

[Презентация к уроку 9.2.1](#)

Текстовая расшифровка видео:

ПРИМЕРЫ ЗАДАЧ СТРИМИНГА

План:

- Примеры задач стриминга;
- Поточковая обработка данных;
- Семантика доставки;
- Звено очереди сообщений;
- Долговечные сообщения;
- Очереди сообщений;
- Пример системы отправки сообщений.

Примеры задач стриминга

Стриминговые события отличаются тем, что они обрабатываются практически без задержек. Задержки могут составлять несколько миллисекунд или несколько секунд. Такие события можно назвать **обработкой в режиме RealTime**.

RealTime-приложения отправляют данные получателям сразу, в стриминговых же системах получатели запрашивают события по мере необходимости.

В связи с этим часто возникает задача – события необходимо где-то хранить.

К примерам стриминговых систем относятся:

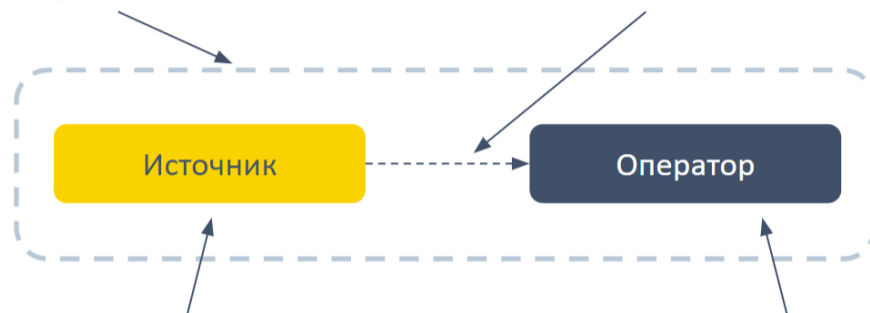
- Отслеживание действий пользователя на сайте (клики, просмотры страниц и т.д.). Эти события впоследствии можно обрабатывать и получать ценную информацию, например, количество юзеров, подозрительное поведение юзеров и т.д.

Потоковая обработка данных

Пять основных компонентов стриминговой обработки:

Задание, также называемое конвейером или топологией, представляет собой реализацию стриминговой системы. Задание строится из компонентов (источников и операторов) и потоков, соединяющих компоненты.

Потоком называется непрерывная поставка событий. Событие, также называемое кортежем, элементом или сообщением в разных сценариях, представляет собой отдельный блок неделимых данных в потоке.



Источник, который вводит внешние данные в потоковую систему. Источники являются точками входа данных в стриминговых системах.

Оператор, также называемый преобразованием, — это часть, которая получает и обрабатывает события. В операторах выполняется вся основная логика.

- Событие;
- Поток событий (обязательная непрерывная поставка событий);
- Источник (откуда вводятся внешние данные в потоковую систему);
- Оператор (место, где события обрабатываются);
- Задание в целом (конвейер/топология, который описывает реализацию стриминговой системы).

Семантика доставки

Концепция семантики доставки (или гарантии доставки) показывает то, как система обрабатывает доставку или обработку событий.

Три основные семантики:

Семантика доставки	Не более одного	Не менее одного	Ровно один
Точность	Точность не гарантирована из-за пропуска событий	Точность не гарантирована из-за дублирования событий	(На первый взгляд) гарантирован точный результат
Задержка (при возникновении ошибки)	Устойчивость к сбоям, отсутствие задержки при возникновении ошибок	Чувствительность к сбоям, возможны задержки при возникновении ошибок	Чувствительность к сбоям, возможны задержки при возникновении ошибок
Сложность	Очень простая	Средняя (в зависимости от реализации)	Высокая

Не более одного. Событие будет отправлено или обработано не более одного раза. Есть вероятность ни одного раза, так как нет достоверности обработанности события.

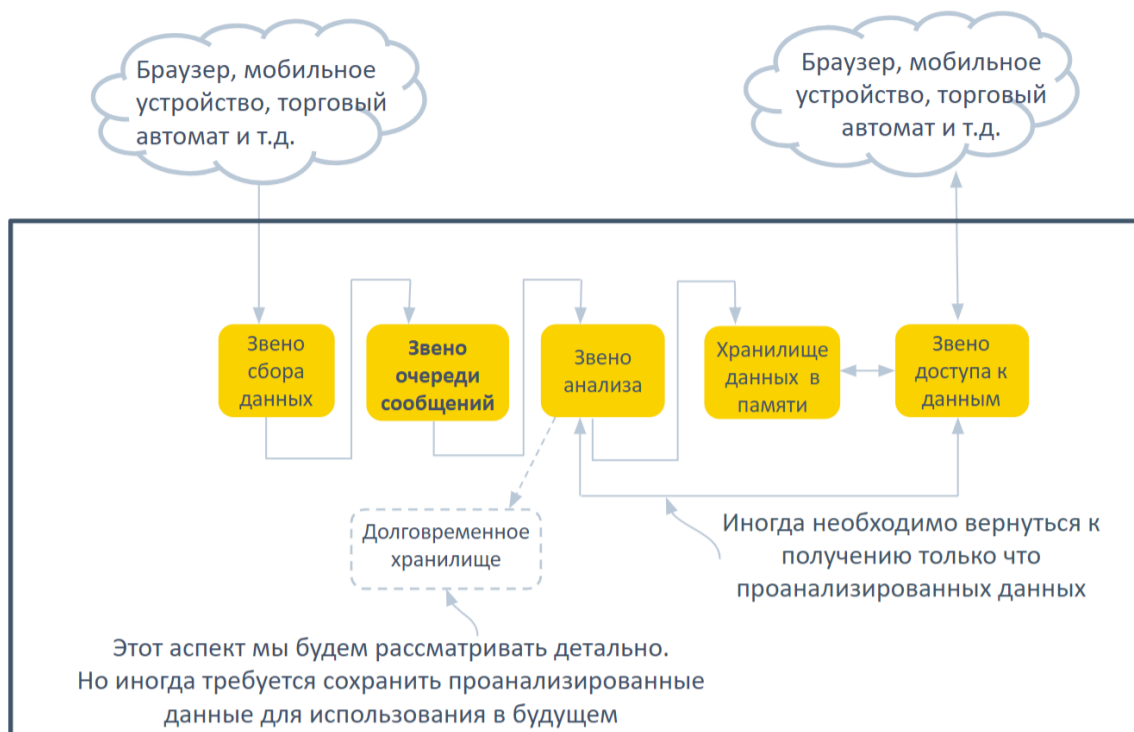
Не менее одного. Событие будет отправляться не менее одного раза, пока не будет успешно обработано или получено. Здесь могут быть дубли.

Ровно один. Событие будет отправляться или обрабатываться один раз до подтверждения успешной обработки и отправки. Это наиболее сложная схема.

Звено очереди сообщений

В случае, если потоковая обработка осуществляется не на одной машине, необходимо поддерживать между собой сеанс разных звеньев обработки сообщений.

Когда мы добавляем некий транспорт в систему стриминга, мы автоматически снимаем с этих звеньев нагрузку, касающуюся маршрутизации, отправки, хранения для отправки и т.д.:



Долговечные сообщения

Например, если данные производятся быстрее, чем они получают на потребителе, нам необходимо долговременное хранилище для сообщений.

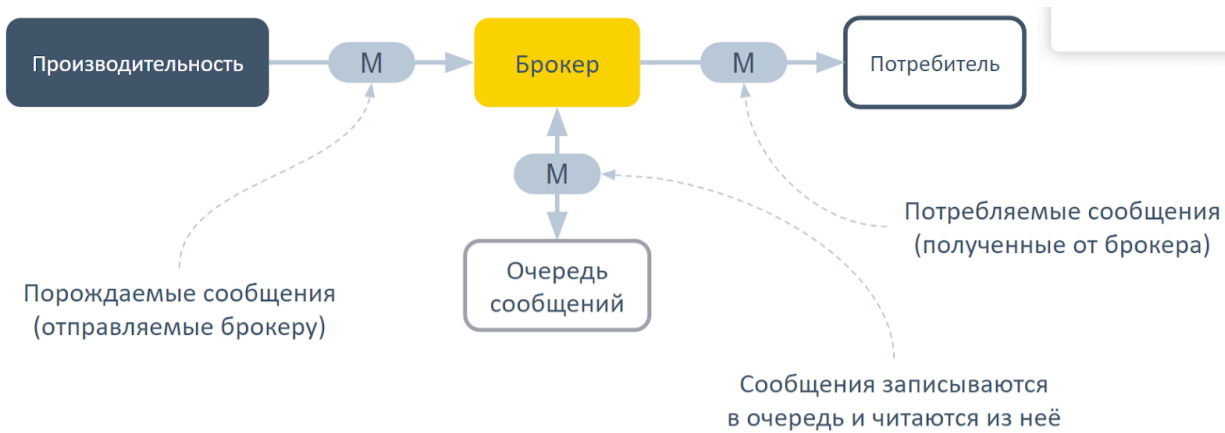
Если критичны потери сообщений за несколько дней (их нужно собирать, даже если они не собирались несколько дней), нам необходимо долговременное хранилище. В данном случае долговременным хранилищем может стать какая-нибудь база, как в примере:



- Файловая система;
- JDBC (Java connector);
- Коммерческое решение в облаке.

Очереди сообщений

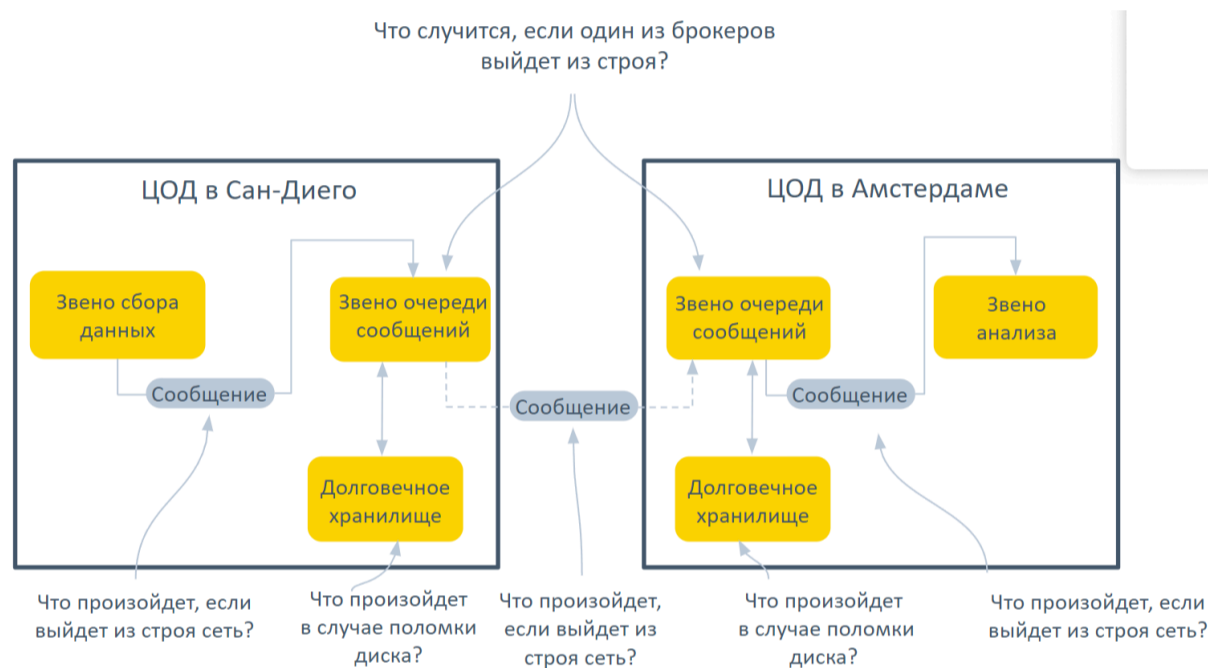
В архитектуре очередей существуют следующие компоненты:



- Производитель (продюсер);
- Брокер (управляет очередями);
- Потребитель (консьюмер).

Брокер называется так, поскольку может управлять сразу несколькими очередями. Задача брокера заключается в загрузке сообщений в очередь.

Пример системы отправки сообщений



При разработке архитектуры Detail-пайплайна и взаимодействий между микросервисами необходимо ответить на четыре вопроса:

- За сколько дней нужно хранить данные и нужно ли их хранить вообще?
- Насколько критично отсутствие связности между звеном сбора данных и анализа данных для бизнеса?
- Что делать, если у нас утеряны данные за несколько дней?
- Какая семантика сообщений нужна для доставки? Предполагает ли система обработку повторных сообщений или они будут влиять на статистику? Можно ли пропускать какие-то данные или же нам нужно будет обязательно доставлять данные ровно один раз?

Как вам урок?



Изучил, далее >