DOI 10.37882/2223-2966.2026.02-2.06

АНАЛИЗ ПРОБЛЕМ И ПОДХОДОВ К ОБРАБОТКЕ ИНФОРМАЦИИ ПРИ МИГРАЦИИ БОЛЬШИХ ДАННЫХ

ANALYSIS OF PROBLEMS AND APPROACHES TO INFORMATION PROCESSING DURING BIG DATA MIGRATION

A. Badanov

Summary. This article examines information processing challenges arising during big data migration in the context of heterogeneous and multivariate data warehouses. A one-time data migration scenario, typical for infrastructure modernization, technology platform transitions, and data source consolidation, is examined as a methodological constraint. It is shown that transferring data and associated metadata between systems with different storage models significantly complicates information processing. This paper identifies and analyzes five key issues that have the greatest impact on the correctness and efficiency of big data migration: differences in supported data types and storage models, limited portability of data processing methods, failures and errors when processing large volumes of information, the specifics of data extraction from non-standard sources, and the limited computing resources of the source systems. For each of these challenges, approaches and mitigation methods are considered, based on adapting data processing processes to architectural and resource constraints. The research results can be used in the design and implementation of big data migration processes in modern information systems and also serve as a basis for developing adaptive and hybrid approaches to information processing.

Keywords: big data, big data migration, information processing, data warehouses, ETL, storage models, computing resources, big data migration challenges.

Баданов Артем Андреевич
ведущий разработчик, ООО «Амбердата»
artem_badanov@inbox.ru

Аннотация. В статье рассматриваются проблемы обработки информации, возникающие при миграции больших данных в условиях использования разнородных и многовариационных хранилищ данных. В качестве методологических ограничений исследуется сценарий единоразовой миграции данных, характерный для задач модернизации инфраструктуры, смены технологических платформ и консолидации источников информации. Показано, что перенос данных и сопутствующих метаданных между системами с различными моделями хранения существенно усложняет процессы обработки информации.

В работе выделены и проанализированы пять ключевых проблем, оказывающих наибольшее влияние на корректность и эффективность миграции больших данных: различия в поддерживаемых типах данных и моделях хранения, ограниченная переносимость методов обработки данных, сбои и ошибки при обработке больших объемов информации, особенности извлечения данных из нестандартных источников, а также ограниченность вычислительных ресурсов исходных систем. Для каждой из выделенных проблем рассмотрены подходы и методы их смягчения, основанные на адаптации процессов обработки данных с учетом архитектурных и ресурсных ограничений.

Результаты исследования могут быть использованы при проектировании и реализации процессов миграции больших данных в современных информационных системах, а также служат основой для разработки адаптивных и гибридных подходов к обработке информации.

Ключевые слова: большие данные, миграция больших данных, обработка информации, хранилища данных, ETL, модели хранения, вычислительные ресурсы, проблемы миграции больших данных.

В условиях цифровизации и роста объемов информации процессы миграции данных приобретают особую значимость. Современные информационные системы постоянно эволюционируют: происходит переход от монолитных архитектур к распределенным, от локальных хранилищ к облачным и гибридным средам, от традиционных реляционных баз данных к MPP— и NoSQL-решениям. В рамках таких преобразований неизбежно возникает необходимость переноса данных между различными системами хранения и обработки.

Миграция данных представляет собой сложный многоэтапный процесс, включающий извлечение, преобразование, передачу и загрузку информации в целевое хранилище. При этом ключевую роль играет именно обработка информации, так как данные зачастую различаются по структуре, формату, семантике и требованиям к качеству. Ошибки, допущенные на этапе обработки, могут приводить к потере данных, снижению их достоверности, нарушению целостности и, как следствие, к некорректной работе прикладных систем.

Целью данной работы является анализ основных проблем, возникающих при обработке информации в процессе миграции данных, а также обзор существующих и перспективных подходов к их решению. В статье рассматриваются как технологические, так и методологические аспекты миграции, что позволяет комплексно оценить эффективность применяемых решений.

Процесс миграции данных, как правило, представляет собой многоэтапную процедуру, направленную на перенос информации между разнородными информационными системами с сохранением ее целостности, корректности и семантического смысла. Каждый этап миграции характеризуется собственным набором задач и потенциальных рисков.

1. Анализ источников и системы-приемника данных

На начальном этапе выполняется детальное исследование исходных систем и целевого хранилища данных. В рамках данного этапа анализируются логическая и физическая структуры данных, используемые форматы хранения, объемы информации, а также существующие ограничения, связанные с типами данных, ключами, индексами и бизнес-правилами. Особое внимание уделяется выявлению различий между схемами источника и приемника, включая несоответствия в именовании атрибутов, типах данных и правилах их интерпретации. Результаты данного анализа служат основой для построения стратегии обработки и преобразования информации на последующих этапах.

2. Извлечение данных

Этап извлечения данных предполагает получение информации из исходных систем без нарушения их работоспособности и целостности. В зависимости от архитектуры источников данные могут извлекаться пакетным или потоковым способом, а также в полном или инкрементальном режиме. На этом этапе возникают проблемы, связанные с доступностью источников, ограничениями по времени выполнения операций и возможными изменениями данных в процессе миграции. Кроме того, извлеченные данные часто содержат избыточную или нерелевантную информацию, что требует их последующей обработки.

3. Обработка и преобразование данных

Обработка и преобразование данных является центральным и наиболее критичным этапом миграции. Именно на данном этапе осуществляется приведение данных к требованиям целевой системы хранения. Процедуры обработки включают очистку данных от ошибок и дубликатов, заполнение пропущенных значений, нормализацию форматов, агрегацию и декомпозицию данных, а также применение бизнес-правил преобразования. Дополнительно может выполняться сопоставление атрибутов и семантическое выравнивание данных между источником и приемником. Ошибки или неточности на данном этапе приводят к искажению информации и могут негативно сказаться на дальнейшей аналитической и операционной деятельности, что делает данный этап ключевым с точки зрения качества миграции.

4. Загрузка данных в систему-приемник

После завершения обработки данные загружаются в целевое хранилище. Данный этап требует учета особенностей архитектуры системы-приемника, включая требования к производительности, ограничения на объемы загрузки и механизмы обеспечения транзакционной целостности. При загрузке могут использоваться различные стратегии, такие как пакетная загрузка, параллельная обработка или поэтапное наполнение хранилища. Некорректная организация данного этапа может привести к частичной загрузке данных, нарушению связей между сущностями или снижению производительности системы.

5. Верификация и контроль качества данных

Заключительным этапом является верификация результатов миграции и контроль качества данных. На данном этапе проводится проверка полноты и корректности перенесенной информации, сопоставление объемов данных до и после миграции, а также контроль логической и ссылочной целостности. Дополнительно могут выполняться выборочные проверки бизнес-показателей и тестирование прикладных сценариев работы с данными. Верификация позволяет выявить ошибки, допущенные на предыдущих этапах, и при необходимости скорректировать процедуры обработки данных.

При анализе проблем обработки информации в процессе миграции данных необходимо ввести ряд методологических ограничений, позволяющих сузить предмет исследования и обеспечить корректность получаемых выводов. В рамках данной статьи рассматривается сценарий единоразовой (one-time) миграции данных, осуществляемой при переходе между информационными системами или хранилищами данных нового поколения. Такой подход характерен для задач модернизации инфраструктуры, смены технологической платформы или консолидации разнородных источников данных.

Дополнительно предполагается, что в процессе миграции используются многовариационные хранилища данных, поддерживающие различные модели представления информации, включая реляционные, колоночные, документоориентированные и аналитические структуры. В рамках рассматриваемого процесса осуществляется перенос не только самих данных, но и сопутствующих метаданных, таких как схемы данных, справочники, словари значений, описания связей и бизнес-правила обработки информации. Это существенно усложняет задачи обработки данных и расширяет спектр возникающих проблем.

Следует отметить, что использование нескольких разнородных хранилищ данных порождает широкий круг проблем, связанных с различиями в архитектуре,

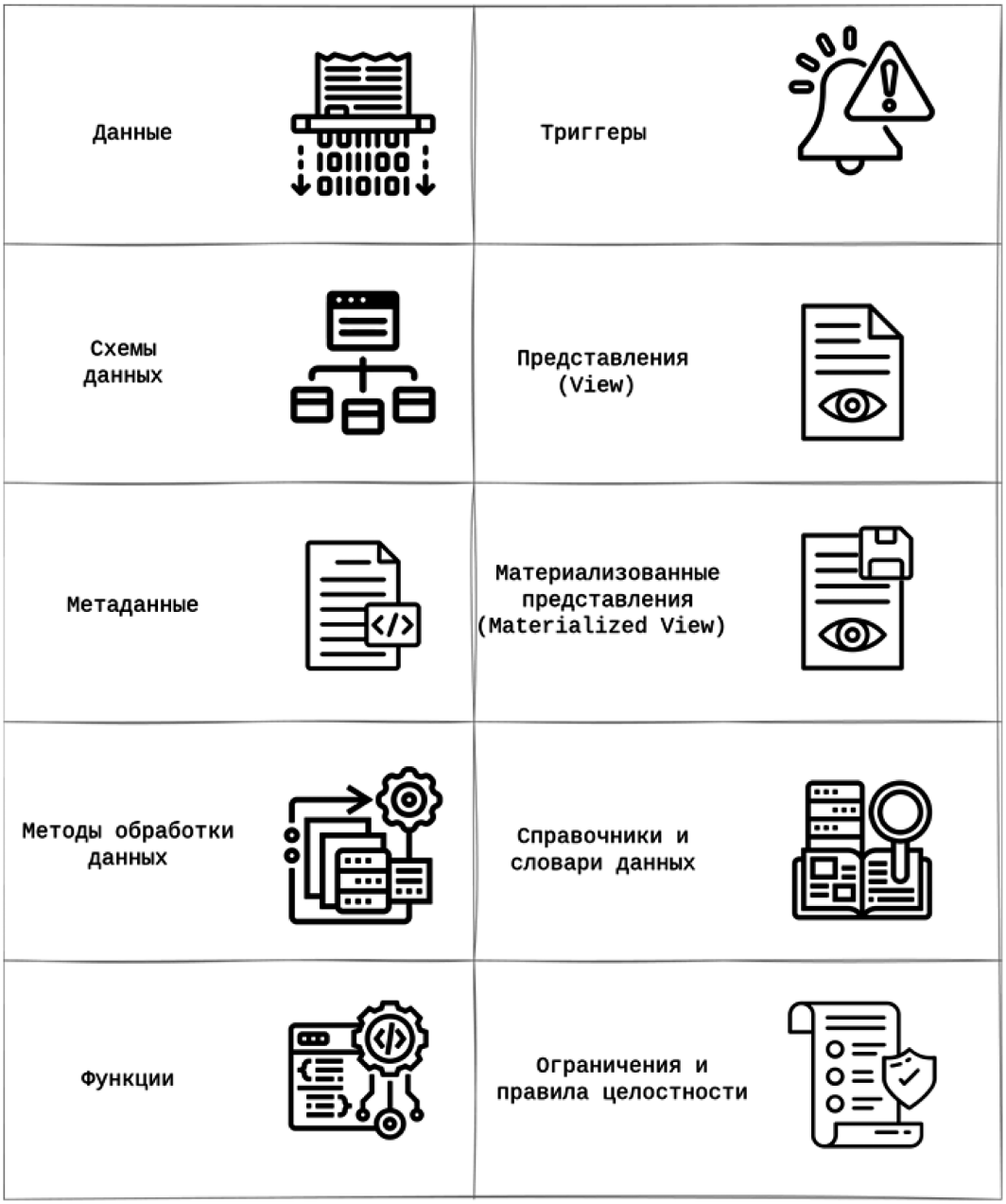

Рис 1. Переносимые элементы в рамках миграции больших данных

поддерживаемых типах данных, методах обработки и ограничениях на использование ресурсов. Полный анализ всех возможных проблем в рамках одной статьи представляется затруднительным. В связи с этим в настоящем исследовании рассматриваются основные и наиболее характерные проблемы обработки информации, оказывающие наибольшее влияние на корректность и эффективность миграции данных в условиях использования многовариационных хранилищ.

Проблема 1. Различия в поддерживаемых типах данных и моделях хранения

Одной из существенных проблем обработки информации при миграции является наличие различных типов данных и моделей их представления в системах-источниках и системах-приемниках. Хранилища данных могут поддерживать различные наборы типов, включая примитивные, составные, вложенные и специализированные типы, что приводит к необходимости дополнительного преобразования данных. Методы обработки, корректно применимые к данным в одном хранилище, могут оказаться неприменимыми или неэффективными в другом вследствие различий в форматах хранения, механизмах сериализации и правилах интерпретации данных.

В результате возникает необходимость адаптации или переопределения процедур обработки информа-

ции с учетом особенностей целевой системы, что увеличивает сложность миграции и повышает риск возникновения ошибок на этапе преобразования данных.

Проблема 2. Ограниченная переносимость методов обработки данных

Методы обработки данных, применяемые в процессе миграции, как правило, тесно связаны не только с вычислительной платформой, но и с внутренними механизмами конкретного хранилища данных. Даже при использовании формально схожих операций, таких как фильтрация, агрегация или работа с текстовыми данными, реализация и поведение соответствующих методов могут существенно различаться в разных системах хранения.

Например, методы обработки текстовых данных, реализованные в реляционных системах управления базами данных, могут опираться на иные принципы индексирования, кодирования и сопоставления строк по сравнению с аналитическими колоночными хранилищами. В результате операции, корректно и эффективно выполняемые в одном хранилище, могут демонстрировать иное поведение, снижение производительности либо вовсе отсутствовать в другом.

Данная проблема усугубляется при использовании многовариационных хранилищ, где в рамках одного процесса миграции задействуются несколько систем с различными наборами поддерживаемых функций и методов обработки. Чем больше количество таких хранилищ, тем выше вероятность того, что отдельные методы обработки данных будут работать по-разному либо не будут поддерживаться целевой системой вовсе. Это приводит к необходимости переопределения логики обработки данных, разработки альтернативных алгоритмов или применения дополнительных этапов преобразования информации.

Проблема 3. Сбои и ошибки, связанные с обработкой больших объемов данных

Работа с большими объемами данных и сложными преобразованиями существенно повышает вероятность возникновения ошибок, связанных с нехваткой вычислительных ресурсов, превышением лимитов памяти, временными задержками и обрывами соединений с источниками или приемниками данных. Такие сбои могут происходить как на этапе извлечения, так и в процессе обработки информации.

В условиях единоразовой миграции повторное выполнение всех этапов обработки данных является крайне нежелательным, так как оно требует значительных временных и вычислительных затрат. В связи с этим возникает необходимость использования промежуточных хранилищ, позволяющих сохранять результаты частично выполненной обработки и продолжать миграцию с точки прерывания, а не начинать процесс заново.

Проблема 4. Особенности извлечения данных из нестандартных источников

Дополнительные сложности возникают в тех случаях, когда источником данных выступают системы, не предназначенные для предоставления данных через формализованные интерфейсы, такие как API. Примером таких источников могут служить корпоративные порталы, внутренние информационные системы или пользовательские приложения, которые одновременно подвергаются модернизации или миграции.

В подобных условиях процесс извлечения данных может быть нестабильным и подверженным изменениям структуры и форматов информации. Это приводит к необходимости дополнительной обработки данных, включая адаптацию методов извлечения и обеспечение устойчивости к изменениям источника в процессе миграции.

Проблема 5. Ограниченность вычислительных ресурсов при миграции больших данных

При миграции больших объемов данных существенной проблемой обработки информации является ограниченность вычислительных и системных ресурсов исходных хранилищ. В большинстве практических сценариев системы-источники продолжают использоваться в штатном режиме другими сотрудниками и прикладными сервисами, что накладывает жесткие ограничения на допустимую нагрузку, создаваемую процессами миграции.

В условиях совместного использования ресурсов выделение достаточных вычислительных мощностей для выполнения операций извлечения и обработки данных оказывается затруднительным. Интенсивные запросы, сложные преобразования и длительные операции чтения могут негативно сказаться на производительности рабочих процессов, что ограничивает возможности применения ресурсоемких методов обработки информации в рамках миграции [6, 7]. В результате процедуры миграции вынужденно адаптируются к доступному объему ресурсов, что может приводить к увеличению времени обработки данных и усложнению логики преобразований [9,10].

Дополнительные сложности возникают при миграции больших данных, требующих параллельной обработки и значительных объемов оперативной памяти. Недостаток ресурсов может приводить к прерыванию операций, увеличению времени отклика систем и воз-

никновению ошибок выполнения. В таких условиях особое значение приобретает планирование использования ресурсов и выбор методов обработки информации, минимизирующих нагрузку на системы-источники [6, 14].

В настоящее время в научной и прикладной литературе представлено значительное количество подходов и решений, направленных на организацию процессов миграции больших данных. Наиболее распространенными среди них являются концепции ETL (Extract, Transform, Load) и ELT (Extract, Load, Transform), широко применяемые при переносе данных между различными информационными системами и хранилищами данных.

ETL-подход предполагает предварительное извлечение данных из источников с последующим выполнением операций обработки и преобразования, после чего данные загружаются в систему-приемник. Такой подход традиционно используется при построении хранилищ данных и ориентирован на обеспечение высокого качества информации за счет выполнения преобразований до момента загрузки. В свою очередь, ELT-подход переносит акцент на использование вычислительных возможностей целевого хранилища, где преобразование данных осуществляется уже после их загрузки. Данный подход особенно актуален в условиях современных аналитических платформ, обладающих высокой производительностью и масштабируемостью [6–8, 14].

Несмотря на широкое распространение указанных подходов, большинство существующих решений носят достаточно абстрактный характер и ориентированы преимущественно на описание общих этапов миграции данных. Как правило, такие решения не учитывают в явном виде проблемы, возникающие при обработке информации в условиях использования разнородных хранилищ, различий в поддерживаемых типах данных, ограниченной переносимости методов обработки, а также рисков, связанных с ресурсными ограничениями и сбоями в процессе миграции больших объемов данных [9–10].

В результате практическая реализация миграции данных в сложных инфраструктурах требует значительной адаптации существующих подходов и разработки дополнительных механизмов обработки информации. Особенно это актуально для сценариев единоразовой миграции, где отсутствует возможность многократной итеративной корректировки процессов, а ошибки обработки данных могут привести к существенным затратам времени и ресурсов.

В связи с этим в рамках данной статьи предлагаются одни из возможных решений, направленные на устранение и минимизацию проблем обработки информации, выявленных на предыдущих этапах исследования. Рассматриваемые решения ориентированы на повышение устойчивости процессов миграции, обеспечение корректности обработки данных и снижение рисков, связанных с использованием многовариационных хранилищ и большими объемами обрабатываемой информации [1, 2, 11].

Дальнейшее изложение будет посвящено детальному рассмотрению предлагаемых решений и их применению для устранения конкретных проблем обработки информации при миграции данных.

Проблема 1. Различия в поддерживаемых типах данных и моделях хранения

Для решения проблемы различий в поддерживаемых типах данных и моделях хранения при миграции больших данных необходимо на этапе планирования и выполнения миграции оперировать исключительно теми типами данных и структурами, которые поддерживаются как системой-источником, так и системой-приемником. В большинстве практических случаев базовые, примитивные типы данных совпадают между различными хранилищами, что позволяет осуществлять их перенос без существенных преобразований. Однако наибольшие сложности возникают при работе с непримитивными и составными структурами данных, такими как массивы, списки, вложенные структуры и пользовательские типы [1, 2, 5].

В подобных ситуациях требуется проведение анализа поддерживаемых типов данных в обоих хранилищах и поиск структурно и семантически близких аналогов. Преобразование таких данных часто требует дополнительной логики конвертации, включая изменение форматов хранения, разбиение сложных структур на более простые элементы или, напротив, агрегацию данных для соответствия требованиям целевой системы. Ошибки на данном этапе могут привести к потере части информации или нарушению ее интерпретации.

Наиболее существенные трудности связаны не столько с типами данных, сколько с различиями в моделях хранения данных. Даже при сравнении формально схожих объектов, таких как материализованные представления в различных хранилищах, различия в принципах их реализации могут быть принципиальными. Например, в аналитических колоночных хранилищах используются дополнительные механизмы агрегации и обработки состояний, которые не имеют прямых аналогов в классических реляционных системах. В таких случаях стандартные методы миграции оказываются неприменимыми, и требуется ручной анализ поведения соответствующих механизмов, а также разработка альтернативных способов хранения и обработки данных в целевой системе [6, 7, 14].

Использование специализированных механизмов обработки, таких как промежуточные состояния агрегации и их последующее объединение, дополнительно усложняет процесс миграции, поскольку требует глубокого понимания внутренней логики работы хранилища-источника. Поиск эквивалентных решений в системе-приемнике часто возможен лишь на уровне логики обработки данных, а не на уровне прямого переноса структур.

Таким образом, чем больше различий выявляется в поддерживаемых типах данных и моделях хранения, тем выше сложность и продолжительность процесса миграции больших данных. Учет данных факторов на этапе обработки информации является необходимым условием для обеспечения корректности и завершенности миграции [10, 15].

Проблема 2. Ограниченная переносимость методов обработки данных

Для решения проблемы ограниченной переносимости методов обработки данных при миграции необходимо на этапе подготовки и выполнения миграции оперировать явным перечнем методов и функций, поддерживаемых каждым из используемых хранилищ данных. В отличие от типов данных, методы обработки информации значительно сильнее зависят от внутренней реализации конкретной системы хранения и не обладают универсальной совместимостью.

На практике единственным возможным подходом к решению данной проблемы является детальный анализ официальной документации хранилищ данных и сопоставление используемых методов обработки. Такой анализ включает проверку наличия аналогичных функций, сопоставление входных параметров, типов возвращаемых значений, а также особенностей поведения методов в граничных и нестандартных ситуациях. Даже при совпадении наименований функций различия в их реализации могут приводить к отличиям в результатах обработки данных.

Дополнительные сложности возникают при использовании пользовательских функций (User-Defined Functions, UDF). В подобных случаях методы обработки данных могут быть реализованы вне стандартного набора функций хранилища и не иметь формализованного описания. Отсутствие или недостаточная полнота документации по UDF существенно затрудняет их перенос и адаптацию в рамках миграции данных.

Ситуация осложняется тем, что разработчик, выполняющий миграцию, не всегда является автором используемых UDF. В таких условиях анализ логики пользовательских функций требует изучения исходного кода, выявления зависимостей от конкретных методов хранилища и оценки возможности воспроизведения аналогичного поведения в целевой системе. При невозможности прямого переноса UDF возникает необходимость разработки альтернативных методов обработки данных, что увеличивает трудоемкость и продолжительность миграции.

Таким образом, ограниченная переносимость методов обработки данных обусловлена как различиями в стандартных функциях хранилищ, так и сложностями, связанными с использованием пользовательских функций. Учет данных факторов является обязательным при проектировании процедур обработки информации и выборе решений для миграции больших данных.

Проблема 3. Сбои и ошибки, связанные с обработкой больших объемов данных

При миграции больших объемов данных между архитектурно различающимися системами хранения возникает принципиальная проблема, заключающаяся в невозможности прямой загрузки данных из хранилища-источника в хранилище-приемник без использования дополнительного промежуточного слоя. В случае кардинальных различий архитектур, моделей хранения и механизмов обработки данных попытка прямого переноса информации приводит к высокой вероятности сбоев, потере данных либо некорректному результату миграции [9, 10, 15].

Такая ситуация характерна, например, при переносе данных из реляционной системы хранения в аналитическое колоночное хранилище. В системе-источнике данные могут обрабатываться с использованием транзакционных механизмов, сложных связей и пользовательских функций, тогда как система-приемник ориентирована на пакетную загрузку, агрегацию и аналитические операции. В подобных условиях выполнение прямых операций извлечения, преобразования и загрузки данных становится затруднительным или невозможным без предварительного приведения информации к универсальному формату.

В связи с этим возникает необходимость использования компромиссного промежуточного решения, обеспечивающего разделение этапов обработки данных и их последующей загрузки. Промежуточное хранилище выполняет роль нейтрального слоя, позволяющего зафиксировать результаты преобразований и обеспечить устойчивость миграционного процесса к сбоям и архитектурным ограничениям целевого хранилища.

В качестве промежуточного хранилища могут использоваться объектные системы хранения данных, такие как S3-совместимые хранилища, обладающие мини-

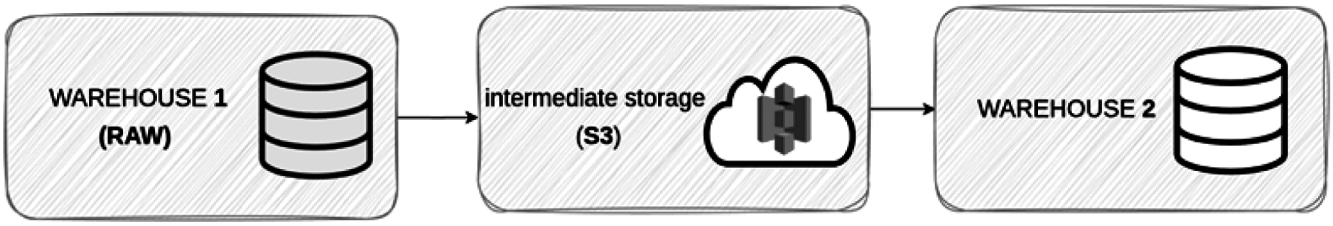

Рис. 2. Использование промежуточного хранилища S3

мальными ограничениями на типы и структуры данных. Подобные хранилища способны принимать и хранить данные в различных форматах без привязки к конкретной модели обработки или типу хранилища. Это делает их универсальным механизмом для аккумулирования преобразованных данных и последующей загрузки в систему-приемник.

Использование промежуточного хранилища позволяет:

— обеспечить корректное завершение этапов обработки данных независимо от состояния хранилища-приемника;
— минимизировать влияние архитектурных различий между системами хранения;
— снизить риск загрузки несогласованных или частично обработанных данных;
— упростить процедуры восстановления и повторного запуска миграции в случае сбоев.

Таким образом, при миграции больших данных между архитектурно разнородными хранилищами использование промежуточного компромиссного хранилища является не просто рекомендательным, а необходимым условием обеспечения корректности и завершенности процесса обработки информации.

Проблема 4. Особенности извлечения данных из нестандартных источников

Отдельную группу проблем при обработке информации в процессе миграции данных составляют особенности извлечения данных из нестандартных источников. В отличие от классических сценариев, где источником данных выступают хранилища, предоставляющие формализованные интерфейсы доступа (например, API или прямое подключение к базе данных), в ряде практических случаев информация может находиться в системах, не предназначенных для автоматизированного извлечения данных.

К таким источникам относятся, в частности, корпоративные порталы, внутренние информационные системы и веб-приложения, доступ к которым ограничен и осуществляется исключительно через пользовательский веб-интерфейс. В подобных условиях прямой доступ к данным отсутствует, что делает невозможным применение стандартных методов миграции и требует использования альтернативных подходов извлечения информации.

В качестве одного из таких подходов применяются экранные роботы, реализующие автоматизированное взаимодействие с пользовательским интерфейсом системы-источника. Процесс извлечения данных в этом случае включает анализ структуры веб-страниц, выявление элементов, содержащих целевую информацию, и последующий парсинг данных. Извлеченная информация загружается во временное или промежуточное хранилище, после чего может быть использована в рамках основного процесса миграции, описанного ранее.

Данный сценарий характеризуется повышенной сложностью и нестабильностью, поскольку структура веб-интерфейса может изменяться без предварительного уведомления, а извлечение данных зависит от корректной интерпретации визуальных и структурных элементов страницы. Кроме того, процесс экранного извлечения, как правило, менее производителен по сравнению с использованием формализованных интерфейсов доступа к данным.

Особый интерес представляет тот факт, что на текущий момент отсутствуют автоматизированные гибридные методы, способные динамически определять оптимальный способ извлечения данных в зависимости от характеристик источника. В частности, не существует универсальных решений, которые могли бы автоматически выявлять наличие API и использовать его при возможности, либо переходить к методам экранного извлечения в случае отсутствия формализованных интерфейсов.

Указанное ограничение открывает перспективы для дальнейших исследований, направленных на разработку интеллектуальных гибридных методов извлечения данных, способных адаптивно выбирать стратегию миграции в зависимости от типа и свойств источника информации.

Проблема 5. Ограниченность вычислительных ресурсов при миграции больших данных

При миграции больших объемов данных основная вычислительная нагрузка, как правило, возникает на этапе выполнения операций обработки информации, таких как фильтрация, агрегация, сортировка и преоб-

разование данных. Именно данные операции требуют значительных вычислительных ресурсов и оказывают наибольшее влияние на производительность системы-источника.

Современные хранилища данных, особенно ориентированные на аналитические или транзакционные нагрузки, не всегда обладают возможностью эффективного распараллеливания выполнения подобных операций в контексте миграции. В ряде случаев архитектура хранилища или его конфигурация ограничивает масштабирование вычислений, что приводит к увеличению времени обработки данных и росту нагрузки на систему. Дополнительной проблемой является тот факт, что выполнение ресурсоемких операций обработки непосредственно в исходном хранилище может существенно ограничить доступ к вычислительным ресурсам для других пользователей и прикладных сервисов. В результате другие запросы могут выполняться с задержками либо вовсе быть отклонены.

В связи с указанными ограничениями на практике широко применяются внешние инструменты обработки больших данных, такие как распределенные вычислительные платформы, позволяющие выносить основную нагрузку за пределы системы-источника. Использование подобных инструментов дает возможность извлекать данные из исходного хранилища, выполнять над ними операции обработки и преобразования во внешней среде, а затем передавать результаты в промежуточное или целевое хранилище.

Одним из наиболее распространенных решений данного класса являются распределенные вычислительные фреймворки, обеспечивающие работу с внутренними структурами данных и предоставляющие механизмы оптимизации вычислений за счет параллельной обработки. Применение таких инструментов позволяет существенно снизить нагрузку на исходное хранилище и повысить масштабируемость процессов обработки информации при миграции больших данных.

В то же время использование внешних вычислительных платформ также имеет определенные ограничения. В отдельных случаях в таких системах могут отсутствовать специализированные методы обработки данных, доступные в исходном хранилище. Хотя подобные ситуации являются маловероятными и большинство базовых и расширенных операций обработки, как правило, поддерживаются, данный фактор должен учитываться при проектировании процесса миграции.

На практике это приводит к архитектуре, в которой данные извлекаются из исходного хранилища, обрабатываются во внешнем инструменте обработки больших данных, а затем загружаются в промежуточное хранилище, откуда осуществляется их дальнейшая передача в целевую систему. Такой подход позволяет оптимизировать использование вычислительных ресурсов, минимизировать влияние миграции на рабочие процессы и повысить устойчивость обработки информации.

В рамках данной работы были рассмотрены основные проблемы обработки информации, возникающие в процессе миграции больших данных между разнородными хранилищами. Проведенный анализ показал, что различия в поддерживаемых типах данных и моделях хранения, ограниченная переносимость методов обработки, нестабильность выполнения операций при больших объемах данных, особенности извлечения информации из нестандартных источников, а также ограниченность вычислительных ресурсов существенно усложняют процессы миграции и требуют комплексного подхода к их решению.

Рассмотренные в статье подходы и решения позволяют повысить устойчивость и управляемость процессов обработки информации при миграции данных, однако не обеспечивают полной гарантии успешного завершения миграции. Даже при использовании промежуточных хранилищ, внешних инструментов обработки и адаптации методов преобразования данных сохраняется вероятность возникновения дополнительных проблем, обусловленных спецификой конкретных систем хранения, особенностями инфраструктуры и условиями эксплуатации.

Следует отметить, что для выполнения миграций больших данных на практике зачастую требуется оперировать значительным количеством различных хранилищ, каждое из которых обладает собственной архитектурой, набором поддерживаемых типов данных и методов обработки информации. С увеличением числа задействованных хранилищ возрастает и сложность процесса миграции, а также вероятность возникновения новых проблем, не охваченных в рамках настоящего исследования.

В этой связи полученные результаты следует рассматривать как шаг к формированию более универсального подхода к миграции больших данных. Проведенное исследование позволяет обосновать целесообразность дальнейших работ, направленных на разработку гибридного адаптивного метода, способного динамически учитывать особенности источников данных, выбирать оптимальные способы извлечения и обработки информации, а также обеспечивать эффективную и устойчивую миграцию данных в условиях разнородных хранилищ.

Таким образом, представленное исследование формирует основу для дальнейшего развития методов миграции больших данных и может быть использовано как теоретическая и практическая база для создания более интеллектуальных и адаптивных решений в данной области.

ЛИТЕРАТУРА

1. Date C.J. An Introduction to Database Systems. 8th ed. // Pearson, 2003. 983 p. URL: https://www.pearson.com/en-us/subject-catalog/p/an-introduction-to-database-systems/P200000003415 (дата обращения: 14.01.2025)
2. Elmasri R., Navathe S. B. Fundamentals of Database Systems. 7th ed. // Pearson, 2016. 1272 p. URL: https://www.pearson.com/en-us/subject-catalog/p/fundamentals-of-database-systems/P200000003245 (дата обращения: 14.01.2025)
3. Дейт К.Дж. Введение в системы баз данных = An Introduction to Database Systems. 8-е изд. // М.: Вильямс, 2006. 1328 с. URL: https://www.ozon.ru/product/vvedenie-v-sistemy-baz-dannyh-v-2-t-komplekt-deyt-k-dzh-1438146150/?at=ywtA5WkJ6uEwZ6A7S1BxGI5sXnwnmTzrmylGT5ZIP3 (дата обращения: 14.01.2025)
4. Кренке Д. Теория и практика построения баз данных. 10-е изд. = Database Processing: Fundamentals, Design, and Implementation // СПб.: Питер, 2018. 976 с. URL: https://www.ozon.ru/product/redkaya-kniga-tverdyy-pereplet-teoriya-i-praktika-postroeniya-baz-dannyh-krenke-devid-2317964010/?at=gpt4ZWpNPsy1zp4fqxomo6cNm44NT0LVRrQs1vJoYp (дата обращения: 14.01.2025)
5. Кириллов В.В. Основы проектирования реляционных баз данных: учебное пособие // СПб.: БХВ-Петербург, 2018. 288 с. URL: https://rusneb.ru/catalog/010003_000061_3a0ac872e9ffdc72f928632e4b30104a/?ysclid=mkarq54pbj722922934 (дата обращения: 14.01.2025)
6. Inmon W.H. Building the Data Warehouse. 5th ed. // Wiley, 2019. 576 p. URL: https://books.google.ru/books?id=QFKTmh5IFS4C&printsec=frontcover&hl=ru#v=onepage&q&f=false (дата обращения: 14.01.2025)
7. Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. // Wiley, 2013. 600 p. URL: https://www.wiley.com/en-us/The+Data+Warehouse+Toolkit%3A+The+Definitive+Guide+to+Dimensional+Modeling%2C+3rd+Edition-p-9781118530801 (дата обращения: 14.01.2025)
8. Кимбалл Р., Росс М. Хранилище данных. Проектирование = The Data Warehouse Toolkit. 2-е изд. // М.: Диалектика, 2020. 576 с. URL: https://books.google.ru/books/about/The_Data_Warehouse_Toolkit.html?id=4rFXzk8wAB8C&redir_esc=y (дата обращения: 14.01.2025)
9. Marz N., Warren J. Big Data: Principles and best practices of scalable realtime data systems // Manning Publications, 2015. 328 p. URL: https://www.manning.com/books/big-data (дата обращения: 14.01.2025)
10. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems // O'Reilly Media, 2017. 616 p. URL: https://www.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/ (дата обращения: 14.01.2025)
11. Sadalage P.J., Fowler M. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence // Addison-Wesley Professional, 2012. 192 p. URL: https://www.informit.com/store/nosql-distilled-a-brief-guide-to-the-emerging-world-9780321826626 (дата обращения: 14.01.2025)
12. Грофф Дж.Р., Вайнберг П.Н., Оппел Э.Дж. SQL: Полное руководство. 3-е изд. = SQL: The Complete Reference // М.: Диалектика, 2021. 960 с. URL: https://www.ozon.ru/product/sql-polnoe-rukovodstvo-3-e-izd-1147649373/ (дата обращения: 14.01.2025)
13. Codd E.F. A Relational Model of Data for Large Shared Data Banks // Communications of the ACM, 1970, Vol. 13, Issue 6, pp. 377–387. URL: https://doi.org/10.1145/362384.362685 (дата обращения: 14.01.2025)
14. Chaudhuri S., Dayal U. An overview of data warehousing and OLAP technology // ACM Sigmod Record, 1997, Vol. 26, Issue 1, pp. 65–74. URL: https://doi.org/10.1145/248603.248616 (дата обращения: 14.01.2025)
15. Большие данные. Big Data. Учебник для СПО: Макшанов, Журавлев, Тындыкарь // Лань, 2022 188 с. https://www.labirint.ru/books/795544/?ysclid=mkas40ac8h636323968 (дата обращения: 14.01.2025)

Журнал «Современная наука: актуальные проблемы теории и практики»