

4.5 Информационные системы и технологии

УДК 004.042

Apache Kafka КАК ИНСТРУМЕНТ ПОТОКОВОЙ ОБРАБОТКИ ДАННЫХ В РЕАЛЬНОМ ВРЕМЕНИ

Бондарев Е. А., студ., Бизюк А. Н., ст. преп.

*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. Apache Kafka представляет собой распределённую систему обмена сообщениями, предназначенную для обработки, хранения и передачи потоков данных в реальном времени. Данная система широко применяется в промышленности, особенно в архитектурах с микросервисами, Big Data и высоконагруженных приложениях. В докладе рассматриваются архитектура Kafka, её ключевые компоненты, преимущества и типовые сценарии применения.

Ключевые слова: Apache Kafka, потоковые данные, брокер сообщений, микросервисы, реальное время, масштабируемость, отказоустойчивость.

С ростом объёма генерируемых данных во всех отраслях – от финансовых сервисов до умных устройств – возрастает потребность в надёжных и масштабируемых системах, способных обрабатывать потоки информации в реальном времени. Одним из ведущих решений в этой области является Apache Kafka – распределённая платформа потоковой передачи данных, созданная для высокопроизводительной публикации, хранения и обработки сообщений. На сегодняшний день Kafka используется более чем в 80 % компаний из списка Fortune 100, что подтверждает её зрелость и востребованность. Платформа активно применяется в разнообразных сценариях – от логирования и мониторинга до построения архитектуры микросервисов и потоковой аналитики. Изначально Kafka была разработана в компании LinkedIn для решения внутренних задач, связанных с обработкой больших объёмов логов, а с 2011 года развивается как проект с открытым исходным кодом под эгидой Apache Software Foundation.

Apache Kafka реализует архитектурную модель pub-sub (publish-subscribe) – публикации и подписки – при которой данные публикуются источниками и затем распределяются среди потребителей (рис. 1).

Основными компонентами системы являются:

- Продюсер (producer) – приложение или сервис, публикующий данные (сообщения) в Kafka; может быть частью веб-приложения, IoT-устройства или другой системы.
- Брокер (broker) – сервер Kafka, который принимает сообщения от продюсеров, хранит их и предоставляет потребителям по запросу. Кластер Kafka может включать множество брокеров, обеспечивая масштабируемость и отказоустойчивость.
- Потребитель (consumer) – приложение, подписанное на определённые топики и извлекающее сообщения для дальнейшей обработки.

– Топик (topic) – логическая структура для хранения сообщений, в рамках которой сообщения группируются по смыслу.

– Раздел (partition) – подструктура внутри топика, обеспечивающая параллельную обработку данных и масштабируемость по горизонтали.

– ZooKeeper – система координации, применяемая в ранних версиях Kafka для

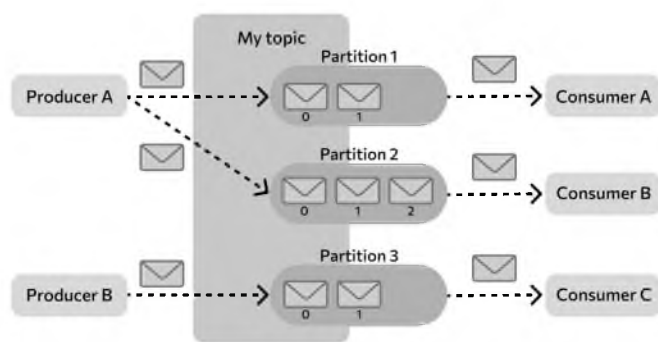


Рисунок 1 – Схема работы Kafka Apache

управления кластерами. В последних релизах Kafka переходит на собственный протокол KRaft, позволяющий отказаться от ZooKeeper.

Сообщения в Kafka организованы в виде неизменяемого журнала, где записи упорядочены по времени. После публикации они сохраняются на определённое время (настраивается) и не могут быть изменены или удалены, что обеспечивает надёжность и аудит операций.

Для коммерческого использования разработчики Kafka создали Confluent Platform – расширение с улучшенной производительностью, поддержкой, готовыми коннекторами, инструментами мониторинга и развертыванием в облаке или локально. Это позволяет предприятиям быстро запускать решения на базе Kafka и адаптировать их под свои задачи.

Kafka успешно применяется в различных архитектурных и бизнес-сценариях, включая:

- Связь микросервисов. Kafka используется как высоконадёжный посредник между микросервисами, обеспечивая их асинхронное взаимодействие и независимость.
- Поточковая аналитика. В связке с фреймворками Kafka Streams, Apache Flink или Druid, Kafka позволяет обрабатывать данные в реальном времени, выявлять аномалии, строить отчёты и применять машинное обучение к текущим событиям.
- ETL (Extract, Transform, Load – дословно «извлечение, преобразование, загрузка») в реальном времени. С помощью Kafka Connect можно настраивать потоки данных между системами – базами данных, API, файлами и другими источниками, обеспечивая синхронизацию и трансформацию данных в реальном времени.
- Логирование и аудит. Kafka широко используется для сбора, хранения и анализа логов, событий, транзакций, а также для построения централизованных систем мониторинга, обеспечивая масштабируемость и надёжность хранения.

В проекте библиотеки Apache Kafka играет центральную роль в обеспечении асинхронного взаимодействия между сервисами. Kafka используется для передачи сообщений, связанных с событиями, происходящими в библиотечной системе. Это позволяет системе быть более масштабируемой и независимой, так как сервисы могут обмениваться данными без необходимости прямой связи.

Kafka интегрирована с помощью библиотеки Spring Kafka, которая предоставляет удобные инструменты для настройки и управления взаимодействием с брокерами Kafka. В каждом из микросервисов проекта определены конфигурационные классы (KafkaConfig), где задаются параметры подключения.

Продюсеры в проекте отвечают за генерацию и отправку сообщений в Kafka. Эта функциональность реализована в основном в сервисе book-storage-service, который отвечает за управление данными о книгах.

- При создании новой книги формируется сообщение с её идентификатором и статусом, которое отправляется в топик book-creation.
- При удалении книги соответствующее сообщение отправляется в топик book-deletion.

Такая структура позволяет отделить процесс создания или удаления книги от дальнейшей обработки этих операций другими сервисами. Продюсеры гарантируют, что события систематически фиксируются и становятся доступными для подписчиков.

Консумеры, в свою очередь, обрабатывают сообщения, поступающие из Kafka, и выполняют действия на их основе. В проекте консумеры реализованы в сервисе book-tracker-service.

- При получении сообщения из топика book-creation создаётся запись об отслеживании книги, где фиксируется её ID и статус.
- При получении сообщения из топика book-deletion обновляется статус книги как «удалено», а соответствующая запись помечается как неактивная.

Использование консумеров позволяет обрабатывать события независимо от их источника, что минимизирует задержки и снижает нагрузку на сервисы, генерирующие события.

Kafka в данном проекте выполняет несколько ключевых функций:

- Асинхронность: сервисы могут обрабатывать события независимо друг от друга, что повышает их отказоустойчивость и масштабируемость.
- Связь между сервисами: Kafka служит посредником, обеспечивая унифицированный обмен данными между микросервисами.
- Обработка событий в реальном времени: сервисы оперативно реагируют на изменения, такие как создание или удаление книг, за счёт мгновенного получения сообщений.

Apache Kafka зарекомендовала себя как надёжная, производительная и гибкая система для потоковой передачи и обработки данных в реальном времени. Её архитектура ориентирована на отказоустойчивость, масштабируемость и минимальные задержки, что делает её незаменимой

для построения современных IT-систем, работающих с большими объемами информации. Благодаря активному сообществу, широкой экосистеме и поддержке коммерческих решений, Kafka по праву считается стандартом де-факто в мире потоковой обработки данных и событийно-ориентированной архитектуры. В условиях стремительного роста требований к цифровым системам Kafka продолжает развиваться и укреплять свои позиции как один из ключевых инструментов в арсенале инженеров данных, разработчиков и архитекторов.

Список использованных источников

1. Apache Kafka: просто о главном / Slurm [Электронный ресурс]. – Витебск, 2025. – Режим доступа: <https://habr.com/ru/companies/slurm/articles/550934/>. – Дата доступа: 13.05.2025.
2. Что такое Kafka и зачем она нужна / Kuper [Электронный ресурс]. – Витебск, 2025. – Режим доступа: <https://habr.com/ru/companies/kuper/articles/738634/>. – Дата доступа: 13.05.2025.
3. Kafka (Apache Kafka) – что это такое / SkillFactory [Электронный ресурс]. – Витебск, 2025. – Режим доступа: <https://blog.skillfactory.ru/glossary/kafka-apache/>. – Дата доступа: 13.05.2025.
4. What is Apache Kafka? / Confluent [Электронный ресурс]. – Витебск, 2025. – Режим доступа: <https://www.confluent.io/what-is-apache-kafka/>. – Дата доступа: 13.05.2025.

УДК 004.6

АНАЛИЗ МЕТОДОВ УПРАВЛЕНИЯ МИГРАЦИЕЙ БАЗ ДАННЫХ НА ОСНОВЕ БИБЛИОТЕК FLYWAY И LIQUIBASE

Мацуганова М. Д., инженер-программист, Деркаченко П. Г., ст. преп.
*Витебский государственный технологический университет,
г. Витебск, Республика Беларусь*

Реферат. В статье рассмотрено сравнение подходов к миграции баз данных с использованием библиотек Flyway и Liquibase, анализируются их основные принципы работы, преимущества и ограничения.

Ключевые слова: Flyway, Liquibase, SQL, ChangeLog, Changeset.

Миграция баз данных играет ключевую роль в разработке программного обеспечения, особенно при создании веб-приложений и других систем, требующих надёжного хранения информации. Грамотное управление процессом переноса данных позволяет сохранить их целостность, повысить производительность и обеспечить масштабируемость приложения. [1].

Миграция описывает последовательность изменений, которые необходимо применить к базе данных. Эти изменения могут включать создание новых таблиц, изменение существующих, добавление или удаление столбцов, создание индексов, изменение ограничений и так далее.

Для эффективной миграции баз данных необходимо ознакомиться со специализированными инструментами, предназначенными для упрощения этого процесса. Два наиболее популярных инструмента в этой области – Flyway и Liquibase.

Flyway – популярный инструмент с открытым исходным кодом, известный своей простотой и удобством использования. Он использует подход к миграции баз данных на основе версий, при котором каждая миграция представляет собой набор SQL-скриптов, которые перемещают базу данных из одной версии в другую [2].

Основная единица работы, обрабатываемая Flyway, называется миграцией. Это любая совокупность операторов SQL, которые вы хотите выполнить за одну операцию. Каждая миграция хранится в отдельном файле SQL и на основе их названий Flyway понимает, при каких условиях и в каком порядке они должны выполняться. Схема миграции с использованием Flyway представлена на рисунке 1.

Ещё одним эффективным способом миграции данных в приложениях является использование Liquibase. Liquibase – это инструмент для миграции баз данных с открытым исходным кодом, который позволяет разработчикам декларативно определять изменения в схеме базы данных и управлять ими. Вместо того чтобы вручную писать SQL-скрипты для изменения структуры