УДК 004.67

ПРЕОДОЛЕНИЕ ПРОБЛЕМ BIG DATA С ПОМОЩЬЮ DATA LAKE

Е.А. Алуев
Инженер-исследователь
АТЕК, бакалавр
технических наук,
alooeff@gmail.com

Е.А. Алуев

Выпускник Брестского государственного технического университета. Область научных интересов связана с облачными технологиями, машинным обучением и мульти-агентным моделированием.

Аннотация. Произведен анализ актуальных проблем предприятий по хранению и обработке неструктурированных данных. Рассмотрены способы решения этих проблем при помощи использования озер данных, рассмотрены особенности их использования. Сделаны выводы о перспективах перехода на использование озер данных в некоторых актуальных областях.

Ключевые слова: Big Data, Data Lake, озеро данных, обработка информации. база данных.

Введение. Фирмы, независимо от вида деятельности, имеют много разных видов данных. Есть PDF-файлы со спецификациями для выпускаемой продукции, CSV-файлы с транзакциями продаж, клиентами и товарами, а также JSON-файлы с обзорами продуктов и журналами. Основная проблемы хранения и обработки данных предприятия - их разнородность.

Современный путь решения этой проблемы - вводить эти данные в озеро данных (Data Lake) и выводить из него, объединять их все вместе и получать необходимую информацию.

Что именно представляет собой озеро данных? Это централизованное хранилище, которое позволяет хранить структурированные и неструктурированные данные в любом масштабе. Это означает, что сколько бы данных ни было, возможно добавить их все в централизованное хранилище. В озере данных можно хранить данные как есть, без необходимости их предварительной структуризации и запускать различные типы аналитики. Основная идея Data Lake заключается в том, что нет необходимости заранее определять схему, то есть структуру этих данных, поэтому возможно обрабатывать необработанные данные, не зная, какие сведения нужно будет получить в будущем.

Ключевые характеристики озера данных. Масштабируемость. Озера данных могут хранить петабайты данных и масштабироваться по мере необходимости без существенного снижения производительности. Эта возможность имеет важное значение в современном мире, управляемом данными, где объем данных, которые мы генерируем и анализируем, продолжает расти. Далее следует экономическая эффективность. Озера данных часто используют объектное хранилище, которое стоит дешевле традиционного хранилища, что делает его экономически эффективным решением для хранения огромных объемов данных. Также гибкость. В отличие от традиционных баз данных, которые требуют форматирования и структурирования данных предопределенным образом, озера данных могут хранить данные в различных форматах, включая PDF, XML, JSON, изображения и многое другое. Эта гибкость позволяет хранить данные из различных источников. И, наконец, доступность.

Централизуя хранение данных, озера данных облегчают различным потребителям данных, таким как специалисты по данным и аналитики, быстрый доступ и анализ больших объемов данных с помощью инструментов и фреймворков, которые интегрируются в среду озера данных. Озера данных служат основой для расширенной аналитики, предоставляя специалистам по данным гибкость для доступа и анализа различных форматов данных. Эта среда идеально подходит для разработки и совершенствования моделей и алгоритмов машинного обучения. Аналогичным образом озера данных все чаще используются для приложений на основе ИИ. Возможность хранить и управлять большими наборами данных из различных данных позволяет группам данных обучать более эффективные модели ИИ. Озера данных часто используются для помощи в обработке и аналитике данных в реальном времени, позволяя группам быстро реагировать на условия бизнеса и рынка. Например, розничные компании анализируют взаимодействие с клиентами и транзакции, поскольку они предлагают персонализированные акции или динамическое ценообразование. В эпоху ИИ и расширенной аналитики традиционные системы управления данными часто не могут справиться с масштабом или обеспечить необходимую гибкость. Озера данных решают проблемы, предлагая более гибкий и масштабируемый подход к их хранению и анализу. Они адаптируются к меняющимся аналитическим требованиям по мере их возникновения.

Необходимость создания озер данных. С появлением Интернета, электронной коммерции и социальных сетей традиционные системы хранения и обработки данных были перегружены. Потребность в технологии, которая обеспечивала бы не только объем, но и разнообразие и скорость обработки данных, стала очевидной, но за этим стояли значительные проблемы. Традиционные базы данных и хранилища данных были структурированы для последовательной, предсказуемой загрузки структурированных данных, таких как имена клиентов и контактная информация, финансовые транзакции и информация о запасах и т. д. Однако новые типы данных, такие как электронные письма, видео, сообщения в социальных сетях, данные датчиков и т. д., были неструктурированными и не вписывались в строки и столбцы. Требовалось решение, которое было бы гибким и разнообразным, как и сами данные. На основании Google MapReduce и Google File System, был разработан Apache Hadoop для хранения и обработки огромных объемов данных в кластерах серверов массового потребления. Он обеспечил хранение их в необработанной форме в распределенных средах. Это было концептуальным началом того, что позже назовут озером данных. Сам термин «озеро данных» был придуман Джеймсом Диксоном, техническим директором Pentaho. Он использовал его для описания большого хранилища необработанных данных, хранящихся в своем исходном формате до тех пор, пока они не понадобятся. В отличие от хранилищ данных, озера данных были разработаны так, чтобы быть гибкими, масштабируемыми и способными обрабатывать сложность и характер данных реального мира. Эволюция озер данных также тесно связана с быстрым развитием облачных вычислений. Поскольку облачные сервисы, такие как Amazon Web Services, Microsoft Azure и Google Cloud, становились все более надежными, они предоставили идеальную экосистему для озер данных [1]. Эти платформы предлагают масштабируемые, безопасные и экономичные решения для хранения данных, позволяя компаниям и командам всех размеров создавать озера данных без необходимости в дорогостоящей локальной инфраструктуре. Концепция демократизации данных, делающая данные доступными для всех без посредников, является еще одним ключевым фактором в развитии озер данных. Позволяя хранить данные в центральном месте, доступном для различных заинтересованных сторон в компании, озера данных обеспечивают среду для совместной работы, в которой идеи и принятие решений доступны для всех отделов.

Основные компоненты архитектуры проще представлять в виде уровней.

Уровень хранения является основой озера данных. Он предназначен для хранения огромного количества необработанных данных в различных форматах, от структурированных до неструктурированных. Этот уровень должен быть

высокомасштабируемым, чтобы приспосабливаться к росту объема данных, не снижая производительности. Есть три основные характеристики, которыми должен обладать уровень хранения: он должен иметь возможность расширяться для хранения больших объемов данных по мере необходимости, он должен быть оптимизирован, чтобы поддерживать управляемые затраты, он должен иметь возможность поддерживать несколько форматов и структур данных.

Уровень приема отвечает за прием данных из различных источников, будь то потоковая передача в реальном времени или пакетами. Основные характеристики уровня приема: он должен гарантировать, что данные собираются точно и надежно без потерь, он должен быть способен обрабатывать большие объемы данных на высокой скорости, он также должен поддерживать различные методы приема для работы с различными источниками данных.

Уровень обработки — это то, где данные преобразуются, агрегируются или объединяются. Этот уровень позволяет применять бизнес-правила и преобразования данных для получения информации из необработанных данных, хранящихся в озере, в дальнейшем. Этот уровень можно рассматривать как часть уровня приема или отдельно.

Основные характеристики уровня обработки: он должен быть оптимизирован для быстрой обработки больших наборов данных, он должен поддерживать несколько потребностей обработки от пакетной обработки больших наборов данных до аналитики в реальном времени, он должен иметь возможность масштабирования для удовлетворения требований к обработке по мере увеличения объема или сложности данных.

Уровень управления гарантирует, что озеро данных остается организованной, безопасной и хорошо управляемой средой. Он включает такие аспекты, как управление метаданными, каталогизация данных, безопасность и соответствие требованиям. Основные характеристики надежного уровня управления: он должен предоставлять механизмы доступа к данным и авторизации для защиты конфиденциальных данных, он также должен позволять вам отслеживать происхождение данных и метаданные для прозрачности и проверяемости, и он должен гарантировать, что методы управления данными соответствуют нормативным требованиям.

Уровень аналитики — это то место, где данные становятся применимыми, предоставляя инструменты и интерфейсы для конечных пользователей для запроса, визуализации и анализа данных, хранящихся в озере. Этот уровень обеспечивает следующее: удобные для пользователя инструменты для доступа к данным и их анализа, совместимость с различными инструментами бизнес-аналитики и аналитики и обычно оптимизирован для быстрого и эффективного выполнения запросов.

Типы хранилищ. В озере данных выбранные типы хранения играют важную роль в том, как данные поддерживаются, доступны и используются. Понимание этих типов очень важно, поскольку это может помочь оптимизировать стратегию работы с данными для баланса стоимости, доступности и производительности. Основные типы хранения, используемые в озерах данных:

Объектное хранилище на сегодняшний день является самым важным типом хранения для озер данных, особенно в облачных средах. Здесь данные хранятся в виде отдельных единиц или объектов, каждый из которых имеет уникальный идентификатор, что позволяет извлекать их из распределенного пула ресурсов хранения. Этот тип хранилища отличается высокой масштабируемостью и экономичностью, что делает его идеальным для хранения огромных объемов неструктурированных данных, таких как фотографии, видео и большие наборы данных. Примерами являются Amazon S3, Google Cloud Storage и Microsoft Azure Blob Storage.

Файловое хранилище организует данные в иерархии файлов и каталогов, что позволяет вам получать доступ к данным и управлять ими способом, который знаком большинству пользователей компьютеров. Хотя он не так масштабируем, как объектное

хранилище, файловые системы хранения, такие как сетевое хранилище или NAS, часто используются для небольших или устаревших приложений, которым требуются традиционные интерфейсы файловой системы.

Блочное хранилище делит данные на блоки одинакового размера, каждый из которых имеет свой собственный уникальный адрес, но без каких-либо метаданных для предоставления большего контекста о содержимом. Этот тип хранилища часто используется для приложений с высокой производительностью, таких как базы данных или системы ERP. Однако он может быть более дорогим и сложным в управлении, что делает его менее распространенным для общего использования озер данных.

Озера данных также могут интегрировать **гибридные решения** для хранения, которые объединяют элементы объектного, файлового и блочного хранения. Эти решения разработаны для обеспечения гибкости объектного хранилища с преимуществами производительности блочного хранилища. Например, распределенная файловая система Hadoop или HDFS или другие программно-определяемые решения для хранения могут эффективно управлять данными в различных типах хранилищ [2]. Каждый тип хранилища в озере данных имеет свой собственный набор преимуществ и идеальных вариантов использования.

Хостинг хранилища. Традиционные локальные решения для хранения данных включают физическую инфраструктуру, расположенную в центре обработки данных компании. Локальное хранилище обеспечивает полный контроль над средой хранения, что актуально для отраслей с высокими требованиями к безопасности и соблюдению требований. Преимуществами этого подхода являются повышенная безопасность и контроль над оборудованием, потенциально более высокая производительность для локализованной обработки данных. С точки зрения вариантов использования это чаще всего наблюдается в строго регулируемых отраслях, таких как финансы и здравоохранение, где безопасность являются приоритетом номер один. Облачное хранилище предоставляет масштабируемые и гибкие варианты хранения данных, размещенные на внешних платформах, предоставляемых такими поставщиками, как Amazon Web Services, Microsoft Azure или Google Cloud platform. Сильными сторонами этого подхода являются масштабируемость, экономичность и доступность из любой точки мира. Облачное хранилище также предлагает различные сервисы, которые можно интегрировать, такие как ИИ и аналитические инструменты. Кроме того, оно переводит затраты из капитальных в операционные, делая расходы более предсказуемыми и управляемыми. С точки зрения вариантов использования облачное хранилище используется в стартапах и предприятиях, которым нужны масштабируемые и гибкие решения без накладных расходов на поддержание физической инфраструктуры. Гибридные решения объединяют локальное и облачное хранилище, чтобы использовать преимущества обеих сред. Такой подход позволяет хранить конфиденциальные данные локально, используя облако для менее конфиденциальных данных. Сильными сторонами этого подхода являются гибкость, масштабируемость и возможность хранить критически важные данные внутри компании, при этом по-прежнему используя облачные технологии. С точки зрения вариантов использования гибридное хранилище отлично подходит для компаний, переходящих в облако; тех, у кого есть динамические потребности в хранении или которым требуется баланс между соответствием требованиям, безопасностью и масштабируемостью.

Структуры папок. Иерархии папок помогают организовать данные структурированным образом, упрощая навигацию, управление и извлечение. Хорошо продуманная структура очень важна в таких средах, как озера данных, где разнообразие и объем данных могут быстро привести к хаосу. Эффективные структуры папок сокращают время и вычислительные ресурсы, необходимые для поиска и доступа к данным. Хорошая структура также упрощает применение политик управления, таких как хранение данных, архивирование и управление жизненным циклом. Она также позволяет применять

политики безопасности на уровне папок, что невероятно важно для соответствия требованиям и защиты данных. Создание эффективной иерархии папок в вашем озере данных подразумевает понимание характера данных, потребностей пользователей и операционной динамики вашей компании. Основой хорошей иерархии папок является логическая группировка, которая классифицирует данные таким образом, чтобы они соответствовали тому, как пользователи взаимодействуют с ними. Также важно подумать о разделении данных на основе требований конфиденциальности и доступа, чтобы оптимизировать управление безопасностью. Еще одна концепция, которую следует учитывать при проектировании озера данных, - это архитектура медальона. Этот подход организует данные в разные слои, как правило, бронзовый, серебряный и золотой. Бронзовый слой — это место, где необработанные данные поступают из разных источников. Он хранится в своем исходном формате и часто используется для исторических записей. Серебряный слой содержит чистые и обогащенные данные. Здесь применяются преобразования, чтобы сделать данные более пригодными для анализа, а золотой слой — это последний тщательно отобранный слой данных. Он оптимизирован для бизнес-аналитики и аналитики, часто содержит агрегированные и специфичные для бизнеса данные. Еще один аспект, который следует учитывать, — это проектирование иерархий, которые будут достаточно гибкими для учета будущих изменений объема данных, новых типов данных и меняющихся бизнес-потребностей. Для данных, привязанных ко времени, таких как журналы или данные транзакций, организация папок по годам, кварталам, месяцам или дням может значительно ускорить запросы, привязанные к времени.

Форматы файлов. В озерах данных форматы файлов определяют, как данные хранятся и как осуществляется к ним доступ. Выбор формата файла может повлиять на все: от затрат до производительности запросов. Наиболее распространенные форматы:

CSV — один из самых простых и широко используемых форматов файлов. Он основан на тексте, прост для понимания и поддерживается почти всеми приложениями для обработки данных. Однако, поскольку в файлах CSV отсутствует какой-либо тип сжатия данных или принудительное применение схемы, они не оптимизированы для больших наборов данных или сложных запросов.

Файлы **JSON** очень гибкие и поддерживают иерархический, полуструктурированный формат данных, который идеально подходит для данных в различных схемах. JSON широко используется в веб-приложениях и для обмена данными между серверами и веб-клиентами. Хотя JSON легко читается и гибок, он может быть менее эффективным для хранения и запросов больших наборов данных.

Apache Parquet — это столбчатый формат файла хранения, оптимизированный для использования в озерах данных. Он предлагает отличные схемы сжатия и кодирования, которые сокращают накладные расходы на хранение и повышают производительность запросов, позволяя эффективно читать определенные столбцы. Parquet особенно хорошо подходит для сложных, вложенных структур данных и широко используется в фреймворках обработки больших данных, таких как Apache Hadoop, Spark и Presto.

ORC — это еще один столбчатый формат хранения, который обеспечивает высокоэффективный способ хранения и обработки данных. Как и Parquet, ORC поддерживает расширенное сжатие и кодирование, что делает его идеальным для высокоскоростных запросов. Файлы ORC предназначены для работы в средах, требующих тяжелых операций чтения и записи, что делает их популярными в системах, которые выполняют частое обновление данных.

Apache Avro предназначен для сериализации данных, предоставляя компактный и быстрый двоичный формат данных. Он также поддерживает эволюцию схемы, когда схемы хранятся внутри данных, что позволяет легко интегрировать сериализованные данные с приложениями. Avro — хороший выбор для конвейеров данных, которым требуются надежные, сложные структуры данных и быстрое получение данных.

Каждый формат файла предлагает разные преимущества и может быть лучшим выбором в зависимости от конкретных требований. CSV и JSON превосходны для простоты и гибкости. С другой стороны, Parquet, ORC и Avro обеспечивают лучшую производительность и эффективность для крупномасштабных операций с данными. Хотя это может быть оптимальным для хранения, возможно добавление практически любого типа файла в озеро данных.

Преобразование данных. Озера данных являются идеальной архитектурой для хранения необработанных данных, но необработанные данные все равно чаще всего необходимо подготовить для расширенной аналитики. Вот где вступает в дело преобразование данных. Преобразование данных в озерах данных включает изменение формата, структуры или значений данных. Основная цель — преобразовать необработанные данные в формат, который больше подходит для запросов и аналитики. Преобразование данных имеет несколько дополнительных преимуществ. Оно может улучшить качество данных за счет исправления ошибок, заполнения пропущенных значений и стандартизации форматов. Оно также может улучшить согласованность данных, упрощая интеграцию с другими источниками данных. Рассмотрим наиболее распространенные типы преобразования:

Очистка включает исправление или удаление неточных, неполных или противоречивых данных из набора данных. Примерами методов очистки являются стандартизация, которая требует корректировки данных для соответствия таким нормам, как форматирование дат, и дедупликация, которая включает в себя выявление и удаление дубликатов записей для предотвращения избыточности. Методы нормализации стандартизируют форматы данных, значения и диапазоны для добавления согласованности во всем наборе данных. Это особенно полезно в озерах данных из-за разнообразия источников и форматов данных. Примером нормализации будет преобразование всех валютных значений в единый валютный формат.

Агрегация включает в себя суммирование данных в более управляемую форму. Этот метод действительно помогает ускорить анализ и отчетность. Примером агрегации будет суммирование данных о продажах для предоставления ежемесячных итогов продаж.

Объединение означает объединение данных из разных источников или наборов данных для предоставления единого представления или единого производного набора данных. Этот метод также помогает ускорить и упростить аналитические запросы, поскольку объединения выполняются не при каждом анализе. Примером может служить объединение информации о клиентах из систем CRM с данными о транзакциях из баз данных продаж. Поскольку потребности в обработке данных становятся все более сложными, для обработки сложных сценариев часто используются передовые методы.

Обогащение данных дополняет исходный набор данных дополнительными источниками данных для предоставления более точной информации. Например, включение демографической информации в существующий набор данных для анализа моделей покупок в различных сегментах клиентов. Алгоритмы машинного обучения также могут применяться для преобразования или обогащения данных на основе изученных моделей и прогнозов. Примером будет использование методов кластеризации для сегментации клиентов на основе поведения. Создание чистых и хорошо отформатированных наборов данных, агрегирование метрик, а также слияние различных источников данных повышают элегантность и эффективность последующих аналитических запросов. Это также повышает точность моделей машинного обучения. По сути, преобразование данных действует как этап предварительной обработки, который позволяет в конечном итоге извлекать значимые идеи из необработанных данных.

Обработка ошибок, ведение журнала и мониторинг. Рано или поздно конвейеры данных сломаются. Обработка ошибок при приеме данных включает в себя прогнозирование, обнаружение и устранение ошибок, возникающих в процессе загрузки

данных. Это предотвращает попадание поврежденных данных в озеро данных и гарантирует, что процессы приема не будут нарушены. Некоторые примеры стратегий обработки ошибок включают реализацию логики повторных попыток для временных ошибок и настройку систем уведомлений для немедленных оповещений о сбоях. Одним из важнейших аспектов обработки ошибок является ведение журнала. Ведение журнала включает в себя запись событий, которые происходят во время процесса приема данных. Оно обеспечивает видимость производительности конвейеров данных и необходимо для диагностики проблем после их возникновения. Журналы должны включать такие сведения, как временная метка, источник данных, описание ошибки, как можно меньше и точнее, и часть процесса приема, где произошла ошибка. С точки зрения передовой практики, вы должны гарантировать, что все этапы процессов приема охватываются журналом, от точек ввода данных до окончательной загрузки. Вы также можете использовать инструменты управления журналами для агрегации, хранения и анализа данных журнала. Такие инструменты, как Elasticsearch, Logstash и Kibana, ELK Stack, могут предоставить мощные аналитические данные о журнале. Наконец, убедитесь, что журналы содержат достаточно информации, чтобы быть полезными, но не нарушают законы о конфиденциальности или правила защиты данных. Надежно храните журналы, чтобы предотвратить несанкционированный доступ. Непрерывный мониторинг процесса приема данных помогает обнаруживать и решать такие проблемы, как потеря данных, задержки или неточности, прежде чем они окажут большее влияние. Лучшей практикой здесь является внедрение инструментов мониторинга в реальном времени, которые могут предоставлять оповещения и панели мониторинга для визуализации работоспособности и производительности конвейеров приема данных. Этот проактивный подход помогает выявлять узкие места или сбои на ранних этапах процесса. Apache NiFi предлагает систему потока данных в реальном времени со встроенными панелями мониторинга для мониторинга пропускной способности данных, задержки и работоспособности системы, а Elastic Stack, включающий Elasticsearch, Logstash и Kibana, идеально подходит для регистрации процессов приема данных и визуализации потоков данных и показателей производительности. Prometheus и Grafana — это инструменты, которые обычно используются для мониторинга производительности систем приема данных путем извлечения метрик из конвейеров данных и отображения их на интуитивно понятных панелях мониторинга.

Выводы. Разница между классическим хранилищем данных и озером данных заключается в том, что хранилище данных предварительно классифицирует данные в точке входа, что приводит к потере времени и энергии, в то время как озеро данных принимает информацию с использованием нереляционной базы данных и не обрабатывает, и не классифицирует данные на входе. Это позволяет решить пробелу пропускной способности хранилища и проблему совместимости форматов входящих данных. В совокупности с возможностями проведения аналитики получаем удобный инструмент для предприятий с разнообразными данными и требованиями по их сбору, хранению и обработки.

Озеро данных может объединить данные о клиентах из платформы CRM с аналитикой социальных сетей, маркетинговой платформы, содержащей историю покупок, и инцидентов. Таким образом, компания сможет определить контингент клиентов, приносящих наибольшую прибыль, причины их оттока, а также подобрать акции или вознаграждения, которые повысят лояльность.

Озеро данных позволяет научно-исследовательским группам проверять гипотезы, уточнять предположения и оценивать результаты, например, выбирать правильные материалы при разработке продукта, что способствует повышению его производительности, проводить геномные исследования, в результате которых можно получить более эффективные лекарства, или понять готовность покупателей платить за различные атрибуты.

Интернет вещей (IoT) предоставляет больше возможностей для сбора данных о таких процессах, как производство, причем данные поступают в реальном времени от устройств, подключенных к Интернету. В озере данных удобно хранить и анализировать данные IoT, генерируемые машинами, и находить способы снижения операционных затрат и повышения качества работы.

Список литературы

[1] Naresh Lokiny Comparative Study of Cloud Providers (AWS, Azure, Google Cloud) using Artificial Intelligence with DevOps. International Journal of Science and Research (IJSR) Volume 8 Issue 8, August 2019 p 2326 – 2329.DOI: g/10.21275/SR24724151213 2

[2] Raj, Anushree & D'Souza, Rio. (2019). A Review on Hadoop Eco System for Big Data. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 343-348. DOI:10.32628/CSEIT195172.

Авторский вклад

Алуев Евгений Александрович – проведение исследования существующего рынка озер данных и их текущих реализаций.

OVERCOMING BIG DATA CHALLENGES WITH DATA LAKE

Eugene Alooeff
R&D Engineer ATEK,
Bachelor of Computer Science,
Bachelor.

Abstract. The analysis of current problems of enterprises on storage and processing of unstructured data is made. The ways of solving these problems by means of using data lakes are analyzed, the features of their use are considered. Conclusions are made on the prospects of transition to using data lakes in some markets.

Keywords: Big Data, Data Lake, data processing, database.