Манохин А.К. Задача непрерывного обновления в системах парсинга
и агрегации данных // Международный журнал информационных технологий и
энергоэффективности. – 2023. –
Т. 8 № 3(29) с. 17–21

Международный журнал информационных технологий и
энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004

ЗАДАЧА НЕПРЕРЫВНОГО ОБНОВЛЕНИЯ В СИСТЕМАХ ПАРСИНГА И АГРЕГАЦИИ ДАННЫХ

Манохин А.К.
МИРЭА - Российский технологический университет, Москва, Россия (119454, г. Москва, пр. Вернадского, 78), e-mail: good12456@mail.ru

В статье рассмотрена актуальность информационных систем парсинга и агрегации данных, а так же рассмотрена задача обновления данных в информационных системах подобного рода без приостановки работы самой системы. Будут предложены два метода решения задачи: метод, основанный на используемом программном обеспечении и архитектурный метод решения задачи.

Ключевые слова: парсинг данных, агрегация данных, веб-приложения; обновление данных, zero downtown deployment, blue green deployment

THE PROBLEM OF UPDATING WITHOUT STOPPING IN DATA PARSING AND AGGREGATION SYSTEMS

Manokhin A. K.
MIREA - Russian Technological University, Moscow, Russia (119454, Moscow, Vernadskogo Ave., 78), e-mail: good12456@mail.ru

The article considers the relevance of information systems of data parsing and aggregation, and also considers the task of updating data in information systems of this kind without suspending the operation of the system itself. Two methods of solving the problem will be proposed: a method based on the software used and an architectural method for solving the problem

Keywords: data parsing, data aggregation, web applications, data update, zero downtown deployment, blue green deployment.

Актуальное положение ИТ сферы и количество данных в мире

Актуальность и рост сферы ИТ в современном обществе не вызывает каких-либо сомнений. Информатизация окружает современного человека со всех сторон и не думает останавливаться. С каждым годом количество пользователей интернета растёт. [1]

Вместе с количеством пользователей растёт и количество генерируемой информации, причём ежегодный прирост количества информации в разы превышает ежегодный прирост пользователей. [2]

Количество ежегодно генерируемой информации по состоянию на 2018 год с прогнозом на последующие годы представлено на графике на Рисунке 1. [3]

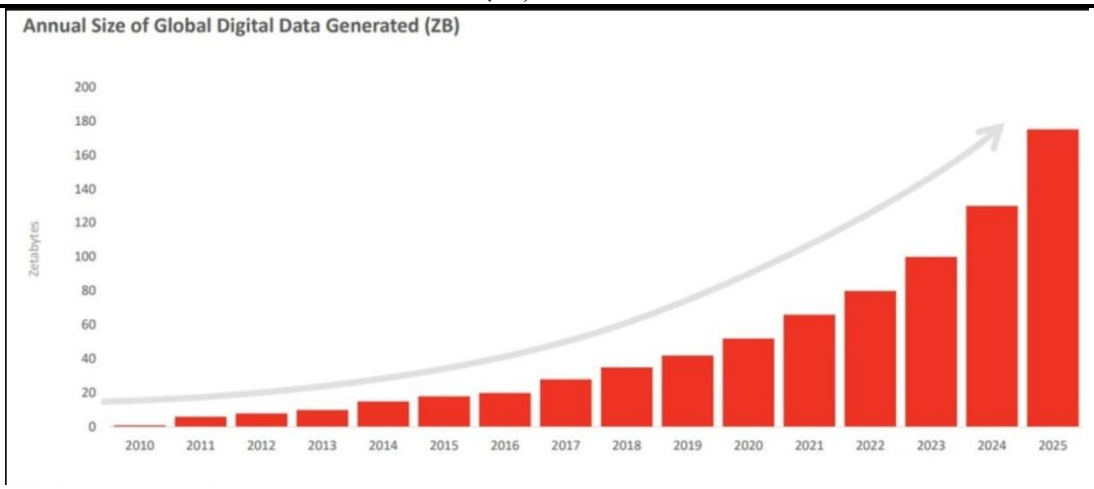

Рисунок 1 – Объём генерируемых цифровых данных в мире.

Методы и цели получения данных с веб-сайтов

Получать информацию от веб-сайтов и веб-приложений можно по сути двумя способами:

1) Воспользоваться публичным API, предоставляемым сайтом.
2) Самостоятельно получить информацию с помощью парсинга.

К сожалению, большинство сайтов с какими-либо данными заточены исключительно на предоставления информации пользователю сайта, но не предоставляют каких-либо программных интерфейсов для взаимодействия с сайтом и предоставляемыми им данными. При этом агрегация информации с различных однотипных сайтов имеет практический смысл для пользователя, поскольку позволяет наглядно сравнить все актуальные предложения из различных источников, и выбирать самый подходящий для пользователя вариант. Причём агрегировать можно фактически любой товар, представленный к продаже несколькими конкурирующими фирмами, начиная с узконаправленных отраслей и заканчивая агрегацией цен на товары первой необходимости в ближайших к пользователю магазинах.

Из-за отсутствия на большинстве сайтов, предлагающих услуги публичного API, получение информации с таких сайтов остаётся возможным только с помощью парсинга данных, а значит задача разработки приложений для парсинга данных является актуальной.

Задача обновления информации в связанных с парсингом данных приложениях.

В рамках систем парсинга данных с веб-сайтов архитектура приложения будет довольно проста. Необходимо хранить информацию, соответственно необходимо хранилище данных или база данных (БД). Необходим модуль получения данных, который будет работать с веб-сайтами, получать информацию и сохранять её, а так же необходим модуль для предоставления данных пользователю по запросу.

Концепция кажется довольно простой на первый взгляд, однако при повторном использовании системы с целью актуализации данных поднимается вопрос – что делать со старыми данными, которые уже потеряли актуальность?

Самый простой выход – удалить все неактуальные данные перед началом сбора новых данных, чтобы можно было заполнить БД с нуля без дублирующихся и устаревших данных.

Однако на заполнение БД нужно время, а запрос от пользователя может прийти во временной промежуток, когда старые данные уже удалены, а новые ещё не собраны в полном объёме. Из-за чего пользователь получит неполную и, следовательно, неверную информацию.

Возможно просто отключать доступ к сервису на время обновления данных, и предупреждать пользователя о временной неработоспособности, с просьбой обратиться позже. Рассмотрим решение задачи обновления данных без остановки работы информационной системы веб-парсинга и агрегации данных двумя различными методами: с использованием особенностей программного обеспечения и с использованием подхода, заложенного в архитектуре информационной системы.

Метод, опирающийся на программное обеспечение.

Поставленную задачу можно решить воспользовавшись внутренними функциями конкретных систем управления базами данных (СУБД), а конкретнее – механизмом транзакций, предусмотренным некоторыми СУБД.

Транзакция – это набор операций с БД, который выполняется либо полностью, либо не выполняется вообще.[4] Начав транзакцию, можно провести несколько операций внутри транзакции, которые никак не повлияют на базу данных, пока не будет подтверждено завершение транзакции. После подтверждения завершения транзакции изменения, совершённые внутри транзакции, будут зафиксированы в самой БД.

Этим можно воспользоваться для обновления данных в системе.

Открыв транзакцию, можно в рамках транзакции отправить запрос на удаление всех существующих строк в таблицах, куда мы хотим записать новые данные. Пока транзакция не будет закрыта, удаление строк будет видно только внутри транзакции и не затронет БД, а приложение сможет обращаться к данным, которые были удалены внутри транзакции. Далее в рамках транзакции мы сможем заполнить таблицы актуальной информацией, и, если не будет обнаружено никаких ошибок, подтвердить завершение транзакции и зафиксировать изменения в БД.

Метод, опирающийся на архитектурное решение.

Рассмотрим альтернативный метод, не зависимый от какой-либо программной платформы, а основанный на архитектуре информационной системы. Метод основан на концепции «Zero downtown deployment» - развёртывание приложения с нулевым временем простоя[5] и методе «blue green deployment» - синее-зелёное развёртывание[5,6].

Метод «blue green deployment» предлагает при разработке новой версии приложения воспользоваться дубликатом имеющейся архитектуры и, при готовности к развёртыванию, просто перенести пользовательские запросы со старой копии архитектуры на новую. Предполагается наличие второго веб-сервера с программным обеспечением, и второго сервера СУБД (Рисунок 2)[6].

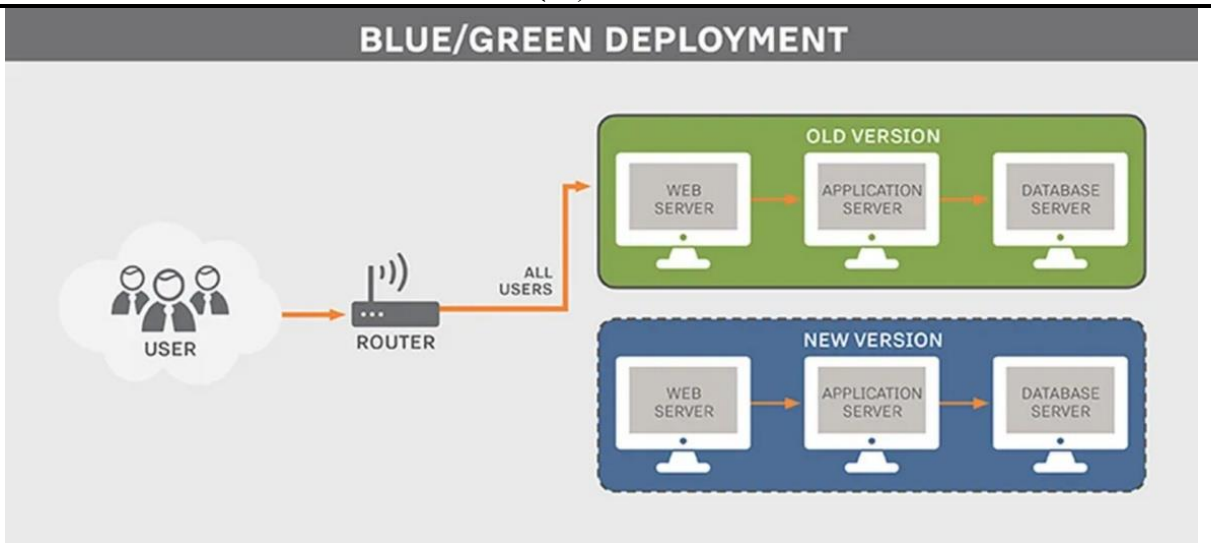

Рисунок 2 – Blue green deployment

На основе метода «blue green deployment» можно разработать архитектурное решение для задачи непрерывного обновления данных в системах веб-парсинга и агрегации данных. В типичную клиент-серверную архитектуру добавим дополнительную базу данных, а в модуль взаимодействия с данными из СУБД программную прослойку, которая будет перенаправлять потоки данных в нужную БД. Архитектура представлена на Рисунке 3.

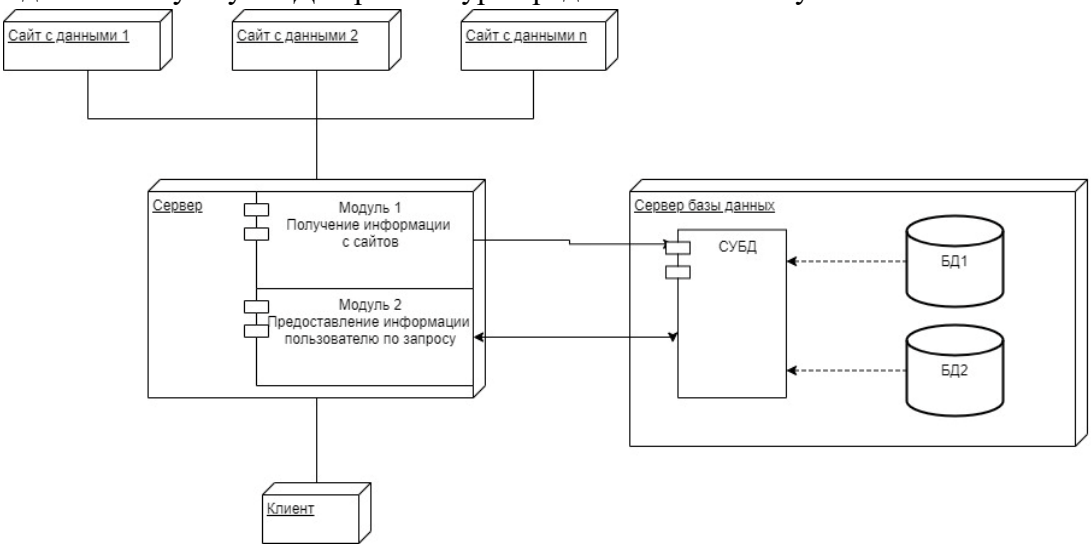

Рисунок 3 – Архитектура с двумя БД

Аналогично методу «blue green deployment» данные при работе будут сначала загружаться в БД1. В следующий раз, при необходимости получить свежие данные, программный модуль должен очистить БД2 и затем сохранить туда новую информацию. Во время процедуры получения данных пользователь всё ещё будет получать информацию из БД1, а как только заполнение БД2 свежей информацией будет успешно завершено необходимо переключить базу данных для предоставление данных пользователю с БД1 на БД2.

Список литературы

1. DIGITAL 2022: ANOTHER YEAR OF BUMPER GROWTH [Электронный ресурс] – URL: https://wearesocial.com/uk/blog/2022/01/digital-2022-another-year-of-bumper-growth-2/ (Дата обращения: 23.03.2023)
2. Мировой объем данных в 2020 году составит 59 зеттабайт [Электронный ресурс] – URL: https://mcs.mail.ru/blog/mirovoj-obem-dannyh-v-2020-godu-sostavit-59-zettabajt (Дата обращения: 23.03.2023)
3. Объём генерируемых цифровых данных в мире [Электронный ресурс] – URL: tadviser.ru/index.php/Статья:Данные (Дата обращения: 23.03.2023)
4. Транзакция (информатика) [Электронный ресурс] – URL: ru.wikipedia.org/wiki/Транзакция_(информатика) (Дата обращения: 23.03.2023)
5. Zero Downtime Deployment и базы данных [Электронный ресурс] – URL: https://habr.com/ru/company/nixys/blog/481932/ (Дата обращения: 23.03.2023)
6. Blue - Green Deployment with Jenkins [Электронный ресурс] – URL: https://medium.com/arabamlabs/blue-green-deployment-with-jenkins-98393bba2327 (Дата обращения: 23.03.2023)

References

1. DIGITAL 2022: ANOTHER YEAR OF BUMPER GROWTH [Electronic resource] – URL: https://wearesocial.com/uk/blog/2022/01/digital-2022-another-year-of-bumper-growth-2/ (Retrieved: 03/23/2023)
2. The global volume of data in 2020 will amount to 59 zettabytes [Electronic resource] - URL: https://mcs.mail.ru/blog/mirovoj-obem-dannyh-v-2020-godu-sostavit-59-zettabajt (Date of access : 23.03.2023)
3. The volume of generated digital data in the world [Electronic resource] - URL: tadviser.ru/index.php/Article: Data (Date of access: 03/23/2023)
4. Transaction (computer science) [Electronic resource] - URL: ru.wikipedia.org/wiki/Transaction_(computer science) (Date of access: 03/23/2023)
5. Zero Downtime Deployment and databases [Electronic resource] - URL: https://habr.com/ru/company/nixys/blog/481932/ (Date of access: 03/23/2023)
6. Blue - Green Deployment with Jenkins [Electronic resource] - URL: https://medium.com/arabamlabs/blue-green-deployment-with-jenkins-98393bba2327 (Date of access: 03/23/2023)