А. Д. Евстратенко

ПОТОКОВАЯ ОБРАБОТКА ТРАНЗАКЦИОННЫХ ДАННЫХ

Рассматривается архитектура и реализация программного модуля потоковой обработки транзакционных данных, основанного на механизме логической репликации PostgreSQL. Модуль осуществляет захват изменений из журнала предзаписи, фильтрацию событий по настраиваемым критериям и доставку данных во внешние системы – поисковые движки и брокеры сообщений.

Введение

В современных распределённых системах актуальной задачей является синхронизация данных между основной базой данных и вспомогательными хранилищами – поисковыми индексами, кэшами, очередями сообщений. Традиционный подход, основанный на двойной записи, не гарантирует согласованности данных при сбоях. Альтернативой является паттерн захвата изменений данных, при котором изменения извлекаются непосредственно из журнала транзакций базы данных [1].

В данной работе представлен программный модуль, реализующий потоковую обработку транзакционных данных PostgreSQL с доставкой событий в несколько целевых систем.

I. Архитектура модуля

Модуль реализован на языке Go и использует протокол логической репликации PostgreSQL с плагином pgoutput. Архитектура построена по принципу конвейерной обработки данных и включает четыре основных компонента (см. рис. 1):

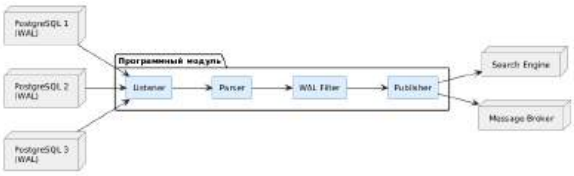

Рис. 1 – Архитектура программного модуля потоковой обработки транзакционных данных

Listener – устанавливает соединение с PostgreSQL по протоколу логической репликаци и принимает поток бинарных сообщений WAL. Реализован механизм периодического обновления соединения и отправки heartbeat-сообщений для поддержания связи [2].

Parser – декодирует бинарные сообщения протокола логической репликации в структурированные объекты: Begin, Relation, Insert, Update, Delete, Commit. Каждый тип сообщения содержит метаданные транзакции и изменённые данные.

WAL Filter – накапливает изменения в рамках транзакции и фильтрует события по конфигурируемым критериям: таблице, набору полей и типу операции.

Publisher – доставляет отфильтрованные события в целевую систему. Реализован паттерн «фабрика»: единый интерфейс позволяет подключать различные системы-потребители без изменения остальных компонентов конвейера.

II. Обеспечение надёжности

Надёжность доставки событий обеспечивается отслеживанием позиции LSN (Log Sequence Number) в журнале предзаписи. Модуль подтверждает обработку каждой транзакции, что позволяет PostgreSQL корректно управлять слотом репликации и обеспечивает возможность повторной доставки при сбоях.

Модуль поддерживает параллельную обработку нескольких баз данных – каждая получает отдельный экземпляр слушателя, парсера и издателя, выполняемый в отдельной горутине. Конфигурация задаётся в формате YAML с возможностью переопределения параметров через переменные окружения.

III. Выводы

Разработанный модуль решает задачу потоковой синхронизации данных между PostgreSQL и внешними системами, используя механизм логической репликации. Конвейерная архитектура и паттерн «фабрика» обеспечивают расширяемость: добавление нового целевого хранилища сводится к реализации единственного интерфейса. Отслеживание LSN гарантирует надёжность доставки событий без потери данных.

Список литературы

1. Kleppmann, M. Designing Data-Intensive Applications / M. Kleppmann. – O'Reilly Media, 2017. – 616 p.
2. Juba, S. Learning PostgreSQL / S. Juba, A. Volkov. – Packt Publishing, 2017. – 428 p.

Евстратенко Артём Дмитриевич, студент кафедры информационных технологий автоматизированных систем БГУИР, art.evstr@gmail.com.

Ярмолик Валерий Иванович, старший преподаватель информационных технологий автоматизированных систем БГУИР, v.jarmolik@bsuir.by