ORES

Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 004.62

ИСПОЛЬЗОВАНИЕ ОЗЕРА ДАННЫХ ПРИ ПОЛУЧЕНИИ ИНФОРМАЦИИ С ПОМОЩЬЮ ПАРСИНГА ИНТЕРНЕТ-САЙТОВ

Морозова Вероника Сергеевна,
студент кафедры «Системы обработки информации», Калужский филиал Московского государственного технического университета имени Н.Э.Баумана, morozovavs@student.bmstu.ru

Антипова Ольга Викторовна,
старший преподаватель кафедры «Системы обработки информации», Калужский филиал Московского государственного технического университета имени Н.Э.Баумана, antipovaov@bmstu.ru

Аннотация

В статье рассмотрен подход для получения, хранения и обработки данных с помощью озера данных.

Ключевые слова: озеро данных, хранение данных, управление данными.

THE USE OF DATA LAKE TECHNOLOGY IN OBTAINING INFORMATION BY PARSING INTERNET SITES

Veronika S. Morozova,
student of the Department of Information Processing Systems, Kaluga Branch of the Bauman Moscow State Technical University, morozovavs@student.bmstu.ru

Olga V. Antipova,
Senior Lecturer at the Department of Information Processing Systems, Kaluga Branch of the Bauman Moscow State Technical University, antipovaov@bmstu.ru

ABSTRACT

The article considers an approach for obtaining, storing and processing data using a data lake.

Keywords: data lake, data storage, data management.

Введение

В эпоху цифровизации, когда поток данных растет с каждым днем, традиционные методы хранения уже не справляются с ними. Данные становятся неоднородными, делая невозможным создание единой картины и принятие основанных решений. Озеро данных - это технология, которая собирает информацию в единое хранилище, решает проблему увеличения объема данных, помогает улучшить интеграцию системами и является более дешевым хранилищем. Оно также решает проблему разрозненности данных, предоставляя центральную платформу для удовлетворения различных потребностей в хранении ресурсов. Это позволяет не только справиться с растущим объемом данных, но и создать единую систему для их анализа и использования. Однако, озеро данных требует тщательной подготовки и регулирования: необходимо управлять метаданными, обеспечивать безопасность, контролировать доступ и следить за жизненным циклом данных, управлять механизмами централизованного контроля доступа. Целью является получение обработанных данных из неотфильтрованных в озере данных и выявление преимуществ и недостатков в данного подхода для управления данными.

Полученные результаты могут быть полезны для разработчиков, аналитиков, специалистам по машинному обучению в области хранения данных.

Озеро данных - это централизованное хранилище, предназначенное для хранения, обработки и защиты больших объемов структурированных, полуструктурированных и неструктурированных данных.[1] В отличие от традиционных хранилищ данных (DWH), озеро данных не требует предварительного определения схемы данных и может хранить информацию в ее исходном формате, также не имеет ограничений по размеру и позволяет обрабатывать в любом виде информацию. Это позволяет использовать данные для различных целей, включая аналитику, машинное обучение и искусственный интеллект.

Одним из свойств озер данных является масштабируемость, то есть размер может расти по мере загрузки данных пользователями. Другое свойство - гибкость, но именно она в значительной степени может привести к неупорядоченным данным и привести к краху обработки данных.

Для повышения эффективности и управления данными введены слои данных: бронзовый (хранит данные в том же формате, что и в исходной системе), серебряный (содержит очищенные данные и преобразованные данные) и золотой (конечные данные для пользования). [2] Каждый слой является основой для следующего.

Деление на слои имеет ряд преимуществ:

1. Безопасное решение. Разделение данных на слои позволяет настроить уровни доступа для разных пользователей, предоставляя только необходимый объем информации;
2. Простое отслеживание истории данных. Слоевая структура позволяет легко отследить путь данных, их изменения и историю;
3. Использование одних и тех же наборов данных для нескольких вариантов применения;
4. Масштабируемость;

Недостатки:

При больших объёмах могут перерасти в сложно управляемые данные. Решение использования разделения на слои необходимо зависит от факторов:

1. Существует несколько или много источников данных
2. Необходима гибкость решения
3. Нет варианта использования собранных данных.[3]

Для более эффективного использования усовершенствовали озеро данных, которые стали совмещать в себе свойства DWH (Data Warehouse) и Data Lake. Хранилище данных - это централизованное место хранения и анализа структурированных ресурсов для конкретных целей, связанных с бизнес-аналитикой. Оно содержит структурированные

данные, что приводит к большим затратам на поддержку по сравнению с озерами данными.

В новом подходе к озеру данных используется объектное хранилище, поэтому оно предоставляет все его достоинства с точки зрения масштабируемости, надежности, производительности, а также содержит возможность хранения неструктурированных данных.

Для управления всеми компонентами производят разделение по уровням для объединения по схожести функционала и создание иерархии, где получатели данных находятся на самом верху, а источники самих данных (т. е. необработанными) - на самом низком уровне иерархии.[3]

Уровень потребления: содержит инструменты для анализа данных, приложения и модели машинного обучения, предоставляющие доступ к данным.

Семантический уровень: предоставляет инструменты для обнаружения и управления метаданными.

Уровень обработки: содержит вычислительные ресурсы для обработки запросов, ETL-процессов и машинного обучения. Здесь могут происходить сложные преобразования, ETL (Extract, Load, Transform) процесс. Этот метод часто используется при работе с большими объемами данных, поскольку он может усиливать возможности обработки ресурсов современных платформ. Он обеспечивает ряд преимуществ, таких как управление байтовыми данными, повышение скорости обработки информации, производительности и экономичности. [4]

Уровень хранения: обеспечивает хранение данных в объектном хранилище.

Уровень приема: обрабатывает входящие данные, проводит простые и сложные преобразования, входят методы для получения данных и может получать данные по расписанию.

Уровень источников данных: Содержит инфраструктуру для получения данных из различных источников. [5]

Экспериментальная_часть

1. Ресурсы данных: данные полученные из интернет - сайтов, собранные методом парсинга.
2. Хранилище собранных данных - объектное хранилище MinIO, где собраны данные и хранятся в форматы CSV или JSON. Для отправки данных используется класс S3FileSystem в Python (указывается информация о подключении и выполняет операции с файловой системой).
3. Обработка данных: очистка и преобразование данных: удаление дубликатов, некорректных значений, преобразование к единому типу или формату данных, вычисления. Здесь для построения ETL процесса используются технологии в Apache Airflow, который будет устанавливать расписание преобразования накопившихся данных с помощью Scheduler, DAG, TASK.
4. Загрузка данных в реляционную СУБД: подключение к СУБД (в данном случае используется PostgreSQL), отправка обработанных данных в таблицы СУБД (Insert запросы).

Рис 1 - схема уровней

В PostgreSQL проверяем выполнения запроса с преобразованными данными из озера данных. Время запроса составило 3744, 497 мс, количество записей составляет 10 млн.

```
EXPLAIN (ANALYZE, BUFFERS)
SELECT *
FROM public.competitors;
```

	QUERY PLAN text
1	Seq Scan on temp_competitors (cost=0.00..149900.10 rows=713810 width=1584) (actual time=3.717..2888.758 rows=10000000 loops=1)
2	Buffers: local read=142762 written=1024
3	Planning Time: 0.068 ms
4	Execution Time: 3744.497 ms

Рис 2 - результат выполнения запроса с преобразованием данных

Используя предыдущей запрос для данных, которые собирались непосредственно в PostgreSQL, получаем выполнение запроса в два раза больше, чем при использовании озера

	QUERY PLAN text
1	Seq Scan on temp_competitors (cost=0.00..149900.10 rows=713810 width=1584) (actual time=0.050..5377.854 rows=10000000 loop…
2	Buffers: local read=142762 dirtied=142762 written=142761
3	Planning:
4	Buffers: shared hit=14
5	Planning Time: 0.192 ms
6	Execution Time: 6275.206 ms

потребления	PostgreSQL	
Уровень обработки	ETL: Обработка полученных данных для модели данных	Загрузка данных в СУБД Postgersql
Уровень хранения	Объктное хранилище MinIO	
Уровень приема	Извлечение данных: получение данных по определенным срокам	
Ресурсы данных	Источники данных: интернет-сайты	

данных, учитывая то, что не все данные попали в БД в нужном виде или вообще не попали.

Рис 3 - результат выполнения запроса без преобразованием данных

Выводы: исходя из полученных результатов экспериментов, получили: время получения результатов для пользования с обработанными данными из озера данных меньше, чем при использовании собранных и нефильтрованных данных в реляционной модели для одного итого же количества записей; при изменении структуры БД ресурсы, находящиеся в озере данных не исчезнут, это дает еще одно преимущество озеру данных. И также при добавлении новых полей в таблицы БД данные, которые будут сохранять историю данных, что позволит получить более точные вычисления для дальнейшего пользования. Из этого следует, что озеро данных - это гибкое решение для регулирования хранения информации.

Список литературы:

1. How to Organize your Data Lake // Microsoft URL: https://techcommunity.microsoft.com/t5/data-architecture-blog/how-to-organize-your-data-lake/ba-p/1182562 (дата обращения: 10.10.2024).
2. How to Build a Data Lake: Step-by-Step Guide // intercept.cloud URL: https://intercept.cloud/en-gb/blogs/how-to-build-a-data-lake (дата обращения: 20.10.2024).
3. Building a Data Lake on AWS: A Comprehensive Guide // TrackIT URL: https://trackit.io/guide-building-data-lake-on-aws/ (дата обращения: 15.10.2024).
4. Data Lake ETL: Integrating Data From Multiple Sources // Integrate.IO URL: https://www.integrate.io/blog/etl-tools-in-data-lake (дата обращения: 12.10.2024)
5. Angling for Insight in Today's Data Lake // Amazon.com URL: https://s3-ap-southeast-1.amazonaws.com/mktg-apac/Big+Data+Refresh+Q4+Campaign/Aberdeen+Research+-+Angling+for+Insights+in+Today's+Data+LakeA2 (дата обращения: 21.10.2024).

References:

1. How to Organize your Data Lake // Microsoft URL: https://techcommunity.microsoft.com/t5/data-architecture-blog/how-to-organize-your-data-lake/ba-p/1182562 (дата обращения: 10.10.2024).
2. How to Build a Data Lake: Step-by-Step Guide // intercept.cloud URL: https://intercept.cloud/en-gb/blogs/how-to-build-a-data-lake (дата обращения: 20.10.2024).
3. Building a Data Lake on AWS: A Comprehensive Guide // TrackIT URL: https://trackit.io/guide-building-data-lake-on-aws/ (дата обращения: 15.10.2024).
4. Data Lake ETL: Integrating Data From Multiple Sources // Integrate.IO URL: https://www.integrate.io/blog/etl-tools-in-data-lake (дата обращения: 12.10.2024)
5. Angling for Insight in Today's Data Lake // Amazon.com URL: https://s3-ap-southeast-1.amazonaws.com/mktg-apac/Big+Data+Refresh+Q4+Campaign/Aberdeen+Research+-+Angling+for+Insights+in+Today's+Data+LakeA2 (дата обращения: 21.10.2024).