УДК 004.67
DOI: 10.20295/2413-2527-2024-137-45-55

Метод перехода от хранилищ данных к озерам данных геоинформационных систем на основе Лямбда-архитектуры

магистр **Р. Абу Хасан**
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия

А. Б. Кириенко
Военно-космическая академия имени А. Ф. Можайского
Санкт-Петербург, Россия

д-р техн. наук **А. Д. Хомоненко**
Петербургский государственный университет путей сообщения Императора Александра I
Военно-космическая академия имени А. Ф. Можайского
Санкт-Петербург, Россия

Аннотация. **В статье рассматривается переход от традиционных хранилищ данных к озерам данных в геоинформационных системах с использованием Лямбда-архитектуры. Приводится обзор основных этапов перехода, включая планирование, сбор и обработку данных, запросы данных, аналитику данных и управление метаданными. Особое внимание уделяется взаимодействию озер данных и ГИС, а также примерному коду обработки больших данных на основе Лямбда-архитектуры. Рассматриваются преимущества использования озер данных в ГИС и возможности интеграции современных технологий обработки данных.**

Ключевые слова: **озера данных, хранилища данных, Лямбда-архитектура, геоинформационные системы, метаданные, обработка больших данных, интеграция данных, аналитика данных, переход от хранилищ данных.**

Для цитирования: **Абу Хасан Р., Кириенко А. Б., Хомоненко А. Д. Метод перехода от хранилищ данных к озерам данных геоинформационных систем на основе Лямбда-архитектуры // Интеллектуальные технологии на транспорте. 2024. № 1 (37). С. 45–55. DOI: 10.20295/2413-2527-2024-137-45-55**

Введение

Хранилища данных призваны обеспечить физическую интеграцию баз данных. Это основа OLAP-приложений (оперативная аналитическая обработка данных) и бизнес-аналитики, на них построена ориентированная на данные стратегия предприятия.

Хранилища данных играют важную роль в обеспечении физической интеграции баз данных, однако с появлением больших данных и разнообразных источников информации возникают проблемы с их обработкой.

Когда хранилища данных только появились (в 1980-х годах), данные предприятия хранились в оперативных OLTP-базах (управления данными о транзакциях. Сегодня все больше и больше полезных данных поступает из разнообразных источников больших данных: журналов веб-серверов, социальных сетей и электронной почты. В результате выявились ***недостатки традиционных хранилищ данных:***

• определение схемы при записи. Обычно хранилище данных опирается на реляционную СУБД, и его структура описывается реляционной схемой. В реляционных СУБД принят подход, который недавно стали называть schema-on-write в противоположность schema-on-read (см. далее). В этом случае данные записываются в базу в фиксированном формате, определенном схемой;

• длительный процесс разработки. Разработка хранилища данных может длиться годами. Основная причина в том, что требуется заранее точно определить и смоделировать необходимые данные;

• обработка OLAP-задач. Хранилище данных обычно оптимизировано для рабочих нагрузок типа OLAP, когда аналитики интерактивно опрашивают данные по различным измерениям, например с помощью кубов данных. Недавно появились OLAP-приложения, которым нужен доступ к оперативным данным в реальном масштабе времени, что довольно трудно поддержать в хранилищах данных;

• трудоемкая разработка с применением ETL (Extract Transform Load). Для интеграции гетерогенных источников данных в глобальную схему необходимы сложные ETL-программы для очистки, преобразования и обновления данных. По мере диверсификации источников разрабатывать такие программы становится все труднее.

В последние годы для работы с большими данными и их аналитики активно применяется концепция озер данных [1].

Переход к озерам данных в геоинформационных системах на основе Лямбда-архитектуры представляет собой перспективное решение для эффективной обработки и анализа данных.

Характеристика озер данных

Озеро данных представляет собой центральное хранилище большого объема данных компании, предприятия, в естественном виде, а также измененные данные, с возможностью последующего их извлечения и анализа разными пользова-

телями и системами. Озеро данных может включать структурированные данные из реляционных баз данных (строки и столбцы), полуструктурированные данные (CSV, журналы, XML, JSON), неструктурированные данные (электронные письма, документы, PDF-файлы) и двоичные данные (изображения, аудио, видео).

Структурированные данные — это организованные данные, которые следуют за фиксированной схемой, такие как таблицы баз данных, где есть четкие определения столбцов, строк и таблиц. Эти данные легко искать и организовывать, так как они следуют строгим правилам. Примеры включают данные из реляционных баз данных (SQL), онлайн-форм и транзакционных систем.

Полуструктурированные данные — это данные, которые не подходят полностью под модель реляционной базы данных, но содержат теги или другие маркеры для отдельных элементов данных. Данные могут быть представлены в таких форматах, как JSON, XML и CSV, где есть некоторый метаязык для обозначения свойств данных, но не требуется схема данных в традиционном смысле.

Неструктурированные данные — это информация, которая не имеет определенного формата или структуры, например, текстовые документы, изображения, видео, аудио, электронные письма и социальные медиа. Обработка и анализ таких данных требуют более сложных подходов, таких как машинное обучение, текстовый анализ и компьютерное зрение.

Как и хранилище данных, озера данных можно использовать для OLAP-приложений и бизнес-аналитики, а также для пакетного или оперативного анализа данных с применением технологий больших данных.

По сравнению с хранилищами данных ***озеро имеет следующие преимущества:***

• определение схемы при чтении. Термином schema-on-read обозначают подход к анализу больших данных, когда во главу угла ставятся загруженные данные, как в Hadoop. В этом случае данные загружаются как есть, в своем «родном» формате: например, в файловую систему Hadoop HDFS (Hadoop Distributed File System). А уже потом, во время чтения данных, на них накладывается схема для выделения представляющих интерес полей. Таким образом, данные можно опрашивать в естественном формате. Это резко повышает гибкость, поскольку в озеро в любой момент можно добавить новые данные. Однако требуется больше усилий для написания кода, применяющего к данным схему. Например, его можно включить в функцию Map каркаса Map Reduce. Разбор данных также приходится проводить во время выполнения запроса;

• обработка различных рабочих нагрузок. Программный стек управления большими данными поддерживает разные методы доступа к одним и тем же данным: например, пакетный анализ с помощью каркаса типа Map Reduce, интерактивные OLAP-приложения или бизнес-аналитика с помощью каркаса типа Spark, анализ в реальном времени с помощью каркасов потоковой обработки данных. Агрегируя различные каркасы, озеро данных может поддержать обработку рабочих нагрузок разных типов;

• экономически эффективная архитектура. Опираясь на кластеры без разделения ресурсов и на программы с открытым исходным кодом для реализации программного стека управления большими данными, озеро данных обеспечивает отличные показатели соотношения стоимости и производительности и отдачи на капитал.

Озеро данных предоставляет следующие основные возможности [1]:

• сбор полезных данных: исходных, преобразованных, поступающих из внешних источников и так далее;

• предоставление возможности пользователям из различных подразделений предприятия исследовать данные и обогащать их метаданными;

• использование для доступа к данным различных методов: пакетных, интерактивных, в режиме реального времени и так далее;

• осуществление руководства данными, обеспечение безопасности, управления данными и задачами.

Проблемы переноса данных из корпоративных хранилищ в озера данных

Переход от традиционных хранилищ и развертывание озера данных, обусловленный несколькими тенденциями и потребностями (объем и разнообразие данных, гибкость в работе с данными, масштабируемость хранилищ, многофункциональность, оптимизация затрат, скорость доступа к данным, инновации инструментов и технологий) сопряжен с многочисленными препятствиями и проблемами, которые потребуется решить [2]:

• разнообразие форматов данных. Данные могут храниться в различных форматах, таких как CSV, Excel, текстовые файлы, реляционные базы данных и NoSQL. Это может усложнить процесс переноса и привести к потере качества данных;

• неоднородность данных. Данные из различных источников могут быть структурированными, полуструктурированными или неструктурированными. Это делает процесс интеграции и преобразования данных более сложным;

• защита данных. При переносе данных из внутренних систем компании могут возникнуть проблемы с соблюдением правил безопасности и конфиденциальности данных;

• масштабируемость. При увеличении объема данных процесс переноса может стать слишком сложным и ресурсоемким;

• производительность и задержки. Перенос данных может привести к снижению производительности и увеличению задержек в работе корпоративных систем;

• сложность управления. Процесс переноса данных требует тщательного планирования, управления и мониторинга;

• интеллектуальная собственность. При переносе данных нужно учитывать вопросы интеллектуальной собственности и авторские права.

Для решения этих проблем необходимо использовать передовые технологии и инструменты для переноса данных, интеграции данных и развертывания систем управления данными.

Также одним из основных препятствий являются непомерные расходы, связанные с хранилищами данных. Затраты на их внедрение и обслуживание высоки.

Облачные решения Data Lake предлагают упрощенное развертывание, хотя и ***могут повлечь за собой значитель-***

ные затраты. Некоторые платформы, такие как Hadoop, имеют открытый исходный код и, следовательно, не требуют никакой оплаты. Однако выполнение и надзор могут потребовать дополнительного времени и более квалифицированного персонала.

Проблема управления — еще одна проблема [2]. Управление озером данных влечет за собой выполнение сложных обязанностей, таких как обеспечение способности инфраструктуры хоста справляться с ростом озера данных и решение проблем избыточности данных и безопасности. Это создает серьезные препятствия даже для опытных инженеров.

Кроме того, существует **потребность в увеличении числа экспертов в предметной области и инженеров**, обладающих подлинными навыками создания озер данных и контроля за ними. Может возникнуть недостаток специалистов по обработке данных и инженеров по обработке данных. Еще одним фактором, который следует принимать во внимание, является длительный срок, необходимый для достижения полной функциональности и бесшовной интеграции с инструментами документооборота и аналитики [3].

Безопасность данных является серьезной проблемой в контексте озер данных, аналогичной хранилищам данных. Для обеспечения соответствия правилам управления данными и защиты данных в озере данных необходимо внедрить определенные меры безопасности, используя опыт специалистов по кибербезопасности и применяя инструменты обеспечения безопасности. Другой существенной трудностью являются вычислительные ресурсы и рост вычислительной мощности. Это связано с тем, что объем данных растет беспрецедентными темпами, превосходящими рост вычислительной мощности. Современные компьютеры недостаточно способны обрабатывать их и управлять ими одновременно из-за нехватки питания. Аналогичным образом платформы данных с открытым исходным кодом сталкиваются с многочисленными фундаментальными проблемами, связанными с управлением озерами данных, которые являются чрезмерно дорогостоящими. Кроме того, устранение этих значительных различий в экспертных знаниях требует использования значительных вычислительных ресурсов.

Для повышения качества озера данных крайне ***важно обновить методы***, с помощью которых создаются и контролируются озера данных. Требуется обеспечить использование облачных вычислений вместо построения сложных систем хранения данных на специально созданной инфраструктуре [4].

Ключевой особенностью архитектуры озера данных является ее ***способность эффективно и без особых усилий получать различные формы данных***, такие как потоковые данные в реальном времени с локальных платформ хранения, структурированные данные, создаваемые и обрабатываемые мэйнфреймами и хранилищами данных, а также неструктурированные или полуструктурированные данные. В процессе приема данных используется значительный уровень параллелизма и минимальная задержка из-за необходимости взаимодействия с внешними источниками данных с ограниченной пропускной способностью. При этом не требуется проводить тщательную проверку загруженных данных. Возможно использовать поверхностный анализ загруженных данных и их метаданных для поддержания фундаментальной организации загружаемых наборов данных. На этапе извлечения данных в data lake management исходные данные преобразуются в предварительно установленную модель данных.

Вместо того чтобы выполнять извлечение отдельных файлов последовательно, можно использовать информацию, полученную в ходе предыдущих процессов извлечения. Существует ограниченное количество исследований по этапу очистки озера данных, и в литературе рассматривается лишь несколько технологий, таких как CLAMS [5].

Для извлечения данных из озера существует два способа: поиск на основе запросов, когда пользователь инициирует поиск, вводя запрос для получения конкретных данных, и поиск на основе данных, когда пользователь исследует озеро данных, используя график связей или иерархическую структуру, чтобы найти интересующие данные [6]. Одним из возможных способов является интеграция методов, основанных на анализе или контексте, которые предполагают расширение набора данных соответствующей информацией и контекстуальными деталями для облегчения задач обучения.

Другая область исследований предполагает изучение ***применения машинного обучения*** в хранилищах данных. В настоящее время проводятся многочисленные исследования с особым упором на использование машинного обучения с целью организации и обнаружения наборов данных. Работа по обнаружению наборов данных часто включает в себя идентификацию «похожих» признаков, которые были получены из данных, метаданных и других источников. Затем эти атрибуты можно использовать в сочетании с задачами классификации или кластеризации.

В нескольких недавних исследованиях использовались методы машинного обучения, включая классификатор K-ближайших соседей (KNN) [7] и модель логистической регрессии для оптимизации коэффициентов признаков [8]. В ближайшие годы процесс обнаружения наборов данных, по прогнозам, дополнят более совершенное глубокое обучение и связанные с ним сложные подходы машинного обучения.

Управление метаданными

Управление метаданными является важнейшей обязанностью в озере данных, поскольку в озере данных отсутствуют всеобъемлющие каталоги данных [2]. Метаданные играют критически важную роль, делая возможным управление, поиск, безопасность и анализ данных. Озера данных обычно содержат огромное количество разнородных данных, которые собираются из различных источников и хранятся в их исходном формате. Метаданные представляют собой данные о данных, они помогают понять контекст, структуру, зависимости и доступность содержимого озера данных.

Отсутствие четких метаданных для наборов данных, особенно на протяжении всего процесса обнаружения и очистки данных, увеличивает риск превращения озера данных в информационное болото.

При работе с метаданными в озерах данных необходимо придерживаться следующих аспектов:

• каталогизация. Метаданные используются для каталогизации активов данных в озере данных, что позволяет пользователям легко находить нужные данные для анализа;

• управление данными. Организация жизненного цикла данных, включая версионирование, архивирование и удаление данных;

• безопасность и управление доступом. Метаданные определяют, кто может получить доступ к определенным данным, включая информацию о разрешениях и политиках безопасности;

• линейность/связь данных (Lineage). Метаданные предоставляют информацию о происхождении данных, их перемещениях и преобразованиях, что важно для отслеживания их источника, состояния и изменений;

• схемы и структуры данных. Они включают информацию о схемах и структурах данных, что позволяет пользователям понимать, как данные организованы и как их можно использовать;

• интеграция данных. Метаданные помогают при интеграции данных, так как они могут содержать информацию об отношениях и зависимостях между разными наборами данных;

• поиск и открытие. Мощные средства поиска могут использовать метаданные для повышения эффективности поиска и открытия нужных данных в пределах озера данных;

• качество данных. Метаданные могут содержать информацию о качестве данных, помогая определить их надежность и точность;

• управление правилами и политиками. Политики обработки данных, такие как правила обеспечения соответствия GDPR или HIPAA, могут быть включены в метаданные;

• аудит и соответствие. Метаданные позволяют отслеживать и подтверждать соответствие данных различным стандартам и правилам, важным для регулятивной среды.

GDPR (General Data Protection Regulation) и HIPAA (Health Insurance Portability and Accountability Act) — два законодательства, которые устанавливают правила для защиты персональных данных в Европейском союзе и США соответственно. Инструменты управления метаданными, встроенные в системы озер данных, такие как Apache Atlas, Alation, Collibra и AWS Glue, предоставляют способы для автоматизации управления метаданными, повышения их доступности и улучшения эффективности работы с данными.

Следовательно, необходимо **извлекать важные метаданные** из источников данных и обеспечивать эффективное хранение и извлечение метаданных. В сфере управления метаданными все еще существует множество областей, требующих дополнительного изучения. Они включают извлечение информации из озера данных и ее интеграцию в уже существующие базы знаний.

Можно организовать аналитическую обработку данных, размещенных в озере данных, с помощью инструментальных средств и методов бизнес-аналитики (рис. 1) [2]. В частности, для многомерного анализа данных можно применить технологию OLAP, а для определения тональности текста можно применить технологию текстовой аналитики (Text mining). Для проведения интеллектуального анализа данных может использоваться технология интеллектуального анализа (data mining).

Другим важным фактором является **управление версиями данных**, которое включает в себя включение новых версий старых файлов в динамическое хранилище данных [8]. Учитывая, что процедуры, связанные с управлением версиями, могут влиять на каждый этап хранилища данных, обращение к этому элементу имеет первостепенное значение. Примером крупномасштабной технологии управления версиями наборов данных является DataHub [24], который предлагает git-подобный интерфейс для управления процессами создания версий, ветвления и объединения.

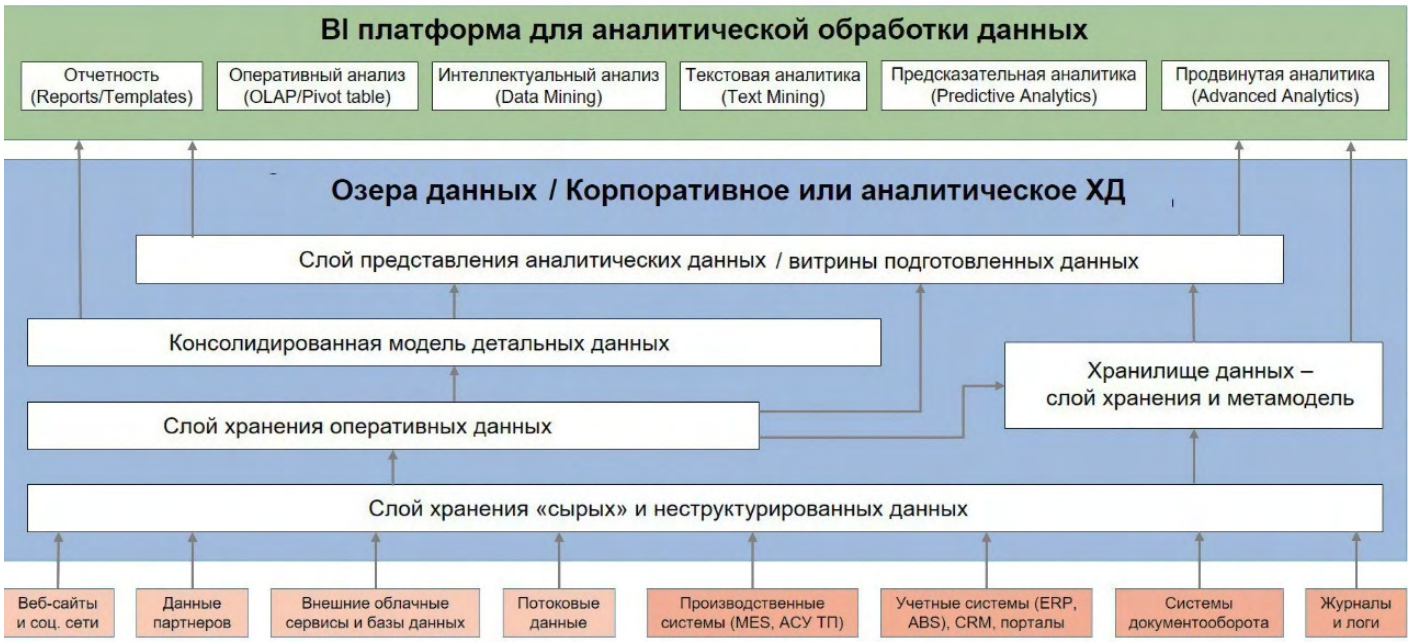

Рис. 1. Инструментальные средства и методы бизнес-аналитики

Наконец, наблюдается растущая тенденция в архитектуре управления данными, известная как Data Lakehouse, которая сочетает **адаптивность** озер данных с функциональными возможностями управления данными хранилища данных. Эту технологию можно рассматривать как **метод хранения всех типов данных** (неструктурированных, полуструктурированных и структурированных) при сохранении высоких стандартов качества данных и управления данными, часто связанных с хранилищем данных [9]. Хранилище данных такого рода может обладать способностью улучшать управление данными, сводить к минимуму передачу и дублирование данных, оптимизировать использование времени и достигать этих преимуществ даже при использовании упрощенной схемы. Ожидается, что концепция хранилища данных Data Lakehouse станет многообещающей областью исследований в области управления данными в ближайшие годы.

Этапы перехода от хранилища данных к озеру данных

Процесс организации перехода от хранилища данных к озеру данных можно представить следующими этапами [9]:

Планирование и подготовка:

• определение целей перехода. Четко сформулировать причины перехода на озеро данных и желаемые результаты;

• оценка текущей среды обработки данных. Оценка источников данных, формата хранения, качества данных и последовательности данных в текущем хранилище данных;

• определение требований к миграции данных. Определение типов данных, объемов и частоты обновления данных;

• выбор технологии. Выбор платформы Data Lake, соответствующей потребностям и бюджету организации;

• разработка стратегии миграции. Планирование процесса миграции данных, включая извлечение, преобразование и загрузку данных (ETL) или методы приема данных.

Извлечение и репликация данных:

• извлечение данных из хранилища данных. Подключение к хранилищу данных и извлечение соответствующих данных с помощью SQL, инструментов ETL или API;

• преобразование данных в формат, удобный для озера данных. Очистка, стандартизация и изменение формы данных в соответствии со структурой и схемой озера данных;

• загрузка или репликация данных в озеро данных. Копирование или потоковая передача преобразованных данных в озеро данных, обеспечивающее целостность и последовательность данных.

Доступ к данным и аналитика:

• настройка доступа к данным и безопасности. Реализация контроля доступа к данным, управления доступом и шифрования данных для защиты конфиденциальной информации;

• определение политики управления данными. Определение стандартов качества данных, отслеживание происхождения и методов управления метаданными;

• разработка конвейеров анализа данных. Применение механизмов обработки данных, таких как Apache Spark или Data Lake, для анализа и извлечения информации из озера данных.

Мониторинг и техническое обслуживание:

• постоянный мониторинг качества данных. Внедрение проверки качества данных для обеспечения целостности и непротиворечивости данных;

• управление объемом и ростом данных. Отслеживание роста данных и внедрение политики хранения данных для оптимизации затрат на хранение;

• решение проблем с производительностью. Оценивание и оптимизация шаблонов доступа к данным, структуры данных и конвейеров обработки для поддержания производительности;

• регулярное обновление и поддержка инфраструктуры озера данных. Обновление платформы и инструментов озера данных для обеспечения совместимости и безопасности.

Озера данных и геоинформационные системы

Взаимодействие озер данных и геоинформационных систем (ГИС) целесообразно организовывать для анализа, визуализации и управления географической информацией. При этом их ***взаимодействие определяется следующими операциями*** [10–12]:

• геокодирование. Озера данных могут содержать большое количество негеокодированной информации, такой как имена компаний, адреса, точки продаж и т.д. Геоинформационные системы могут использовать эту информацию для привязки к географическим координатам, что позволяет визуализировать и анализировать данные на карте;

• загрузка данных. Геоинформационные системы позволяют загружать данные из различных источников, включая озера данных. Это может включать загрузку данных о погоде, транспортных потоках, демографии и других параметрах, которые могут быть полезны для анализа и визуализации;

• анализ данных. Озера данных часто содержат большое количество данных, которые не были проанализированы или обработаны. Геоинформационные системы предоставляют инструменты для анализа этих данных, включая пространственный анализ, который позволяет исследовать взаимосвязи между различными географическими объектами и явлениями;

• визуализация данных. Геоинформационные системы обеспечивают визуализацию данных на карте, что делает их более понятными и наглядными. Они позволяют отображать различные типы данных, такие как точки, линии, полигоны и так далее, а также предоставлять информацию о них в виде таблиц, графиков и других форматов;

• интеграция с другими системами. Геоинформационные системы часто интегрируются с другими системами, такими как системы управления базами данных, системы мониторинга и контроля и другими. Это позволяет более эффективно использовать информацию и обмениваться ею между различными системами.

ГИС активно наполняются данными с помощью мобильных наземных, водных, воздушных средств, космических комплексов дистанционного зондирования и устройств сбора данных. Так, только в космическом пространстве находится огромное количество КА, собирающих различную информацию о земной поверхности, часть которой затем используется в базах ГИС.

Многообразие данных, высокая степень динамичности, вызванная необходимостью поддержания их актуальности, приводит к возникновению проблем, присущих традиционным хранилищам данных, связанным с интеграцией пространственных данных, их обработкой и хранением.

В настоящее время на рынке ГИС существует не менее 100 коммерческих систем и более 300 свободно распространяемых программных комплексов для работы с пространственной информацией. При этом не существует единых форматов хранения данных и унифицированных средств для интеграции систем друг с другом.

Исходя из концепции единого информационного пространства, ГИС должна позволять всем ее пользователям взаимодействовать с единой системой пространственных баз данных, дать возможность всем потребителям пространственных данных доступа к актуальным данным для их просмотра, использования и применения инструментов аналитики независимо от территориальной расположенности.

В ряде работ, например [20], предлагается использовать подход к организации хранения и обработки пространственных данных, основанный на концепции озер данных. При этом целесообразным, по нашему мнению, является организация перехода от традиционных хранилищ данных к озерам данным на основе Лямбда-архитектуры обработки больших данных. Охарактеризуем кратко этот способ.

Способ организации перехода от хранилищ данных к озерам данных в геоинформационных системах на основе Лямбда-архитектуры

Переход от традиционных хранилищ данных (Data Warehouse) к озерам данных (Data Lakes) в контексте геоинформационных систем (ГИС) является частью более широкой тенденции в области больших данных и аналитики.

Интеграция данных в реальном времени часто требует комплексной архитектуры, которая позволяет оперативно обрабатывать и анализировать большие объемы разнообразной информации. Озера данных (Data Lakes) в сочетании с Лямбда-архитектурой (Lambda Architecture) предоставляют один из подходов к решению этой задачи.

Лямбда-архитектура при этом играет ключевую роль в управлении и анализе потоковых и пакетных данных, особенно в системах, обрабатывающих большое количество информации с высокой скоростью и из разнообразных источников.

Основой Лямбда-архитектуры является выполнение произвольных функций над распределенными наборами данных в реальном времени, а также сочетание возможностей пакетной обработки и обработки в реальном времени для балансировки задержки данных, пропускной способности и устойчивости к сбоям. Для выполнения этой задачи используются несколько средств и техник для создания полной системы обработки больших данных. Лямбда-архитектура решает проблему выполнения произвольных функций параллельно с распределенными данными в реальном времени, представляя трехуровневую структуру, состоящую из слоя пакетной обработки, слоя обработки в реальном времени и слоя обслуживания, от сбора данных до анализа данных, с визуализацией и обратной связью (рис. 2) [19].

Лямбда-архитектура в ГИС нацелена на обеспечение возможности быстрого анализа специальных данных. В ней сочетаются два подхода к обработке данных:

1. *Пакетная обработка* (Batch Layer) отвечает за обработку больших объемов накопленных данных. Это «долговременная память» системы, позволяющая выполнять комплексные запросы и анализ данных за длительные периоды времени. Задачи могут выполняться с задержкой (не в реальном времени), но позволяют провести полноценный анализ исторических данных.

2. *Потоковая обработка* (Speed Layer) предназначена для обработки данных в режиме реального времени. Этот слой обеспечивает быстрые ответы на запросы, используя последние данные и минимизируя задержку между событием и возможностью анализировать это событие.

Исходя из результатов пакетной и скоростной обработки, слой обслуживания формирует данные таким образом, чтобы они были доступны для запросов пользователей, часто в виде предварительно агрегированных представлений.

Архитектуру для геопространственных данных можно представить пятью последовательными этапами:

1. *Генерация геопространственных данных.* Геопространственные данные создаются различными датчиками, сенсорами, средствами дистанционного зондирования Земли

Рис. 2. Лямбда-архитектура для работы с геопространственными данными

и представляют собой картографические данные, спутниковые изображения, данные GPS-трекеров, погодные данные и др. Особенности распределенной архитектуры Lambda позволяют не только обрабатывать, но и генерировать новые геоданные на основе аналитики, моделирования и предсказаний, используя алгоритмы машинного обучения и искусственного интеллекта.

2. *Сбор данных.* Данные перед размещением в озере проходят процедуру предварительной «очистки» и разделения специальными средствами. К ним относятся, например, Flink и Samza (фреймворки распределенной потоковой обработки), Flume (сервис передачи логов Hadoop), Kafka (фреймворк, обеспечивающий работу в режиме реального времени конвейеров данных и приложений для обработки потоков) и Sqoop (платформа для интеграции данных из СУБД SQL и NoSQL в Hadoop). Все данные должны сопровождаться метаданными, отражающими их особенности (источник, типы, форматы, пространственная привязка, уровни доступа).

3. *Хранение и обработка данных.* Хранение пакетных данных может быть организовано с помощью HDFS (применяется примерно в 75% озер данных). HDFS — это распределенная система хранения, обладающая высокой степенью масштабируемости и возможностью обработки всех типов данных. HDFS хорошо подходит для хранилищ без схем и хранилищ больших объемов неструктурированных данных. Предобработка выполняется с помощью MapReduce, который хорошо подходит для очень больших данных, но менее эффективен для быстрых, потоковых данных. Для них используются альтернативные фреймворки обработки, например Apache Spark. Spark особенно подходит для обработки в режиме реального времени. Точно так же Apache Flink и Apache Storm подходят и для обработки данных в реальном времени.

4. *Запросы данных.* Данные могут быть доступны через классические языки запросов, такие как SQL для реляционных СУБД, JSONiq для MongoDB, XQuery для СУБД XML или SPARQL для ресурсов RDF. Spark SQL и SQL++ могут использоваться как для запросов в реляционные СУБД, так и к полуструктурированным данным в формате JSON. Можно использовать Apache Phoenix для автоматического преобразования SQL-запросов в язык запросов NoSQL или, например, Apache Drill, который позволяет объединить данные из нескольких систем хранения. Apache Hive может применяться для обработки данных и предоставляет SQL-подобный интерфейс для удобного анализа данных, хранящихся в HDFS или других хранилищах данных, таких как Apache HBase. Apache Impala — это инструмент для массовой параллельной обработки запросов, предназначенный для обработки запросов на чтение/запись в режиме реального времени в HDFS или HBase.

5. Аналитика данных. Например, для выполнения анализа больших данных может применяться инструмент Radoop, который интегрируется с экосистемой Hadoop. Он предлагает интерфейс, позволяющий выполнять сложные аналитические процедуры на больших данных без необходимости писать код на Hadoop или его компонентов, таких как Pig и Hive. Для визуализации данных может быть применен Tableau, который используется для превращения сырых данных в легко понимаемые формы, такие как графики, диаграммы и интерактивные панели управления. Он также позволяет исследовать и анализировать данные, создавая интерактивные и разделяемые отчеты.

Для **реализации перехода к озерам данных в ГИС** требуется выполнить следующие шаги:

- определение новой стратегии управления данными;
- выбор подходящих технологий для создания и управления озером данных;
- обеспечение правильного уровня безопасности и конфиденциальности;
- привлечение специалистов, способных работать с новой архитектурой данных.

Код обработки больших данных ГИС на основе Лямбда-архитектуры

На рисунке 3 приведен примерный контур кода, который использует Лямбда-архитектуру для обработки данных геоинформационной системы.

Код написан на Python и использует библиотеку GeoPandas для упрощения обработки геопространственных данных. Код предполагает наличие системы управления данных (например, Apache Hadoop для пакетной обработки и Apache Kafka для потоковой обработки) и распределенной системы хранения данных. Код обеспечивает обработку геопространственных данных в архитектуре Лямбда, однако он сильно упрощен и предназначен только для демонстрации концепций. Отметим, что в реальных системах потребуется учитывать множество дополнительных аспектов, включая управление ошибками, масштабирование, безопасность и оптимизацию производительности.

Заключение

Переход от традиционных хранилищ данных к озерам данных дает возможность хранить данные разных типов и управлять ими: обрабатывать структурированные, полуструктурированные и неструктурированные данные из различных источников, обеспечивая централизованное хранилище для всех геопространственных данных. Использование озера данных на основе ГИС позволяет обогатить анализ и визуализацию географических данных, а также улучшить управление и использование пространственной информацией в целом.

Рассмотренный подход к организации перехода от хранилищ данных к озеру данных в геоинформационных системах, основанных на Лямбда-архитектуре, обладает следующими ***преимуществами***:

- гибкость в хранении. Озера данных могут хранить неструктурированные данные, такие как изображения, видео, аудиоданные, а также структурированные и полуструктурированные данные. Это идеально подходит для ГИС, где данные часто приходят в различных форматах;
- масштабируемость. Озера данных хорошо масштабируются для хранения огромных объемов данных, что крайне важно для ГИС, работающих с большими наборами пространственных данных;
- затраты. Использование технологий хранения вроде Hadoop для озер данных часто бывает более экономичным, поскольку они спроектированы для работы на стандартном

```python
import geopandas as gpd
from kafka import KafkaConsumer
from pyspark.sql import SparkSession
from pyspark.streaming import StreamingContext
from pyspark.streaming.kafka import KafkaUtils
# Подготовка Spark сессии
spark = SparkSession.builder.appName("GISLambdaArchitecture").getOrCreate()
sc = spark.sparkContext
ssc = StreamingContext(sc, 1)
# Пакетная обработка исторических геоданных (Batch Layer)
def batch_process():
    # Загрузка исторических данных ГИС
    historial_gis_data = gpd.read_file("historical_gis_data.geojson")
    # Выполнить преобразования / агрегацию данных
    # Это могут быть операции вроде преобразования координат, фильтрации или суммирования данных.
    processed_data = historial_gis_data  # Здесь должна быть логика обработки
    # Сохранение обработанных данных для дальнейшего анализа
    processed_data.to_file("processed_historical_gis_data.geojson")
# Потоковая обработка реальных данных (Speed Layer)
def stream_process():
    # Функция для обработки каждого сообщения
    def process_msg(msg):
        # Преобразование сообщений в геопространственный формат
        gis_content = gpd.GeoDataFrame.from_features(msg)
        # Здесь может быть логика обработки
        # Потенциально здесь можно обновить слой Serving с новыми данными
    # Консьюмер Kafka для потоковых данных ГИС
    kafka_consumer = KafkaConsumer('gis_messages', bootstrap_servers='localhost:9092')
    # Создаем Kafka DStream
    kafka_stream = KafkaUtils.createDirectStream(ssc, ['gis_messages'], {"metadata.broker.list": 'localhost:9092'})
    # Применение функции обработки к каждому элементу DStream
    kafka_stream.foreachRDD(lambda rdd: rdd.foreachPartition(process_msg))
# Запускаем оба процесса
batch_process()
stream_process()
# Запускаем потоковый контекст
ssc.start()
ssc.awaitTermination()
```

Рис. 3. Пример кода обработки данных, основанной на Лямбда-архитектуре

оборудовании, в отличие от более дорогостоящих решений для хранилищ данных;

• интеграция данных в реальном времени. Озера данных в сочетании с Лямбда-архитектурой позволяют ГИС интегрировать потоки данных в реальном времени с большими пакетами исторических данных для более динамичного и комплексного анализа;

• улучшенный анализ. Озера данных позволяют использовать передовые технологии обработки, включая машинное обучение и искусственный интеллект, для обработки и анализа данных в ГИС.

Организация перехода от хранилищ данных к озерам данных в геоинформационных системах на основе Лямбда-архитектуры позволяет эффективно обрабатывать и анализировать большие объемы геопространственных данных. Применение современных технологий обработки данных и интеграции данных открывает новые возможности для управления информацией и повышения качества аналитики в ГИС. Дальнейшие исследования в этой области могут способствовать развитию интеллектуальных систем обработки геоданных.

Литература

1. Ёсу, М. Т. Принципы организации распределенных баз данных = Principles of Distributed Database Systems. Fourth Edition / М. Т. Ёсу, П. Вальдуриес; пер. с англ. А. А. Слинкина. — Москва: ДМК Пресс, 2021. — 672 с.

2. Bhattacherjee, S. RStore: A Distributed Multi-Version Document Store / S. Bhattacherjee, A. Deshpande // Proceedings of the 34th International Conference on Data Engineering (ICDE 2018), (Paris, France, 16–19 April 2018). — Institute of Electrical and Electronics Engineers, 2018. — Pp. 389–400. DOI: 10.1109/ICDE.2018.00043.

3. Leveraging the Data Lake: Current State and Challenges / C. Giebler, C. Gröger, E. Hoos, [et al.] // Big Data Analytics and Knowledge Discovery (DaWaK 2019): Proceedings of the 21st International Conference (Linz, Austria, 26–29 August 2019) / C. Ordonez, [et al.] (eds.). — Cham: Springer Nature, 2019. — Pp. 179–188. — (Lecture Notes in Computer Science. Vol. 11708). DOI: 10.1007/978–3–030–27520–4_13.

4. Lock, M. Maximizing Your Data Lake with a Cloud or Hybrid Approach / M. Lock; Aberdeen Group. — 2016. — 4 p. URL: http://technology-signals.com/wp-content/uploads/download-manager-files/maximizingyourdatalake.pdf (дата обращения 12.01.2024).

5. Extending Data Lake Metadata Management by Semantic Profiling / J.W. Ansari, N. Karim, S. Decker, [et al.] // Proceedings of the 15th International Extended Semantic Web Conference (ESWC 2018), (Heraklion, Crete, Greece 03–07 June 2018). — Springer International Publishing, 2018. — 15 p. URL: http://2018.eswc-conferences.org/wp-content/uploads/2018/02/ESWC2018_paper_127.pdf (дата обращения 12.01.2024)

6. CoreDB: A Data Lake Service / A. Beheshti, B. Benatallah, R. Nouri, [et al.] // Proceedings of the 2017 ACM Conference on

Information and Knowledge Management (CIKM '17), (Singapore, Singapore, 06–10 November 2017). — New York: Association for Computing Machinery, 2017. — Pp. 2451–2454. DOI: 10.1145/3132847.3133171.

7. Data Lake Management: Challenges and Opportunities / F. Nargesian, E. Zhu, R.J. Miller, [et al.] // Proceedings of the VLDB Endowment. 2019. Vol. 12, Is. 12. Pp. 1986–1989. DOI: 10.14778/3352063.3352116.

8. CLAMS: Bringing Quality to Data Lakes / M. Farid, A. Roatis, I.F. Ilyas, [et al.] // Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16), (San Francisco, CA, USA, 26 June-01 July 2016). — New York: Association for Computing Machinery, 2016. — Pp. 2089–2092. DOI: 10.1145/2882903.2899391.

9. Keeping the Data Lake in Form: DS-kNN Datasets Categorization Using Proximity Mining / A. Alserafi, A. Abelló, O. Romero, T. Calders // Model and Data Engineering (MEDI 2019): Proceedings of the 9th International Conference (Toulouse, France, 28–31 October 2019) / K.-D. Schewe, N.K. Singh (eds.). — Cham: Springer Nature, 2019. — Pp. 35–49. — (Lecture Notes in Computer Science. Volume 11815). DOI: 10.1007/978–3–030–32065–2_3.

10. Dataset Discovery in Data Lakes / A. Bogatu, A. A.A. Fernandes, N.W. Paton, N. Konstantinou // Proceedings of the IEEE 36th International Conference on Data Engineering (ICDE 2020), (Dallas, TX, USA, 20–24 April 2020). — Institute of Electrical and Electronics Engineers, 2020. — Pp. 709–720. DOI: 10.1109/ICDE48307.2020.00067.

11. Goods: Organizing Google's Datasets / A. Halevy, F. Korn, N.F. Noy // Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16), (San Francisco, CA, USA, 26 June-01 July 2016). — New York: Association for Computing Machinery, 2016. — Pp. 795–806. DOI: 10.1145/2882903.2903730.

12. Sawadogo, P.N. On Data Lake Architectures and Metadata Management / P.N. Sawadogo, J. Darmont // Journal of Intelligent Information Systems. 2021. Vol. 56, Is. 1. Pp. 97–120. DOI: 10.1007/s10844–020–00608–7.

13. Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics / M. Armbrust, A. Ghodsi, R. Xin, M. Zaharia // Proceedings of the 11th Annual Conference on Innovative Data Systems Research (CIDR 21), (11–15 January 2021, Online). — 8 p. URL: http://cidrdb.org/cidr2021/papers/cidr2021_paper17.pdf (дата обращения 12.01.2024).

14. Jensen, R., Shen, H., & Yue, P. (2017). Geo-Analytics: Integrating Geospatial Information Systems and Big Data Analytics. In Geographic Information Science (pp. 297–315). Springer International Publishing.

15. Gao, S., & Liu, Z. (2021). A Review of Big Data and Geospatial Data Integration for Geocomputation and Decision Support. Remote Sensing, 13(2), 316. https://doi.org/10.3390/rs13020316.

16. International Network Performance and Security Testing Based on Distributed Abyss Storage Cluster and Draft of Data Lake Framework / B.-R. Cha, S. Park, J.-W. Kim // Security and Communication Networks. 2018. Art. No. 1746809. 14 p. DOI: 10.1155/2018/1746809.

17. Rituerto, Á., & Alvarez, J. M. (2019). Geo-Big Data: A Literature Review. ISPRS International Journal of Geo-Information, 8(11), 471. https://doi.org/10.3390/ijgi8110471.

18. Безворотных, А. В. Lambda architecture для корпоративного «Озера данных» / А. В. Безворотных; науч. рук. Р. И. Кузьмич // Молодость. Интеллект. Инициатива: Материалы X Международной научно-практической конференции студентов и магистрантов (Витебск, Беларусь, 22 апреля 2022 г.). — Витебск: Витебский гос. ун-т имени П.М. Машерова, 2022. — С. 6–8.

19. Implementing Big Data Lake for Heterogeneous Data Sources / H. Mehmood, E. Gilman, M. Cortes, [et al.] // Proceedings of the IEEE 35th International Conference on Data Engineering Workshops (ICDEW 2019), (Macao, China, 08–12 April 2019). — Institute of Electrical and Electronics Engineers, 2020. — Pp. 37–44. DOI: 10.1109/ICDEW.2019.00–37.

20. Marz, N. Big Data: Principles and best practices of scalable realtime data systems / N. Marz, J. Warren. — Shelter Island (NY): Manning Publications, 2015. — 328 p.

21. Sawadogo, P.N. Metadata Management for Textual Documents in Data Lakes / P.N. Sawadogo, T. Kibata, J. Darmont // Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019), (Heraklion, Crete, Greece, 03–05 May 2019). — SciTePress, 2019. — Vol. 1. — Pp. 72–83. DOI: 10.5220/0007706300720083.

22. Visual Bayesian Fusion to Navigate a Data Lake / K. Singh, K. Paneri, A. Pandey, [et al.] // Proceedings of the 19th International Conference on Information Fusion (FUSION 2016), (Heidelberg, Germany, 05–08 July 2016). — Institute of Electrical and Electronics Engineers, 2016. — Pp. 987–994.

23. Munshi, A. A. Data Lake Lambda Architecture for Smart Grids Big Data Analytics / A.A. Munshi, Y. A.-R. I. Mohamed // IEEE Access. 2018. Vol. 6. Pp. 40463–40471. DOI: 10.1109/ACCESS.2018.2858256.

24. DataHub — A Metadata Platform for the Modern Data Stack. URL: http://datahubproject.io (дата обращения 25.12.2023).

Информация об авторах

Абу Хасан Рахеб — магистр, аспирант кафедры «Информационные и вычислительные системы», Петербургский государственный университет путей сообщения Императора Александра I. Область научных интересов: информационные системы, обработка больших данных, моделирование надежности; ragheb1997@yandex.ru.

А. Б. Кириенко — адъюнкт кафедры «Математическое и программное обеспечение» ВКА им. А. Ф. Можайского. Область научных интересов: информационные системы, обработка больших данных, вероятностное моделирование геоинформационных систем, генетические алгоритмы, информационная безопасность; vka_kaf27_1@mil.ru.

А. Д. Хомоненко — докт. техн. наук, профессор; профессор кафедры «Информационные и вычислительные системы», Петербургский государственный университет путей сообщения Императора Александра I; профессор кафедры «Математическое и программное обеспечение», ВКА им. А. Ф. Можайского. Область научных интересов: информационные системы, базы данных, обработка больших данных, вероятностное моделирование информационных систем, информационная безопасность. E-mail: khomon@mail.ru. Адрес: 190031, Санкт-Петербург, Московский пр., 9.

Статья поступила в редакцию 29.01.2024; одобрена после рецензирования 17.03.2024.

УДК 004.67
DOI: 10.20295/2413-2527-2024-137-45-55

Method of Transition from Data Warehouses to Geographic Information System Data Lakes Based on Lambda Architecture

Master **R. Abu Khasan**
Emperor Alexander I St. Petersburg State Transport University
Saint Petersburg, Russia

A. B. Kirienko
A. F. Mozhaysky's Military Space Academy
Saint Petersburg, Russia

Gr. PhD **A. D. Khomonenko**
Emperor Alexander I St. Petersburg State Transport University
A. F. Mozhaysky's Military Space Academy
Saint Petersburg, Russia

Abstract. **This paper discusses the transition from traditional data warehouses to data lakes in geographic information systems using Lambda architecture. Provides an overview of the key transition steps, including planning, data collection and processing, data querying, data analytics, and metadata management. Particular attention is paid to the interaction of data lakes and GIS, as well as sample big data processing code based on Lambda architecture. The advantages of using data lakes in GIS and the possibilities of integrating modern data processing technologies are considered.**

Keywords: **data lakes; data warehouses; Lambda architecture; geographic information systems; metadata; big data processing; data integration; data analytics; transition from data warehouses.**

For citation: **Abou Hasan R., Kirienko A. B., Khomonenko A. D. Method of Transition from Data Warehouses to Geographic Information System Data Lakes Based on Lambda Architecture // Intellectual Technologies on Transport. 2024. No 1 (37). P. 45–55. (In Russ.). DOI: 10.20295/2413-2527-2024-137-45-55**

References

1. Özsu M. T., Valduriez P. Principles of Distributed Database Systems Fourth Edition [Printsipy organizatsii raspredelennykh baz dannykh]. Moscow, DMK Press, 2021, 672 p.

2. Bhattacherjee S., Deshpande A. RStore: A Distributed Multi-Version Document Store, *Proceedings of the 34th International Conference on Data Engineering (ICDE 2018), Paris, France, April 16–19, 2018*. Institute of Electrical and Electronics Engineers, 2018, Pp. 389–400. DOI: 10.1109/ICDE.2018.00043.

3. Giebler C., Gröger C., Hoos E., et al. Leveraging the Data Lake: Current State and Challenges. In: *Ordonez C., et al. (eds.). Big Data Analytics and Knowledge Discovery (DaWaK 2019): Proceedings of the 21st International Conference, Linz, Austria, 26–29 August 26–29, 2019. Lecture Notes in Computer Science, Vol. 11708*. Cham, Springer Nature, 2019, Pp. 179–188. DOI: 10.1007/978–3–030–27520–4_13.

4. Lock M. Maximizing Your Data Lake with a Cloud or Hybrid Approach. May 2016, 4 p. Available at: http://technology-signals.com/wp-content/uploads/download-manager-files/maximizingyourdatalake.pdf (accessed 12 Jan 2024).

5. Beheshti A., Benatallah B., Nouri R., et al. CoreDB: A Data Lake Service, *Proceedings of the 2017 ACM Conference on Information and Knowledge Management (CIKM '17), Singapore, Singapore, November 06–10, 2017*. New York, Association for Computing Machinery, 2017, Pp. 2451–2454. DOI: 10.1145/3132847.3133171.

6. Nargesian F., Zhu E., Miller R. J., et al. Data Lake Management: Challenges and Opportunities, *Proceedings of the VLDB Endowment*, 2019, Vol. 12, Is. 12, Pp. 1986–1989. DOI: 10.14778/3352063.3352116.

7. Farid M., Roatis A., Ilyas I. F., et al. CLAMS: Bringing Quality to Data Lakes, *Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16), San Francisco, CA, USA, June 26–July 01, 2016*. New York, Association for Computing Machinery, 2016, Pp. 2089–2092. DOI: 10.1145/2882903.2899391.

8. Alserafi A., Abelló A., Romero O., Calders T. Keeping the Data Lake in Form: DS-kNN Datasets Categorization Using Proximity Mining. In: *Schewe K.-D., Singh N.K. (eds.) Model and Data Engineering (MEDI 2019): Proceedings of the 9th International Conference, Toulouse, France, October 28–31, 2019. Lecture Notes in Computer Science, Vol. 11815*. Cham, Springer Nature, 2019, Pp. 35–49. DOI: 10.1007/978–3–030–32065–2_3.

9. Bogatu A., Fernandes A. A. A., Paton N. W., Konstantinou N. Dataset Discovery in Data Lakes, *Proceedings of the IEEE 36th International Conference on Data Engineering (ICDE 2020), Dallas, TX, USA, April 20–24, 2020*. Institute of Electrical and Electronics Engineers, 2020, Pp. 709–720. DOI: 10.1109/ICDE48307.2020.00067.

10. Halevy A., Korn F., Noy N. F. Goods: Organizing Google's Datasets, *Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16), San Francisco, CA, USA, June 26–July 01, 2016*. New York, Association for Computing Machinery, 2016, Pp. 795–806. DOI: 10.1145/2882903.2903730.

11. Sawadogo P. N., Darmont J. On Data Lake Architectures and Metadata Management, J*ournal of Intelligent Information Systems*, 2021, Vol. 56, Is. 1, Pp. 97–120. DOI: 10.1007/s10844–020–00608–7.

12. Armbrust M., Ghodsi A., Xin R., Zaharia M. Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics, *Proceedings of the 11th Annual Conference on Innovative Data Systems Research (CIDR 21), January 11–15, 2021, Online*. 8 p. Available at: http://cidrdb.org/cidr2021/papers/cidr2021_paper17.pdf (accessed 12 Jan 2024).

13. Cha B.-R., Park S., Kim J.-W. International Network Performance and Security Testing Based on Distributed Abyss Storage Cluster and Draft of Data Lake Framework, *Security and Communication Networks*, 2018, Art. No. 1746809, 14 p. DOI: 10.1155/2018/1746809.

14. Bezvorotnykh, A. V. Lambda Architecture for the Corporate «Data Lake» **[Lambda architecture** dlya korporativnogo «Ozera dannykh»**],** *Youth. Intelligence. Initiative: Materials of the X International Scientific and Practical Conference of Students and Undergraduates [Molodost. Intellekt. Initsiativa: Materialy X Mezhdunarodnoy nauchno-prakticheskoy konferentsii studentov i magistrantov], Vitebsk, Belarus, April 22, 2022*. Vitebsk, Vitebsk State University named after P.M. Masherov, 2022, Pp. 6–8.

15. Mehmood H., Gilman E., Cortes M., et al. Implementing Big Data Lake for Heterogeneous Data Sources, *Proceedings of the IEEE 35th International Conference on Data Engineering Workshops (ICDEW 2019), Macao, China, April 08–12, 2019*. Institute of Electrical and Electronics Engineers, 2020, Pp. 37–44. DOI: 10.1109/ICDEW.2019.00–37.

16. Marz N., Warren J. Big Data: Principles and best practices of scalable realtime data systems. Shelter Island (NY), Manning Publications, 2015, 328 p.

17. Sawadogo P.N., Kibata T., Darmont J. Metadata Management for Textual Documents in Data Lakes, *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019), Heraklion, Crete, Greece, May 03–05, 2019. Volume 1*. SciTePress, 2019, Pp. 72–83. DOI: 10.5220/0007706300720083.

18. Singh K., Paneri K., Pandey A., et al. Visual Bayesian Fusion to Navigate a Data Lake, *Proceedings of the 19th International Conference on Information Fusion (FUSION 2016), Heidelberg, Germany, July 05–08, 2016*. Institute of Electrical and Electronics Engineers, 2016, Pp. 987–994.

19. Munshi A. A., Mohamed Y.A.-R. I. Data Lake Lambda Architecture for Smart Grids Big Data Analytics, *IEEE Access*, 2018, Vol. 6, Pp. 40463–40471. DOI: 10.1109/ACCESS.2018.2858256.

20. DataHub — A Metadata Platform for the Modern Data Stack. Available at: http://datahubproject.io (accessed 25 Dec 2023).

Information about Authors

Abu Khasan Raheb — Master. Postgraduate student of the Department of Information and Computing Systems in Emperor Alexander I St. Petersburg State Transport University. Research interests: information systems, big data processing, reliability modeling; ragheb1997@yandex.ru

Andrey Borisovich Kiriyenko — Associate Professor of the Department of Mathematics and Software VKA named after A. F. Mozhaisky. Research interests: information systems, big data processing, probabilistic modeling of geoinformation systems, genetic algorithms, information security; vka_kaf27_1@mil.ru.

Anatoly Dmitrievich Khomonenko — Doctor of Technical Sciences, Professor. Professor of the Department of Information and Computing Systems in Emperor Alexander I St. Petersburg State Transport University. Professor of the Department of Mathematics and Software in VKA named after A. F. Mozhaisky. Research interests: information systems, databases, big data processing, probabilistic modeling of information systems, information security; khomon@mail.ru.

The article was submitted 29.01.2024; approved after reviewing 17.03.2024.