рис. 5.3. Алгоритм интеллектуальной кластеризации разнородных больших данных: 1 – старт; 2 – N записей; 3 – сортировка; 4 - установка размера окна W равным M; 5 - перемещение окна вниз по очереди и запись его в переменную D; размещение D записей; 7 – сравнение записанных n-й и n-W-й записей; 8 – очистка дубликатов; 9 – быстрая сортировка; 10 – порог превзойден?; 11 - эффективное соответствие; 12 - улучшенные правила транзитивности

Учитывая, что размер окна W трудно определить в алгоритме SNM,

атрибуты анализируются и наборы данных сортируются много раз, что делает повторяющиеся записи более агрегированными, таким образом, попадая в одно и то же скользящее окно в одно и то же время.

Далее, при сопоставлении полей алгоритм присваивает специальный вес каждому атрибуту и вводит понятие эффективного веса, умножает вес на сходство соответствующего непустого атрибута, а затем объединяет их, чтобы получить сходство всей записи. И он используется для определения того, дублируют ли две записи значение.

Наконец, в процессе выбора атрибута обоснованность выбора доказывается путем проверки сходства между m конкретными окнами. На данный момент завершена разработка алгоритма интеллектуальной кластеризации разнородных больших данных в среде сложных атрибутов.

5.1.3. Эксперимент

5.1.3.1. Условия проведения эксперимента

Для проверки эффективности алгоритма 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.

Экспериментальный прототип основан на кластере hadoop, кластере hive, кластере sqoop и т.д. Hadoop - это распределенная файловая система. Она может обрабатывать крупномасштабные данные параллельно с кластером hadoop. Hive в основном отвечает за преобразование файла структурированных данных в таблицу базы данных и предоставление функции SQL-запроса. Затем SQL-инструкция преобразуется в задачу MapReduce и загружается в кластер для реализации.

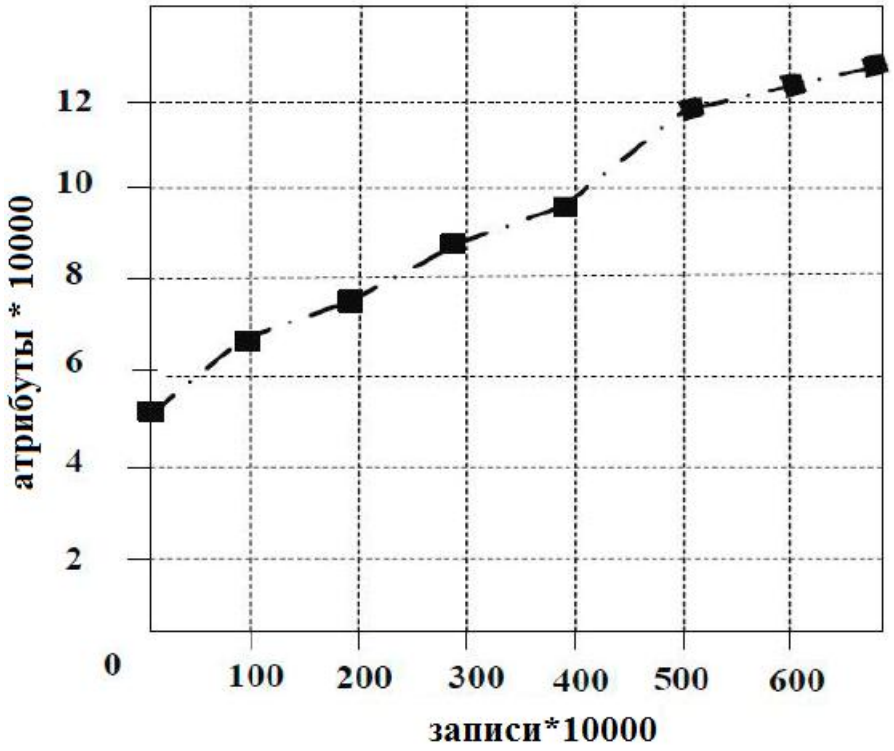
5.1.3.2. Подготовка данных

Подготовка данных - это первый шаг очистки данных. Цель этого шага - описать данные процесса, а затем выбрать наиболее подходящий образец данных для моделирования. Основная задача этого шага - извлечь данные из базы данных и проверить набор данных. Для извлечения эффективных данных из базы данных, определения рабочей области, анализа требований и любых изменений условий эксплуатации, а также обеспечения эффективности извлечения информации для извлечения эффективных данных из базы данных с помощью выборок и переменных.

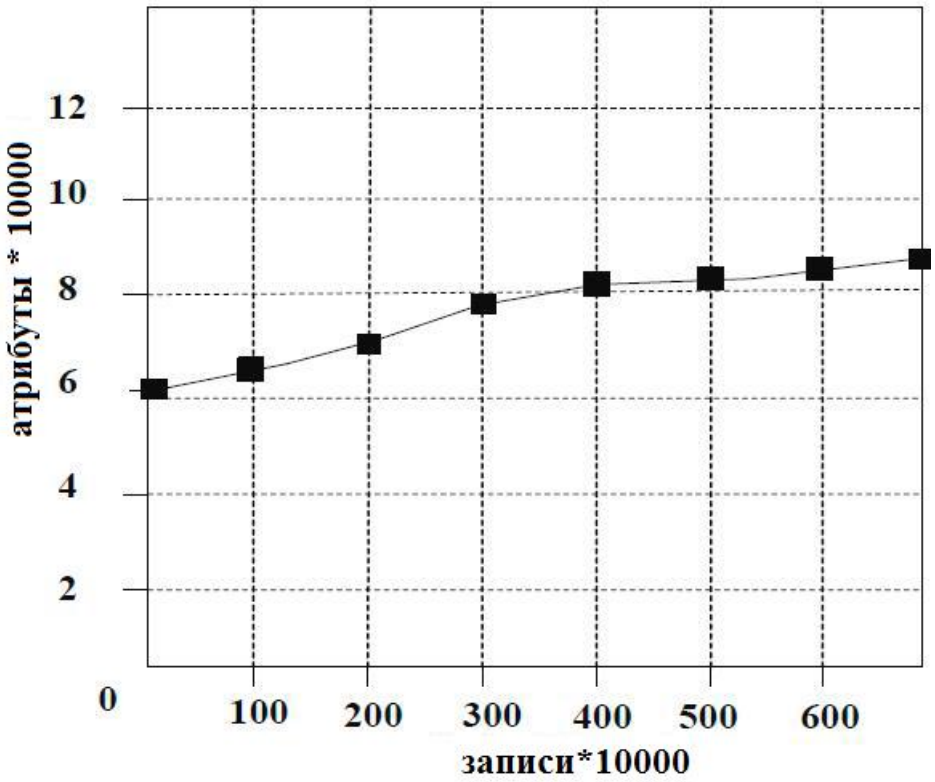
5.1.3.3. Эксперимент и анализ

Перед контрастным тестированием необходимо проанализировать набор данных, чтобы узнать их характеристики. Для описания источника данных выбраны два параметра: количество атрибутов в записи данных и отношение шума к содержащимся в записи данных "грязным" данным.

На рис. 5.4 мы видим, что количество кортежей увеличивается по мере увеличения количества вводимых записей. На рис. 5.4 показано, что доля аналогичных дубликатов и несоответствий в наборе данных по-прежнему очень высока, некоторые из них достигли более 25%, самый низкий показатель также превышает 10%. Показано, что в среде сложных атрибутов не только масштаб записи данных огромен, но и количество атрибутов данных также достигает миллиона, и данные содержат данные о различных проблемах с качеством данных значительного масштаба, а количество исключенных свойств превышает количество установленных проверок. Показано, что разработанный алгоритм является относительно стабильным, а эксперимент по проверке соответствует эксперименту с реальными данными. Показано также, что схема эксперимента является разумной, а результаты соответствуют действительности.



а) Экспериментальные результаты применения традиционных алгоритмов



б) Экспериментальные результаты работы предложенного алгоритма

Рис. 5.4. Экспериментальные результаты

5.1.4. Алгоритм интеллектуальной кластеризации для гетерогенных больших данных - итоги

Из-за большого количества атрибутов кластеризации гетерогенных больших данных в традиционных алгоритмах интеллектуальной кластеризации гетерогенных больших данных в средах со сложными атрибутами в данной статье предлагается новый алгоритм интеллектуальной кластеризации гетерогенных больших данных. В среде сложных атрибутов путем очистки пространства параметров среды сложных атрибутов, введения элементов правил кластеризации в разреженном подпространстве, настройки показателей интеллектуальной кластеризации разнородных больших данных, чтобы реализовать интеллектуальный расчет кластеризации разнородных больших данных. Экспериментальная проверка показывает, что предложенный алгоритм интеллектуальной кластеризации для гетерогенных больших данных дает лучший эффект при кластеризации гетерогенных больших данных в среде со сложными атрибутами.

Качество анализа данных в алгоритме интеллектуальной кластеризации разнородных больших данных зависит от качества данных, собранных из разных источников, поскольку наборы данных в реальных приложениях часто содержат несогласованные данные, зашифрованные данные, зашумленные значения и интеграцию данных, вызванную различными ошибками. Поскольку качество данных часто меняется в процессе сбора, хранения, объединения и анализа данных, очистка данных не ограничивается подзадачей предварительной обработки данных. Она проходит через все звенья обработки данных. Учитывая растущую массу разнородных данных из нескольких источников и более сложные структуры данных, для повышения эффективности выявления похожих повторяющихся записей и решения проблемы качества данных, связанной с неточностью данных, необходимо постоянно совершенствовать последующие исследования.

5.2. Исследование метода оптимизации данных для эксплуатации и сопровождения базы знаний программного обеспечения на основе облачных вычислений

5.2.1. Повышение точности сопоставления данных при эксплуатации и обслуживании базы знаний программного обеспечения

Для повышения точности сопоставления данных при эксплуатации и обслуживании базы знаний программного обеспечения предложен метод оптимизации данных, основанный на облачных вычислениях. Для достижения цели точного обнаружения были усовершенствованы этапы обнаружения аномалий в данных о работе и обслуживании базы знаний программного обеспечения, а также завершена оптимизация данных о работе и обслуживании базы знаний программного обеспечения. Наконец, эксперимент доказывает, что точность согласования метода оптимизации данных по эксплуатации и техническому обслуживанию базы знаний программного обеспечения, основанного на облачных вычислениях, значительно повышается по сравнению с традиционным методом эксплуатации и технического обслуживания.

С быстрым развитием современных технологий масштабы кластера данных базы знаний программного обеспечения постепенно расширяются. Чтобы лучше гарантировать стабильность работы базы знаний, необходимо дальнейшее совершенствование технологии работы базы данных и обслуживания базы знаний программного обеспечения [5.15]. Таким образом, в результате анализа и исследования распространенных в настоящее время методов оптимизации данных по эксплуатации и техническому обслуживанию было установлено, что из-за неразумной структуры базы знаний программного обеспечения и отсутствия эффективного модуля сопоставления ассоциаций сложно распределять и обрабатывать массивные данные измерений, накапливающиеся в базе данных, в результате чего в низ-

ком качестве работы и обслуживания данных и других проблемах [5.16]. Для решения вышеуказанных проблем в сочетании с методом облачных вычислений для оптимизации работы и технического обслуживания базы знаний программного обеспечения используется метод обработки данных посредством эффективного сбора многомерной информации базы знаний, и в соответствии с результатами сбора анализируется ассоциация данных измерений базы данных программного обеспечения, и размерность взаимосвязь данных по эксплуатации и техническому обслуживанию базы данных оценивается и сопоставляется научно и обоснованно. Таким образом, можно эффективно изучить возможности управления данными и потенциальную ценность эксплуатации и обслуживания базы знаний программного обеспечения, а также повысить качество и эффективность эксплуатации и обслуживания данных.

5.2.2. Оптимизация данных при эксплуатации и обслуживании базы знаний программного обеспечения на основе облачных вычислений

5.2.2.1. Алгоритмы объединения данных при эксплуатации и обслуживании базы знаний на основе облачных вычислений

База знаний, связанная с эксплуатацией и техническим обслуживанием, состоит из двух частей: базы данных по эксплуатации и техническому обслуживанию и системы поиска данных. Для достижения цели исследования, заключающейся в оптимизации данных по эксплуатации и техническому обслуживанию, первым шагом является инициализация объединения данных по эксплуатации и техническому обслуживанию в Базе знаний. При построении структуры базы знаний программного обеспечения о характерном периоде изменения и стабильности данных по эксплуатации и техническому обслуживанию в основном судят по времени задержки и качеству факторов защиты от помех в процессе эксплуатации и технического обслуживания [5.17]. Без учета влияния факторов помех на данный мо-

мент, чем дольше период изменения характеристик данных, тем выше точность извлечения данных из базы знаний программного обеспечения.

В соответствии с вышеуказанными принципами в сочетании с алгоритмом планирования оптимизации сети время произвольного изменения объектов данных в базе знаний устанавливается равным α. Если объем данных в базе знаний программного обеспечения равен n, а степень влияния объема данных на получение признаков равна x, то, используя принцип извлечения распределения признаков, алгоритм может эффективно описать начальную степень изменения признаков в данных. Конкретная формула расчета может быть выражена следующим образом:

$$C_n(x) = \begin{cases} 1 - \left(\frac{x}{a}\right)^{-a}, & x \geq a \\ 0, & x > 0 \end{cases} \tag{5.5}$$

Согласно принципу упомянутого выше алгоритма, если поток данных ς в базе знаний программного обеспечения DRC обрабатывается с помощью классификации признаков, если m - общий характеристический параметр данных, D - поток данных об изменении данных в базе данных HOL, а ϑ - наименьший конечный параметр потенциальной релевантности данных. Тогда вероятностный алгоритм для сопоставления признаков аномальных данных может быть описан следующим образом:

$$S_m(x) = \begin{cases} \dfrac{[J * C(x)]^V}{[DRC - HOL_{n-1}]^J} - J \\ \exp\left[D * \left(\dfrac{a-1}{1+\sqrt{n}}\right)\right] \dfrac{[DRC_n * C(x)]^V}{[HOL_m]^J} \end{cases} \tag{5.6}$$

Если показатель вероятности изменения признака всех данных в базе знаний равен t, то может быть получена стандартная информационная ассоциация, сопоставляющая параметры всех данных в потоке данных ς(x) и ϑ(x), которая выражается как [5.18, 5.19]:

$$A(n)=\begin{cases} S_m(x)\left[a - t(n)\right] - 1, t_{max}[n] > a \\ S_m(x)\left[t(n) - 1\right] + \dfrac{t}{a}, t_{max}[n] < a \end{cases} \quad (5.7)$$

В соответствии с приведенным выше алгоритмом мы можем точно определить относительность данных об эксплуатации и техническом обслуживании в базе знаний программного обеспечения.

5.2.2.2. Обнаружение аномалий в работе и техническом обслуживании базы знаний программного обеспечения

В процессе эксплуатации и технического обслуживания легко возникает большое количество аномальных данных, что приводит к несвязанным результатам эксплуатации и технического обслуживания данных. В работе объединены метод облачных вычислений для оптимизации работы базы данных и метод управления техническим обслуживанием [5.20, 5.21]. В соответствии с полученными характеристиками аномальных данных можно точно оценить типы и характеристики аномальных данных и сопоставить их в базе данных, а также выбрать необходимые информационные данные для сравнения и коррекции, чтобы обеспечить эффективную эксплуатацию и обслуживание аномальных данных. Очень важно собирать ключевые параметры аномальных характеристик данных в процессе обнаружения и исправления аномальных данных. Если при сборе данных возникают ошибки, это напрямую влияет на сопоставление ассоциаций данных, а также на эксплуатацию и техническое обслуживание [5.22, 5.23]. Благодаря относительно сложной функции сбора аномальных данных, для облегчения последующих операций визуализируется пространственное поведение данных, что способствует эффективной работе и сопровождению аналогичных аномальных данных в будущем и, наконец, достижению оптимизации работы и сопровождения данных кислотно-щелочной базы знаний.

База знаний о пространственном поведении программного обеспечения включает данные о социальном поведении, данные о логическом языковом поведении и данные о пространственном перемещении. Этапы сбора данных о пространственном поведении делятся на четыре этапа: подготовка данных о пространственном поведении, интеллектуальный анализ данных, представление данных и оценка данных. Процесс показан на рис. 5.5.

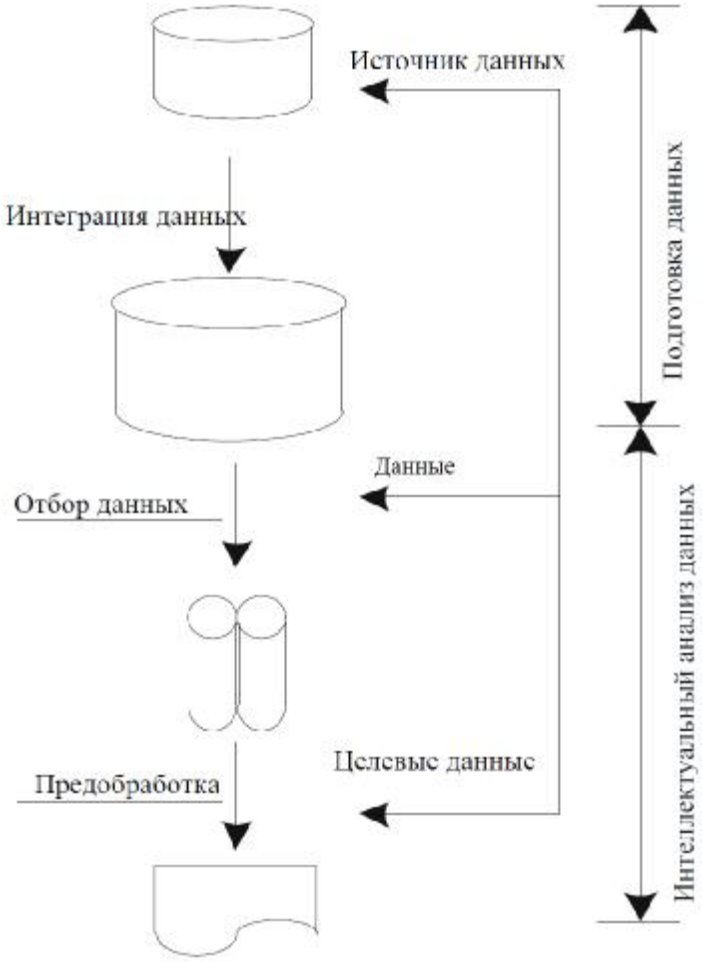


Рис. 5.5. Этапы интеллектуального анализа данных при эксплуатации и обслуживании базы знаний программного обеспечения

В процессе интеллектуального анализа данных о пространственном поведении при эксплуатации и обслуживании базы знаний программного обеспечения в соответствии с описанными выше шагами предполагается, что минимальная степень поддержки интеллектуального анализа данных равна s, а минимальная степень достоверности равна C_0, после чего будет

получен набор элементов-кандидатов. Если степень поддержки набора больше или равна минимальной степени поддержки, он называется набором часто используемых элементов.

Данные в базе данных проверяются до тех пор, пока не будут созданы новые наборы-кандидаты.

В соответствии с описанными выше этапами интеллектуального анализа данных по эксплуатации и обслуживанию базы знаний программного обеспечения обрабатываются данные о социальном поведении. Социальные данные, темы, именованные объекты и их связи определяются как иерархическая семантическая модель. Каждое сообщение определяется как узел.

$$П=\{n: n\hat{I}\ V_T\} \tag{5.8}$$

Здесь n представляет данные сообщения в данных о пространственном поведении, а V_T - набор сообщений той же темы. График кластеризации, полученный после обхода раздела, представлен матричным вектором, а выражение графа показано в формуле (5.9).

$$AG=<V_T, E_T> \tag{5.9}$$

В уравнении (5.9) E_T - это отношение классификации именованных объектов. Согласно тому же методу интеллектуального анализа данных, данные о логическом поведении языка и пространственном перемещении в spatial behavior обрабатываются, а кластеризация объединяет данные о пространственном поведении для получения окончательных результатов интеллектуального анализа.

Информация о пространственном поведении, содержащаяся в данных о работе и техническом обслуживании базы знаний программного обеспечения, преобразуется в графическое представление, поэтому необходимо преобразовать траекторию пространственного поведения. Процесс преобразования разделен на два этапа, а именно: создание траектории пространственного поведения и преобразование траектории. Для формирова-

ния траектории пространственного поведения необходимо рассчитать дистанцию поведения и оценить направление пространственного поведения. При вычислении расстояния и направления необходимо пройти каждый узел в пространстве, и буфер пути пересекается, чтобы получить заданное пространственное поведение L, тогда общая длина расстояния также равна L, где коэффициент длины между каждыми двумя узлами равен j, тогда общее значение направления пространственного поведения поведение рассчитывается следующим образом:

$$L_{\alpha}=k_1\alpha_1+ k_2\alpha_2+\ldots+ k_n\alpha_n \quad (5.10)$$

Угол направления каждого пространственного сегмента равен α_n, вычисляются значение угла направления и значение длины расстояния, и, наконец, получается результат преобразования траектории пространственного поведения данных о работе и размерах базы знаний программного обеспечения.

5.2.2.3. Реализация оптимизации данных для эксплуатации и сопровождения базы знаний программного обеспечения

После завершения эффективного сопоставления массивных данных с помощью алгоритма сопоставления данных по эксплуатации и техническому обслуживанию базы знаний обнаруживаются и исправляются ненормальные данные, которые не совпали. После завершения исправления аномальных данных используется существующая технология передачи данных системы эксплуатации и технического обслуживания для шифрования данных эксплуатации и технического обслуживания и других аспектов управления и обработки передачи, чтобы избежать проникновения агрессивных данных [5.24, 5.25]. Поскольку содержание управления работой с данными и их обслуживанием тривиально, я не буду здесь делать дополнительных заявлений.

После управления данными и их валидации информация о валидации

передается обратно для повышения точности, безопасности и эффективности работы с данными и обслуживания базы знаний программного обеспечения. Для того чтобы эффективно оптимизировать эксплуатацию и обслуживание данных базы знаний программного обеспечения, прежде всего, необходимо собрать и проанализировать характеристики данных, а также добавить соответствующие функции распознавания признаков, категориального анализа, хранения данных и поиска связанных данных в структуру базы знаний программного обеспечения [5.26, 5.27]. Установив и оптимизировав модуль управления данными, модуль изменения классификации объектов (прикладной уровень) и модуль управления передачей и публикацией данных (уровень поддержки данных) в базе знаний программного обеспечения, можно эффективно реализовать точное извлечение объектов из различных данных по эксплуатации и техническому обслуживанию. В соответствии с характеристиками сбора данных можно определить корреляцию между данными, а в базе данных программного обеспечения - завершить работу по передаче данных и управлению ими. Интегрируя предыдущие идеи и алгоритмы, спроектирована архитектура программной системы эксплуатации и сопровождения базы знаний программного обеспечения (рис. 5.6).

Как показано на рис. 5.6, в сочетании с предыдущим методом оптимизации работы базы знаний программного обеспечения и процесса передачи данных, в сочетании со сбором данных, шифрованием данных и другими техническими принципами, процесс передачи данных в зашифрованном виде и проверки их расшифровки реализует построение модели управления перемещением данных Интернета вещей и их технического обслуживания, таким образом, аномальные данные в базе знаний своевременно обнаруживаются для достижения цели точного сбора и оптимизации данных по эксплуатации и техническому обслуживанию.

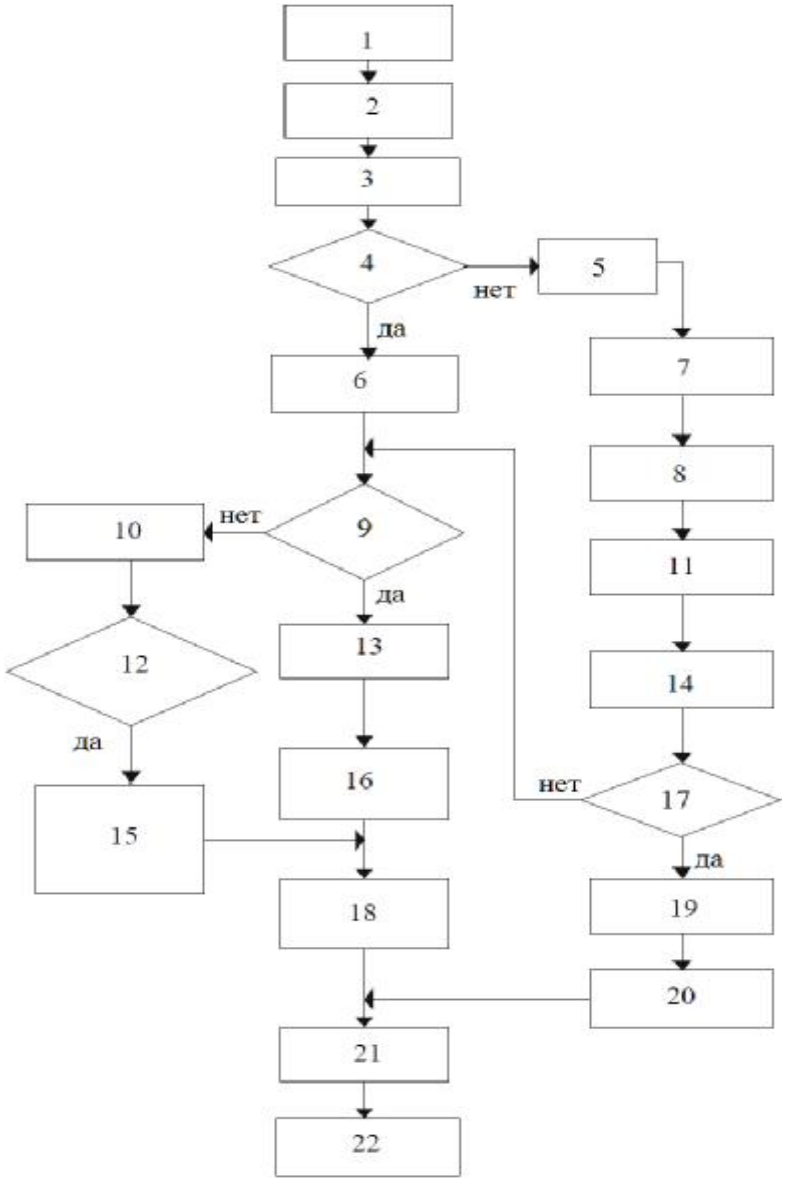


Рис. 5.6. Архитектура программной системы эксплуатации и сопровождения базы знаний программного обеспечения: 1 - Идентификация; 2 - Анализ соответствия параметров данных; 3 - Инициализация информационной совокупности; 4 - Есть ли свободный порт?; 5 - Генетические манипуляции; 6 - Кодирование набора параметров; 7 - Классификация подмножеств выборки; 8 - Категория кластеризации вычисление; 9 - Совпадают ли данные; 10 - Оповещение об информационном исключении системы; 11 - Оценка информационной группы; 12 - Не является ли номер пустым; 13 - Запрос доступной информации об устройстве; 14 - Проверка соответствия индекса; 15 - Запрос информации о конфигурации номера доступа; 16 - Проверочный тест; 17 - Выполняется ли правило остановки?; 18 - Запрос порта простоя устройства; 19 - Генетическая манипуляция; 20 - Определение набора параметров проблемы; 21 - Расшифровка данных для проверки; 22 - Конец

5.2.3. Анализ результатов эксперимента

Чтобы проверить практическую эффективность метода оптимизации данных по эксплуатации и техническому обслуживанию базы знаний программного обеспечения на основе облачных вычислений, был проведен имитационный эксперимент, в ходе которого были сравнены и протестированы функциональные и нефункциональные возможности системы эксплуатации и технического обслуживания. Чтобы уточнить цель обнаружения, проблемы в тесте исправляются и совершенствуются. Сначала регистрируются данные информационной системы эксплуатации и технического обслуживания, и получается следующая табл. 5.1.

Таблица 5.1

Стандарты тестирования по оптимизации данных эксплуатации и технического обслуживания

Цель	**Стандарт тестирования**
Требования тестирования	Разумное планирование и полный набор функций
	Сопоставление информации
Тестовая процедура	Регистрация; вход в систему; аутентификация; загрузка данных
	Обнаружение функциональной операции
	Обнаружение информации
	Определение подлинности данных
Результаты теста	Совпадение приемлемое, и тест прошел успешно
	Обнаружение данных завершено без дефектов, данные соответствуют ожиданиям, количество аномальных данных уменьшено, степень соответствия улучшена, и функциональный тест пройден.

Эксперименты проводились в соответствии со стандартизированными данными, приведенными в таблице выше. Метод оптимизации данных по эксплуатации и техническому обслуживанию базы знаний программного обеспечения и традиционный метод эксплуатации и технического обслуживания сравниваются и проверяются путем объединения метода облачных вычислений. Эффективность эксплуатации и технического обслу-

живания оценивается путем сравнения и анализа степени соответствия данных. Конкретные результаты обнаружения приведены ниже (рис. 5.7).

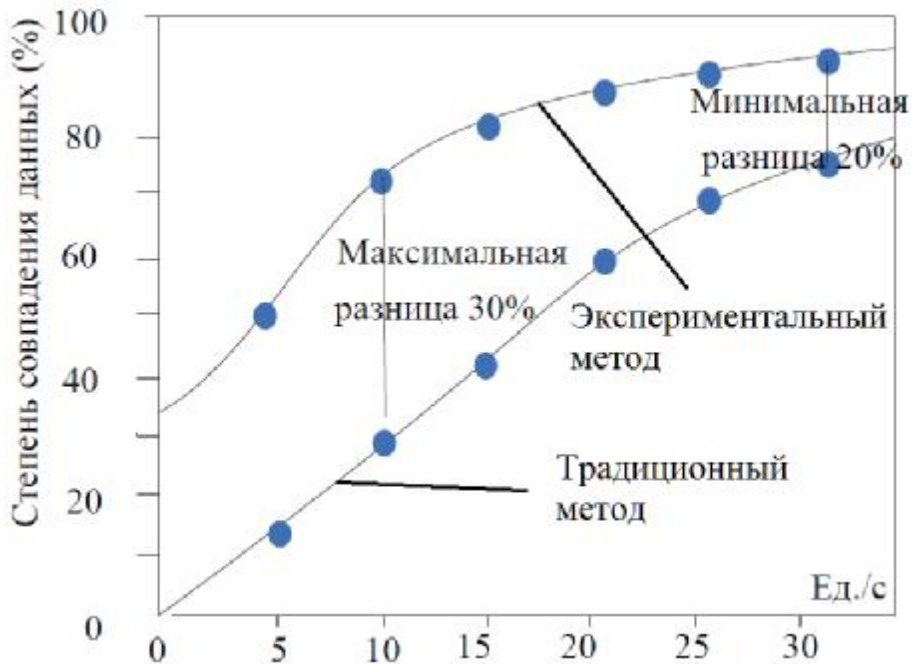
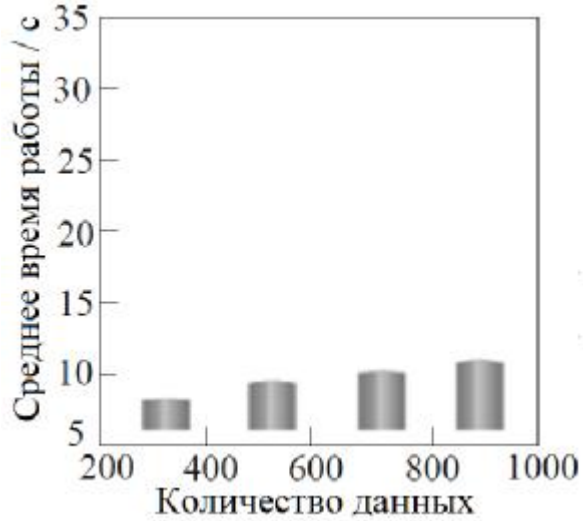


Рис. 5.7. Сравнение результатов экспериментальных испытаний

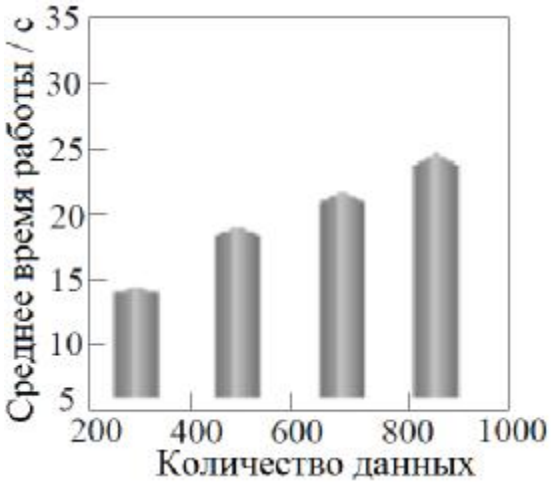
Согласно приведенным выше экспериментальным результатам, нетрудно обнаружить, что степень соответствия данных базы знаний программного обеспечения по сравнению с традиционным методом улучшается с помощью метода облачных вычислений, а общая ситуация с соответствием улучшается на 20-30% по сравнению с традиционным методом. В процессе эксплуатации и технического обслуживания, чем выше степень соответствия данных, тем меньше аномальных данных и тем выше эффективность эксплуатации и технического обслуживания. Таким образом, подтверждено, что метод облачных вычислений полностью удовлетворяет исследовательским требованиям метода эксплуатации и обслуживания данных базы знаний программного обеспечения.

Предлагаемый метод сравнивается с традиционным методом работы с данными и их сопровождения. Результаты экспериментов показаны на рис. 5.8. Показано среднее время работы с данными и их обслуживания для

двух методов при разных объемах данных. Согласно рис. 5.8, с увеличением объема данных время работы обоих методов увеличивалось. Однако, при сравнении можно увидеть, что время работы традиционного метода намного выше, чем у предлагаемого метода, и время работы еще больше увеличивается в ходе эксперимента. Для сравнения, этот метод превосходит традиционный и обеспечивает более высокую эффективность обработки данных и технического обслуживания.



а) Среднее время работы метода, представленного в данной статье



б) Среднее время работы традиционных методов

Рис. 5.8. Сравнение среднего времени работы различными методами

5.2.4. Эффект оптимизации метода облачных вычислений

Метод облачных вычислений используется для обновления и опти-

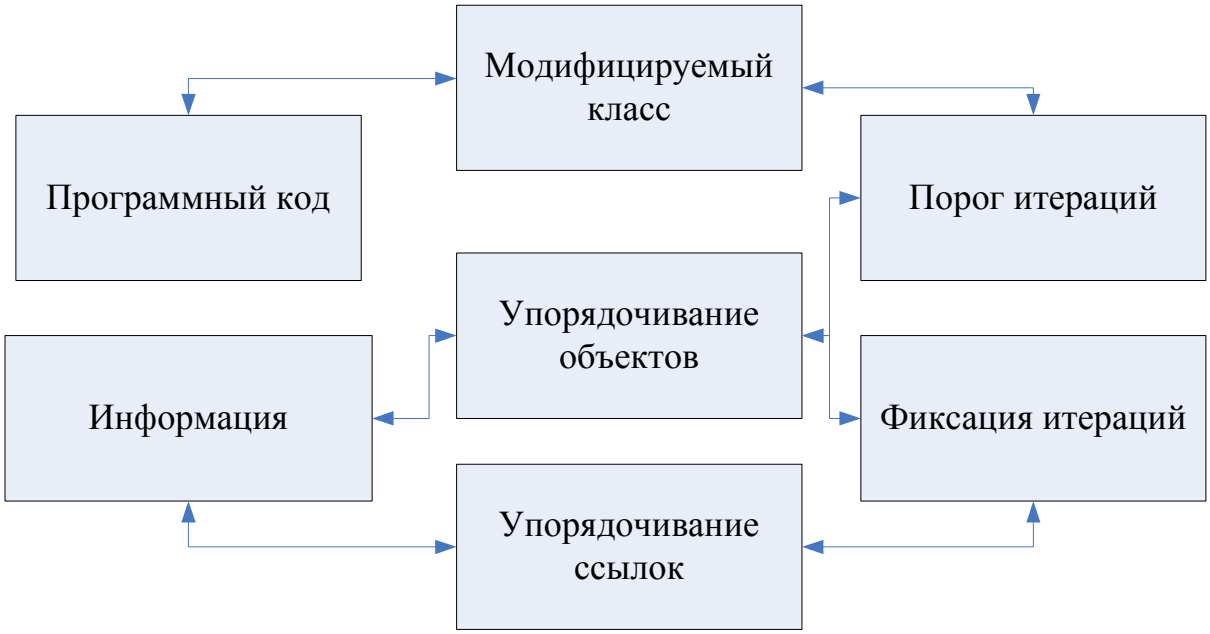
мизации базы знаний программного обеспечения software knowledge base для обеспечения безопасной и надежной работы базы знаний. Сравнительный эксперимент проводится для сравнения эффекта оптимизации метода. Эксперимент доказывает, что сочетание метода облачных вычислений имеет большое практическое значение для работы с данными базы знаний программного обеспечения и метода обслуживания данных, который позволяет быстрее и точнее обрабатывать информацию о данных и обеспечивать получение знаний.

5.3. Архитектура программной системы оптимизации распределения больших данных в Интернете вещей

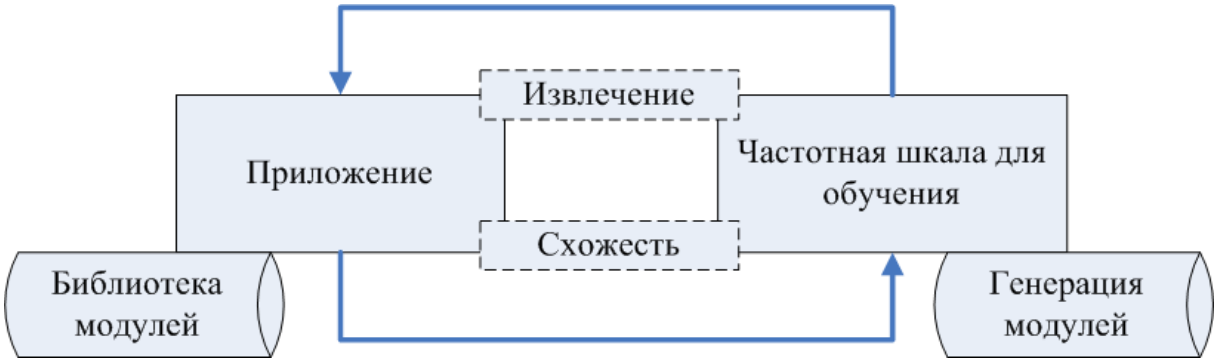
Разработанная архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей представлена на рис. 5.9.

Укрупненно архитектура содержит три больших блока – итерационное распределение больших данных, обработка данных в процессе итерации и оптимизация.

а) Укрупненная архитектура



б) Архитектура блока итерационного распределения больших данных



в) Архитектура блока адаптации и оптимизации

5.4. Выводы по главе 5

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

г) Архитектура блока обработки данных в процессе итерации

Рис. 5.9. Архитектура программной системы оптимизации больших данных от датчиков в Интернете вещей

Литература к главе 5

5.1. Anonymous: Large data optimal clustering algorithms in cloud computing environment based on PSO. Electron. Des. Eng. 26(19), 86–89+94 (2018)

5.2. Qujie: Research on intelligent parallel clustering method for large data in virtual environment. Comput. Measur. Control 25(6), 257–260 (2017)

5.3. Yi, M., Ting, X., Shaobin, L.: Research on NoSQL distributed large data mining method in complex attribute environment. Sci. Technol. Eng. 17(09), 244–248 (2017)

5.4. Chunhua, H.: Clustering algorithm analysis of multidimensional data de-duplication in large data environment. Comput. Prod. Circ. 32(11), 151 (2017)

5.5. Anonymous: Prediction and analysis of energy consumption behavior of integrated energy system users under multi-source heterogeneous large data. Smart Power 46(10), 92–101 (2018)

5.6. Li, B.H., Junhua, C., et al.: Distributed clustering algorithms of attribute graph under multiagent architecture. Comput. Sci. 44(S1), 407–413 (2017)

5.7. Linjing, W., Lulu, N., Bin, G., et al.: Sparse fractional feature selection clustering algorithms based on entropy weighting in large data. Comput. Appl. Res. 35(8), 59–60 + 69 (2018)

5.8. Houlisa: Clustering algorithm design for eliminating redundant features in large data sets.Mod. Electron. Technol. 41(14), 56–58+62 (2018)

5.9. Xiaoyu, C., Xiaojing, L., Haiying, M.: A fast automatic clustering algorithm for large data. Comput. Appl. Res. 34(9), 2651–2654 (2017)

5.10. Xiaoyan, T.: A large data text clustering algorithm based on word embedding and density peak strategy. Innov. Appl. Sci. Technol. 6, 90–90 (2017)

5.11. Fu, W., Liu, S., Srivastava, G.: Optimization of big data scheduling in social networks. Entropy 21(9), 902 (2019)

5.12. Sun, G., Liu, S. (eds.): ADHIP 2017. LNICST, vol. 219. Springer, Cham (2018). https://doi. org/10.1007/978-3-319-73317-3

5.13. Shuai, L., Weiling, B., Nianyin, Z., et al.: A fast fractal based compression for MRI images. IEEE Access 7, 62412–62420 (2019)

5.14. Liu, S., Li, Z., Zhang, Y., et al.: Introduction of key problems in long-distance learning and training. Mob. Netw. Appl. 24(1), 1–4 (2019)

5.15. Ding Y. Technical analysis of intelligent monitoring system for operation and maintenance data collection and business process of broadcasting system// Western Radio Telev. 2018, 426(10), 189-193.

5.16. Jin X., Yan L., Liu J. et al. Fault analysis and operation research for enterprise database: taking Oracle Database as an example// Software 2017, 38(10), 178-181.

5.17. Dou J., Dai F. Research on operation and maintenance scheme of intelligent log analysis platform based on big data environment// J. Jiujiang Vocat. Tech. Coll. 2017, 25(04), 96-98.

5.18. Jing G., Hu C. et al. Software vulnerability detection algorithm for 8031 single chip microcomputer system based on vulnerability knowledge base// J. Beijing Inst. Technol. 2017, 34(4), 371-375

5.19. Zheng P., Shuai L., Arun S., Khan M. Visual attention feature (VAF): a novel strategy forvisual tracking based on cloud platform in intelligent surveillance systems// J. Parallel Distrib. Comput. 2018, 120, 182-194.

5.20. Zhong L., Guo T., Zhang M. Design and implementation of online ceramic mineralresources management knowledge base based on LINGO// Intell. Comput. Appl. 2018, 36(1), 68-71.

5.21. Liu S., Li Z., Zhang Y., Cheng X. Introduction of key problems in long-distance learningand training. Mobile Netw. Appl. 2018, 24(1), 1-4. https://doi.org/10.1007/s11036-018-1136-6.

5.22. Wang F. Summary of technological innovation of broadcasting operation and maintenance data collection and process monitoring and processing system// Modern Telev. Technol. 2017, 34 (5), 138-141.

5.23. Zhao C., Sun L., Gao X. et al. Discussion on protection intelligent operation andmaintenance technology based on source data maintenance mechanism// Power Grid Clean Energy 2017, 52(09), 82-87.

5.24. Tan L., Zhong H. Operation and maintenance practice of data center construction based on cloud computing model// Inf. Comput. (Theor. Ed.) 2018, 408(14), 21-23.

5.25. Shuai L., Weiling B., Nianyin Z. et al. A fast fractal based compression for MRI images// IEEE Access 2019, 7, 62412-62420

5.26. Yang Y., Zhang S., Kang Q. Design of intelligent early warning system based on grid operation and maintenance data// Inner Mongolia Electric Power Technol. 2017, 35(4), 20-23.

5.27. Shuai L., Gelan Y. Advanced Hybrid Information Processing. - Springer, New York. 594 p.

Заключение

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

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.

Список использованных источников

1. 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://moitvivt.ru/ru/journal/pdf?id=2004 DOI: 10.26102/2310-6018/2025.50.3.026.

2. 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.

3. 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

4. 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

5. 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.

6. 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

7. 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.

8. 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

9. 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.

10. 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.

11. 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.

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.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.

14.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.

15. Aggi, M. K., & Jain, S. (2018). Survey towards an integration of big data analytics to big insights for valuecreation// Information Processing & Management, 54(5), 758–790. doi:10.1016/j.ipm.2018.01.010

16. Akter, S., Bandara, R., Hani, U., Wamba, S. F., Foropon, C., & Papadopoulos, T. (2019). Analytics-based decision-making for service systems: A qualitative study and agenda for future research// International Journal of Information Management, 48, 85–95. doi:10.1016/j.ijinfomgt.2019.01.020

17. Al-Turjman F (2018) Information-centric framework for the internet of things (IoT): traffic modeling& optimization. Futur Gener Comput Syst 80:63–75

18. Anonymous: Large data optimal clustering algorithms in cloud computing environment based on PSO. Electron. Des. Eng. 26(19), 86–89+94 (2018)

19. Anonymous: Prediction and analysis of energy consumption behavior of integrated energy system users under multi-source heterogeneous large data. Smart Power 46(10), 92–101 (2018)

20. Ardagna, D., Cappiello, C., Sam, W., Vitali, M., 2018. Context-aware data quality assessment for big data. Future Generat. Comput. Syst. 89, 548–562.

21. Arndt, H. (2018). Knowledge discovery and anomalies - towards a dynamic decision-making model for medical informatics [Doctoral dissertation]. Stellenbosch University.

22. Atencia, M., David, J., Euzenat, J., Napoli, A., Vizzini, J., 2020. Link key candidate extraction with relational concept analysis. Discrete Appl. Math., https://doi.org/10.1016/j.dam.2019.02.012.

23. Ballou, D.P., Pazer, H.L., 2003. Modeling completeness versus consistency tradeoffs in information decision contexts. IEEE Trans. Knowl. Data Eng. 15 (1), 240–243.

24. Barhamgi, M., Benslimane, D., Amghar, Y., Cuppens-Boulahia, N.,

Cuppens, F., 2013. Privcomp: a privacy-aware data service composition system. EDBT/ICDT 55 (6), 86–97.

25. Bertino, E., Ferrari, E., 2018. Big data security and privacy. In: A Comprehensive Guide through the Italian Database Research over the Last 25 Years. Springer, pp. 425–439.

26. Bi K, An K, Li X (2020) A resource optimization allocation strategy for China's shipbuilding industry green innovation system. Int J Innov Technol Manag 17(4):2050029–2050042

27. Bijarbooneh FH, Du W, Ngai CH (2017) Cloud-assisted data fusion and sensor selection for internet-of-things. IEEE Internet Things J 3(3):257–268

28. Boulakbech, M., Messai, N., Sam, Y., Devogele, T., Hammoudeh, M., 2017. IoT mashups: from iot big data to iot big service. In: ICFNDS.

29. Brynjolfsson, E., Hitt, L. M., & Kim, H. H. (2011). Strength in numbers: How does data-driven decisionmaking affect firm performance?

30. Cai, L., Zhu, Y., 2015. The challenges of data quality and data quality assessment in the big data era. Data Sci. J. 14, https://doi.org/10.5334/dsj-2015-002.

31. Casado, R., & Younas, M. (2015). Emerging trends and technologies in big data processing// Concurrency and Computation, 27(8), 2078–2091.

32. Chiheb, F., Boumahdi, F., & Bouarfa, H. (2019). A New Model for Integrating Big Data into Phases of Decision-Making Process// Procedia Computer Science, 151, 636–642. doi:10.1016/j.procs.2019.04.085.

33. Choi K, Chung SH (2017) Enhanced time-slotted channel hopping scheduling with quick setup time for industrial Internet of Things networks. Int J Distrib Sens Netw 13(6):1362–1377

34. Chunhua, H.: Clustering algorithm analysis of multidimensional data de-duplication in large data environment. Comput. Prod. Circ. 32(11), 151 (2017)

35. Colleoni, E., Rozza, A., & Arvidsson, A. (2014). Echo chamber or public sphere? Predicting political orientation and measuring political homophily in Twitter using big data// Journal of Communication, 64(2), 317–332. doi:10.1111/jcom.12084.

36. Courtney, J. F. (2001). Decision making and knowledge management in inquiring organizations: Toward a new decision-making paradigm for DSS// Decision Support Systems, 31(1), 17–38. doi:10.1016/S0167-9236(00)00117-2.

37. De Maio, C., Fenza, G., Gallo, M., Loia, V., Senatore, S., 2014. Formal and relational concept analysis for fuzzy-based automatic semantic annotation. Appl. Intell. 40 (1), 154–177.

38. De Maio, C., Fenza, G., Loia, V., Orciuoli, F., 2017. Distributed online temporal fuzzy concept analysis for stream processing in smart cities. J. Parallel Distr. Comput. 110, 31–41.

39. De Mauro, A., Greco, M., & Grimaldi, M. (2015). What is big data? A consensual definition and a review of key research topics// A. I. P. Dans (Ed.),

AIP conference proceedings (pp. 97–104). Academic Press. doi:10.1063/1.4907823

40. Decker, G., & Debevoise, T. (2015, April). Quick Guide to Decision Modeling using DMN 1.0. Signavio, Inc.

41. Dietrich, D., Heller, B., & Yang, B. (2015). Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. John Wiley & Sons, Inc.

42. Ding K, Zhao H, Hu X (2017) Distributed channel allocation and time slot optimization for green internet of things. Sensors 17(11):2479–2491

43. Ding Y. Technical analysis of intelligent monitoring system for operation and maintenance data collection and business process of broadcasting system// Western Radio Telev. 2018, 426(10), 189-193.

44. Dou J., Dai F. Research on operation and maintenance scheme of intelligent log analysis platform based on big data environment// J. Jiujiang Vocat. Tech. Coll. 2017, 25(04), 96-98.

45. Efraim, T. (2011). Decision decision-making systems, modeling, and support. Pearson Education India.

46. Elgendy, N., & Elragal, A. (2016). Big data analytics in support of the decision making process// Procedia Computer Science, 100, 1071–1084. doi:10.1016/j.procs.2016.09.251.

47. Elsevier. (2017). Scopus Content Coverage Guide.

48. Emmanuel, I., & Stanier, C. (2016). Defining big data. Proceedings of the International Conference on Big Data and Advanced Wireless Technologies. Academic Press. doi:10.1145/3010089.3010090.

49. Ferchichi, H., Akaichi, J., 2016. Using mapreduce for efficient parallel processing of continuous k nearest neighbors in road networks. J. Software Syst. Dev., https://doi.org/10.5171/2016.356668.

50. Forest, H., Foo, E., Rose, D., & Berenzon, D. (2014). Big Data, how it can become a differentiator. Deutsche Bank.

51. Forman, E. H., & Selly, M. A. (2001). Introduction: Management Decision-Making Today// Decision by objectives: how to convince others that you are right (p. 1). Scientific, World.

52. Fu, W., Liu, S., Srivastava, G.: Optimization of big data scheduling in social networks. Entropy 21(9), 902 (2019)

53. Gabrel, V., Manouvrier, M., Murat, C., 2015. Web services composition: complexity and models. Discrete Appl. Math. 196, 100–114.

54. Gai, K., Qiu, M., Zhao, H., 2018. Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distr. Comput. 111, 126–135.

55. Gai, K., Qiu, M., Zhao, H., Tao, L., Zong, Z., 2016. Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. 59, 46–54.

56. Gai, K., Xu, K., Lu, Z., Qiu, M., Zhu, L., 2019. Fusion of cognitive

wireless networks and edge computing. IEEE Wirel. Commun. 26 (3), 69–75.

57. Galinina O, Tabassum H, Mikhaylov K (2016) On feasibility of 5Ggrade dedicated RF charging technology for wireless-powered wearables. IEEE Wirel Commun 23(2):28–37

58. Google Cloud. (2018, October 17). Data Lifecycle. Retrieved from https://cloud.google.com/solutions/datalifecycle-cloud-platform.

59. Gorry, G. A., & Scott Morton, M. S. (1971). A framework for management information systems.

60. Grover, V., Chiang, R. H., Liang, T.-P., & Zhang, D. (2018). Creating Strategic Business Value from Big Data Analytics: A Research Framework. Journal of Management Information Systems, 35(2), 388–423. doi:10.1080/07421222.2018.1451951.

61. Hall, O., & Odd, S. (2019). Business Decisions or Rules – Why not Both? The Views of Three Decision Modelling Experts// International Journal of Information System Modeling and Design. Volume 10. Issue 4.

62. Hao, F., Park, D.-S., Min, S.D., Park, S., 2016. Modeling a big medical data cognitive system with n-ary formal concept analysis. In: Advanced Multimedia and Ubiquitous Engineering. Springer, pp. 721–727.

63. Hazen, B.T., Boone, C.A., Ezell, J.D., Jones-Farmer, L.A., 2014. Data quality for data science, predictive analytics, and big data in supply chain management: an introduction to the problem and suggestions for research and applications. Int. J. Prod. Econ. 154, 72–80.

64. He Y, Zhang S, Tang L (2020) Large scale resource allocation for the internet of things network based on ADMM. IEEE Access 8:57192–57203

65. Horita, F. E., Albuquerque, J. P., Marchezini, V., & Mendiondo, E. M. (2017). Bridging the gap between decisionmaking and emerging big data sources: An application of a model-based framework to disaster management in Brazil// Decision Support Systems, 97.

66. Hossain, M.S., Moniruzzaman, M., Muhammad, G., Ghoneim, A., Alamri, A., 2016. Big data-driven service composition using parallel clustered particle swarm optimization in mobile environment. IEEE Trans. Serv. Comput. 9 (5), 806–817.

67. Houlisa: Clustering algorithm design for eliminating redundant features in large data sets.Mod. Electron. Technol. 41(14), 56–58+62 (2018)

68. Hu, H., Wen, Y., Chua, T.-S., & Li, X. (2014). Toward scalable systems for big data analytics: A technology tutorial. IEEE Access : Practical Innovations, Open Solutions.

69. Huang, L., Zhao, Q., Li, Y., Wang, S., Sun, L., Chou, W., 2017. Reliable and efficient big service selection. Inf. Syst. Front 19 (6), 1273–1282.

70. IBM. Extracting business value from the 4 V's of big data. Retrieved from https://www.ibmbigdatahub.com/infographic/extracting-business-value-4-vs-big-data.

71. Infochimps. (2012). CIOs & Big Data: What Your IT Team Wants

You to Know. http://www.infochimps.com/resources/report-cios-big-data-what-your-it-team-wants-you-to-know-6/.

72. Inoubli, W., Aridhi, S., Mezni, H., Maddouri, M., Nguifo, E.M., 2018. An experimental survey on big data frameworks. Future Generat. Comput. Syst. 86, 546–564.

73. Jamil, H.M., Rivero, C.R., 2017. A novel model for distributed big data service composition using stratified functional graph matching. In: Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics. ACM, p. 34.

74. Janeiro, J., & Eduardsen, J. S. (2018). How can big data affect uncertainty in strategic decision-making? Aalborg University.

75. Jatoth, C., Gangadharan, G., Fiore, U., Buyya, R., 2018. Qos-aware big service composition using mapreduce based evolutionary algorithm with guided mutation. Future Generat. Comput. Syst. 86, 1008–1018.

76. Jiang W, Wang H, Li B (2020) A multi-user multi-operator computing pricing method for Internet of things based on bilevel optimization. Int J Distrib Sens Netw 16(1):155014–155032

77. Jiao J, Sun Y, Wu S (2020) Network utility maximization resource allocation for NOMA in satellite-based internet of things. IEEE Internet Things J 7(4):3230–3242

78. Jin X., Yan L., Liu J. et al. Fault analysis and operation research for enterprise database: taking Oracle Database as an example// Software 2017, 38(10), 178-181.

79. Jing G., Hu C. et al. Software vulnerability detection algorithm for 8031 single chip microcomputer system based on vulnerability knowledge base// J. Beijing Inst. Technol. 2017, 34(4), 371-375

80. Kathiravelu, P., 2017. Software-defined Inter-cloud Composition of Big Services. EMJD-DC, pp. 1–2.

81. Ke H, Wang J, Wang H (2019) Joint optimization of data offloading and resource allocation with renewable energy aware for IoT devices: a deep reinforcement learning approach. IEEE Access 7:179349–179363

82. Kitchenham, B. (2004). Procedures for performing systematic reviews. Keele, UK: Keele University.

83. Kitchin, R., & Lauriault, T. P. (2015). Small data in the era of big data// GeoJournal, 80(4), 463–475. doi:10.1007/s10708-014-9601-7

84. Kolomvatsos, K., & Hadjiefthymiades, S. (2015). An efficient time optimized scheme for progressive analytics in big data// Big Data Research, 2(4), 155–165. doi:10.1016/j.bdr.2015.02.001.

85. Kościelniak, H., & Puto, A. (2015). BIG DATA in decision making processes of enterprises// Procedia Computer Science, 65, 1052–1058. doi:10.1016/j.procs.2015.09.053

86. Kumar, C.A., Singh, P.K., 2014. Knowledge representation using formal concept analysis: a study on concept generation. In: Global Trends in In-

telligent Computing Research and Development. IGI Global, pp. 306–336.

87. Kuznetsov, S.O., Makhalova, T., 2018. On interestingness measures of formal concepts. Inf. Sci. 442, 202–219.

88. Lahmar, F., Mezni, H., 2020. Security-aware multi-cloud service composition by exploiting rough sets and fuzzy fca. Soft Comput. 1–20.

89. Li Q, Ma X, Peng H (2015) Data fusion optimization model of elastic wave in wireless sensor networks. J Comput Inf Syst 11(3):815–822

90. Li X, Tan L, Li F (2019) Optimal cloud resource allocation with cost performance tradeoff based on internet of things. Internet Things J IEEE 6(4):6876–6886

91. Li, B.H., Junhua, C., et al.: Distributed clustering algorithms of attribute graph under multiagent architecture. Comput. Sci. 44(S1), 407–413 (2017)

92. Li, D., Wu, J., Deng, Z., Chen, Z., Xu, Y., 2017. Qos-based service selection method for big data service composition. In: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1. IEEE, pp. 436–443.

93. Li, T. Z., Wang, S. H., & Ma, J. (2014). Study on Fair Definitions and Application Modes of Big Data// Applied Mechanics and Materials.

94. Linjing, W., Lulu, N., Bin, G., et al.: Sparse fractional feature selection clustering algorithms based on entropy weighting in large data. Comput. Appl. Res. 35(8), 59–60 + 69 (2018)

95. Litherland, N. (2017, September 26). Decision-Making Process of Managers. Recupere sur bizfluent: https://bizfluent.com/how-does-5280248-decisionmaking-process-managers.html.

96. Liu M, Li D, Zeng Y (2020a) Combinatorial-oriented feedback for sensor data search in internet of things. IEEE Internet Things J 7(1):284–297

97. Liu S., Li Z., Zhang Y., Cheng X. Introduction of key problems in long-distance learningand training. Mobile Netw. Appl. 2018, 24(1), 1-4. https://doi.org/10.1007/s11036-018-1136-6.

98. Liu X, Ding H, Zhang X (2020b) Rate satisfaction-based power allocation for NOMA-based cognitive Internet of Things. Ad hoc Netw 98(Mar):102063.1-102063.8

99. Liu X, Jia M, Ding H (2020c) Uplink resource allocation for multicarrier grouping cognitive internet of things based on K-means Learning. Ad hoc Netw 96(Jan):1020021–1020029

100. Liu X, Zhang X (2020) NOMA-based resource allocation for cluster-based cognitive industrial internet of things. IEEE Trans Industr Inf 16(8):5379–5388

101. Liu XM (2020) Uplink resource allocation for multicarrier grouping cognitive internet of things based on K-means Learning. Ad Hoc Netw 96:102002–102002

102. Liu, S., Li, Z., Zhang, Y., et al.: Introduction of key problems in long-distance learning and training. Mob. Netw. Appl. 24(1), 1–4 (2019)
103. Lunenburg, F. C. (2010). The decision making process// National Forum of Educational Administration and Supervision Journal, 27(4), 12.
104. Luong NC, Hoang DT, Wang P (2017) Data collection and wireless communication in internet of things (IoT) using economic analysis and pricing models: a survey. IEEE Commun Surv Tutor 18(4):2546–2590
105. Malki, A., Barhamgi, M., Benslimane, S.-M., Benslimane, D., Malki, M., 2014. Composing data services with uncertain semantics. IEEE Trans. Knowl. Data Eng. 27 (4), 936–949.
106. Marr, B. (2015). Big Data: Too Many Answers, Not Enough Questions. Forbes.
107. Martin, T. N. (2016). Smart Decisions: The Art of Strategic Thinking for the Decision Making Process. Springer. doi:10.1057/9781137537003.
108. Mcnarie T, Quist G, Lewinger K (2017) Into the information age: application of internet of things for sewer maintenance optimization. Proc Water Environ Fed 2017(2):499–516
109. Mezni, H., Kbekbi, M., 2019. Reusing process fragments for fast service composition: a clustering-based approach. Enterprise Inf. Syst. 13 (1), 34–62.
110. Mezni, H., Sellami, M., 2017. Multi-cloud service composition using formal concept analysis. J. Syst. Software 134, 138–152.
111. Mezni, H., Sellami, M., 2018. A negotiation-based service selection approach using swarm intelligence and kernel density estimation. Software Pract. Ex. 48 (6), 1285–1311.
112. Mintzberg, H., Raisinghani, D., & Theoret, A. (1976). The structure of" unstructured" decision processes// Administrative Science Quarterly, 21(2), 246–275. doi:10.2307/2392045.
113. Montana, P. J., & Charnov, B. H. (2000). Management Decision-Making: Types and Styles. In P. J. Montana, & B. H. Charnov (Eds.), Business Review Books Management Third Edition (pp. 86-105).
114. Mutin D.I., Kaperko 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.
115. Nayak, A., Pai, M. M., & Pai, R. M. (2016). Prediction models for Indian stock market// Procedia Computer Science, 89, 441–449. doi:10.1016/j.procs.2016.06.096
116. Negulescu, O. (2014). Using a decision-making process model in strategic management. Review of General Management.
117. Ning Z, Wang X, Kong X (2017) A social-aware group formation framework for information diffusion in narrowband internet of things. IEEE

Internet Things J 5:1527–1538

118. OMG. (2016, May). Decision Model and Notation (DMN) V1.1 with change bars.

119. Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., & Belfkih, S. (2017). Big Data technologies: A survey// Journal of King Saud University-Computer and Information Sciences, 30(4), 431-448.

120. Panneerselvam, J., Liu, L., & Hill, R. (2015). An Introduction to Big Data// B. Akhgar, G. B. Saathoff, H. R. Arabnia et al. (Eds.), Application of Big Data for National Security (pp. 3-13). Elsevier. doi:10.1016/B978-0-12-801967-2.00001-X.

121. Panneerselvam, J., Liu, L., & Hill, R. (2015). An Introduction to Big Data// Application of Big Data for National Security (pp. 3–13). Elsevier. doi:10.1016/B978-0-12-801967-2.00001-X

122. Parker, J. S., & Moseley, J. D. (2008). Kepner-Tregoe decision analysis as a tool to aid route selection Part 1// Organic Process Research & Development, 12(6), 1041-1043.

123. Poleto, T., de Carvalho, V. D. H., & Costa, A. P. C. S. (2015, May). The roles of big data in the decision-support process: an empirical investigation// Proceedings of the International conference on decision support system technology (pp. 10-21). Cham: Springer.

124. Poleto, T., de Carvalho, V. D., & Costa, A. P. (2017). The Full Knowledge of Big Data in the Integration of InterOrganizational Information: An Approach Focused on Decision Making// International Journal of Decision Support System Technology, 9(1), 16–31. doi:10.4018/IJDSST.2017010102.

125. Provost, F., & Fawcett, T. (2013). Data science and its relationship to big data and data-driven decision making// Big Data, 1(1), 51–59. doi:10.1089/big.2013.1508 PMID:27447038.

126. Qujie: Research on intelligent parallel clustering method for large data in virtual environment. Comput. Measur. Control 25(6), 257–260 (2017)

127. Ram, S., Zhang, W., Williams, M., & Pengetnze, Y. (2015). Predicting asthma-related emergency department visits using big data// IEEE Journal of Biomedical and Health Informatics, 19(4), 1216–1223.

128. Rehman M, Liew C, Wah T (2015) Mining personal data using smartphones and wearable devices: a survey. Sensors 15(2):4430–4469

129. Rouane-Hacene, M., Huchard, M., Napoli, A., Valtchev, P., 2013. Relational concept analysis: mining concept lattices from multi-relational data. Ann. Math. Artif. Intell. 67 (1), 81–108.

130. Rullo A, Midi D, Serra E (2017) Pareto optimal security resource allocation for internet of things. ACM Trans Inf Syst Secur 20(4):15.1-15.30

131. Safia A, Aghbari Z, Kamel I (2017) Efficient data collection by mobile sink to detect phenomena in internet of things. Information 8(4):123–143

132. Salam A, Javaid Q, Ahmad M (2020) Bioinspired mobility-aware clustering optimization in flying ad hoc sensor network for internet of things:

BIMAC-FASNET. Complexity 20:1–20

133. Schroeck, M., Shockley, R., Smart, J., Romero-Morales, D., & Tufano, P. (2012). Analytics: The real-world use of big data// IBM Global Business Services, 12, 1–20.

134. Sellami, M., Hacid, M.-S., Gammoudi, M.M., 2018. A fca framework for inference control in data integration systems. Distributed Parallel Databases 1–44.

135. Shehu U., Safdar G., Epiphaniou G., 2015. Towards network-aware composition of big data services in the cloud, Int. J. Adv. Comput. Sci. Appl. 6 (10).

136. Shirdastian, H., Laroche, M., & Richard, M.-O. (2017). Using big data analytics to study brand authenticity sentiments: The case of Starbucks on Twitter// International Journal of Information Management.

137. Shuai L., Gelan Y. Advanced Hybrid Information Processing. - Springer, New York. 594 p.

138. Shuai L., Weiling B., Nianyin Z. et al. A fast fractal based compression for MRI images// IEEE Access 2019, 7, 62412-62420

139. Shuai, L., Weiling, B., Nianyin, Z., et al.: A fast fractal based compression for MRI images. IEEE Access 7, 62412–62420 (2019)

140. Simon, H. A. (1960). The executive as decision maker// The new science of management decision, 1-7.

141. 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.

142. 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.

143. 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.

144. 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.

145. Stella K, Giorgos A, Symeon P (2015) On the optimization of a

probabilistic data aggregation framework for energy efficiency in wireless sensor networks. Sensors 15(8):19597–19617

146. Sun Z, Xing X, Wang T (2019) An optimized clustering communication protocol based on intelligent computing in informationcentric internet of things. IEEE Access 7(99):28238–28249

147. Sun, G., Liu, S. (eds.): ADHIP 2017. LNICST, vol. 219. Springer, Cham (2018). https://doi. org/10.1007/978-3-319-73317-3

148. Taherkordi, A., Eliassen, F., Horn, G., 2017. From iot big data to iot big services. In: Proceedings of the Symposium on Applied Computing. ACM, pp. 485–491.

149. Taleb, I., Dssouli, R., Serhani, M.A., 2015. Big data pre-processing: a quality framework. In: 2015 IEEE International Congress on Big Data. IEEE, pp. 191–198.

150. Taleb, I., El Kassabi, H.T., Serhani, M.A., Dssouli, R., Bouhaddioui, C., 2016. Big data quality: a quality dimensions evaluation. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld). IEEE, pp. 759–765.

151. Tan L., Zhong H. Operation and maintenance practice of data center construction based on cloud computing model// Inf. Comput. (Theor. Ed.) 2018, 408(14), 21-23.

152. Tao H, Miaowang Z, Lijuan Z (2018) A channel-aware duty cycle optimization for node-to-node communications in the internet of medical things. Int J Parallel Prog 48:264–279

153. Taylor, J. (2016). Bringing Clarity to Data Science Projects with Decision Modeling: A Case Study. International Institute for Analytics.

154. Taylor, J. (2016). Decision Modeling with DMN. Decision Management Solutions.

155. Taylor, J. (2017). Framing Analytic Requirements. Decision Management Solutions.

156. Turet, J. G., & Costa, A. P. (2018). Big Data Analytics to Improve the Decision-Making Process in Public Safety: A Case Study in Northeast Brazil// Proceedings of the International Conference on Decision Support System Technology. Academic Press. doi:10.1007/978-3-319-90315-6_7.

157. Valtchev, P., Missaoui, R., Godin, R., 2004. Formal concept analysis for knowledge discovery and data mining: the new challenges. In: International Conference on Formal Concept Analysis. Springer, pp. 352–371.

158. Vavilis, S., Petkovi, M., Zannone, N., 2014. Data leakage quantification. In: IFIP Annual Conference on Data and Applications Security and Privacy. Springer, pp. 98–113.

159. Vavilis, S., Petkovi, M., Zannone, N., 2016. A severity-based quantification of data leakages in database systems. J. Comput. Secur. 24 (3), 321–

345.

160. Wang F. Summary of technological innovation of broadcasting operation and maintenance data collection and process monitoring and processing system// Modern Telev. Technol. 2017, 34 (5), 138-141.

161. Wang M, Xu C, Chen X (2019) Design of multipath transmission control for information-centric internet of things: a distributed stochastic optimization framework. IEEE Internet Things J 6(6):9475–9488

162. Wang, H., Xu, Z., Fujita, H., & Liu, S. (2016). Towards felicitous decision making: An overview on challenges and trends of Big Data// Information Sciences, 367, 747–765. doi:10.1016/j.ins.2016.07.007.

163. Wang, R.Y., Strong, D.M., 1996. Beyond accuracy: what data quality means to data consumers. J. Manag. Inf. Syst. 12 (4), 5–33.

164. Wang, X., Yang, L.T., Feng, J., Chen, X., Deen, M.J., 2016. A tensor-based big service framework for enhanced living environments. IEEE Cloud Comput. 3 (6), 36–43.

165. Ward, J. S. (2013). Undefined by data: a survey of big data definitions.

166. Ward, J. S., & Barker, A. (2013). Undefined by data: a survey of big data definitions.

167. Wei, L., Zhao, Q., Shu, H., 2018. Design of manufacturing big data access platform based on soa. In: 2018 IEEE 4th International Conference on Computer and Communications (ICCC). IEEE, pp. 1841–1845.

168. Xiang M, Wang D (2019) Performance analysis and optimization for coverage enhancement strategy of Narrow-band Internet of Things. Future Gener Comput Syst 101(C):434–443

169. Xiaoyan, T.: A large data text clustering algorithm based on word embedding and density peak strategy. Innov. Appl. Sci. Technol. 6, 90–90 (2017)

170. Xiaoyu, C., Xiaojing, L., Haiying, M.: A fast automatic clustering algorithm for large data. Comput. Appl. Res. 34(9), 2651–2654 (2017)

171. Xu S, Wang X, Yang G (2020) Routing optimization for cloud services in SDN-based internet of things with TCAM capacity constraint. J Commun Netw 22(2):145–158

172. Xu, X., Motta, G., Wang, X., Tu, Z., Xu, H., 2018. A new paradigm of software service engineering in the era of big data and big service. Computing 100, 353–368.

173. Xu, X., Sheng, Q.Z., Zhang, L.-J., Fan, Y., Dustdar, S., 2015. From big data to big service. Computer (7), 80–83.

174. Ya´nez W, Mahmud R, Bahsoon R, Zhang Y, Buyya R (2020) Data allocation mechanism for internet-of-things systems with blockchain. IEEE Internet Things J 7(4):3509–3522

175. Yang Y., Zhang S., Kang Q. Design of intelligent early warning system based on grid operation and maintenance data// Inner Mongolia Electric Power Technol. 2017, 35(4), 20-23.

176. Yi, M., Ting, X., Shaobin, L.: Research on NoSQL distributed large data mining method in complex attribute environment. Sci. Technol. Eng. 17(09), 244–248 (2017)

177. Yin X, Li S, Lin Y (2019) A novel hierarchical data aggregation with particle swarm optimization for internet of things. Mobile Netw Appl 24(6):1994–2001

178. Ylijoki, O., & Porras, J. (2016). Perspectives to definition of big data: A mapping study and discussion// Journal of Innovation Management, 4(1), 69–91. doi:10.24840/2183-0606_004.001_0006.

179. Zhang, H., Zhang, L., Cheng, X., & Chen, W. (2016). A novel precision marketing model based on telecom big data analysis for luxury cars// Proceedings of the 2016 16th International Symposium on Communications and Information Technologies (ISCIT) (pp. 307-311). IEEE.

180. Zhao C., Sun L., Gao X. et al. Discussion on protection intelligent operation andmaintenance technology based on source data maintenance mechanism// Power Grid Clean Energy 2017, 52(09), 82-87.

181. Zheng P., Shuai L., Arun S., Khan M. Visual attention feature (VAF): a novel strategy forvisual tracking based on cloud platform in intelligent surveillance systems// J. Parallel Distrib. Comput. 2018, 120, 182-194.

182. Zheng, Z., Zhang, Y., Lyu, M.R., 2010. Distributed QoS evaluation for real-world web services. In: 2010 IEEE International Conference on Web Services. IEEE, pp. 83–90.

183. Zhong L., Guo T., Zhang M. Design and implementation of online ceramic mineralresources management knowledge base based on LINGO// Intell. Comput. Appl. 2018, 36(1), 68-71.

184. Zhou, L., Chen, H., Yu, T., Ma, J., Wu, Z., 2008. Ontology-based scientific data service composition: a query rewriting-based approach. In: AAAI Spring Symposium: Semantic Scientific Knowledge Integration, pp. 116–121.

185. Zhou, X., Jin, Y., Zhang, H., Li, S., & Huang, X. (2016). A map of threats to validity of systematic literature reviews in software engineering// Proceedings of the 2016 23rd Asia-Pacific Software Engineering Conference (APSEC) (pp. 153-160). Academic Press.