Очередь сообщений Kafka. Spark Streaming

LSML #3

Что сегодня изучим?

- Очереди сообщений
- Apache Kafka
- Подход – micro batch streaming
- Spark streaming
- Примеры реализации стримингового ETL

Шина данных Kafka

Очередь сообщений

Решение для передачи, временного хранения и обработки данных в режиме реального времени. Часто используются для распределения данных между различными приложениями или компонентами системы. Широко используется для стриминговой обработки данных, создания централизованных журналов событий и реализации архитектур "событие-ориентированных" систем

Очередь сообщений

Модели очередей

PULL

Инициатива запроса данных: получатель запрашивает данные у отправителя по мере необходимости. Получатель активно инициирует запрос данных.

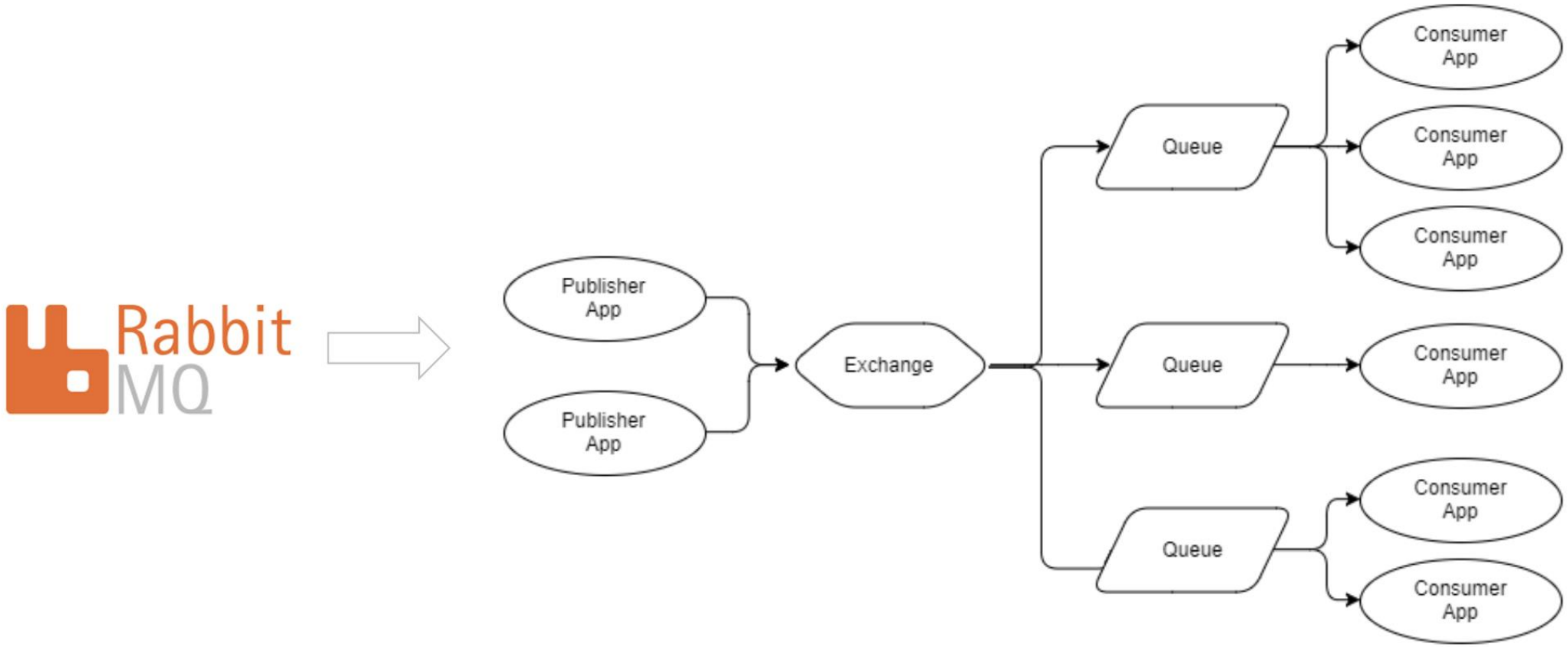
Примеры: запросы к базе данных, загрузка вебстраниц, клиент-серверное взаимодействие, где клиент запрашивает данные у сервера.

PUSH

Инициатива передачи данных: данные передаются от отправителя к получателю без запроса со стороны получателя.

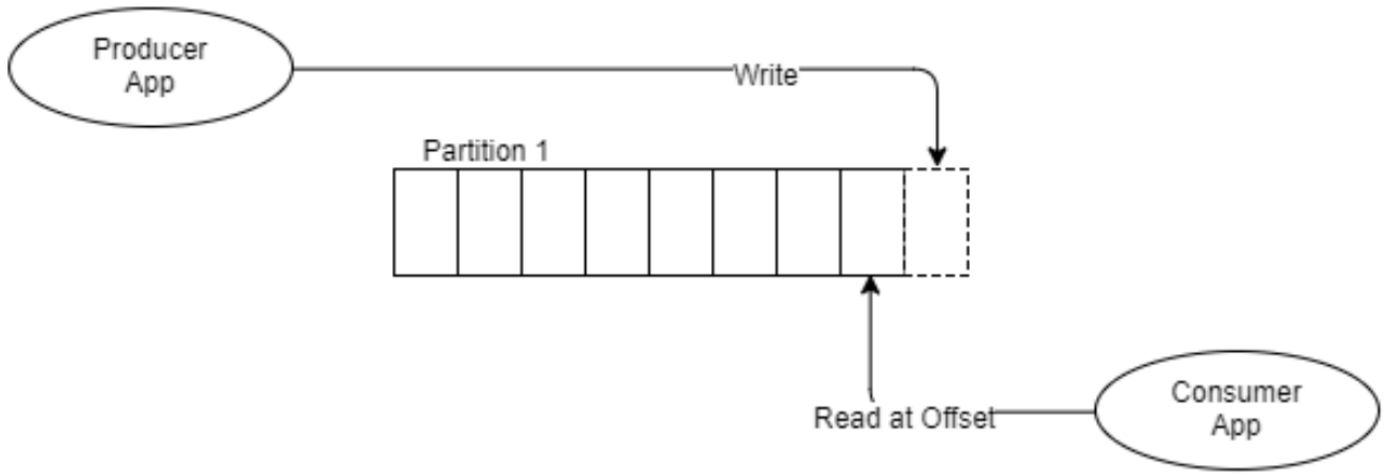
Примеры: уведомления на мобильных устройствах, рассылка электронной почты, стриминговые сервисы, где данные активно отправляются клиенту

PUSH-модель


Rabbit
MQ
Publisher App
Publisher App
Exchange
Queue
Queue
Queue
Consumer App
Consumer App
Consumer App
Consumer App
Consumer App
Consumer App

PULL-модель

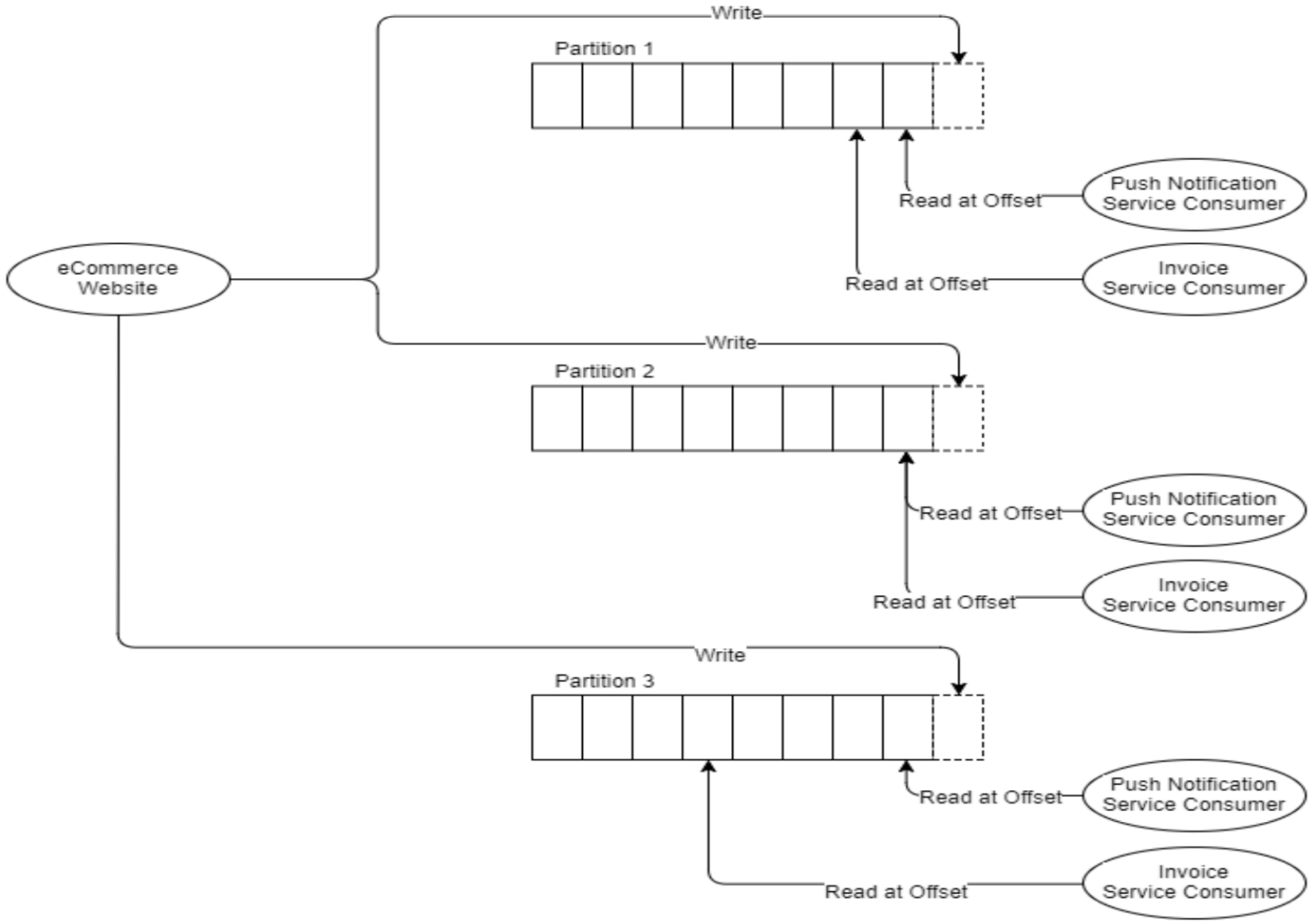
kafka



Определение Apache Kafka

Распределенная платформа, которая предоставляет высокопроизводительное, надежное и масштабируемое решение для передачи, хранения и обработки данных в реальном времени. Шина данных Kafka позволяет управлять потоками данных с использованием тем (topics) для организации информации и разделения ее между различными приложениями или компонентами системы

Принцип работы Kafka

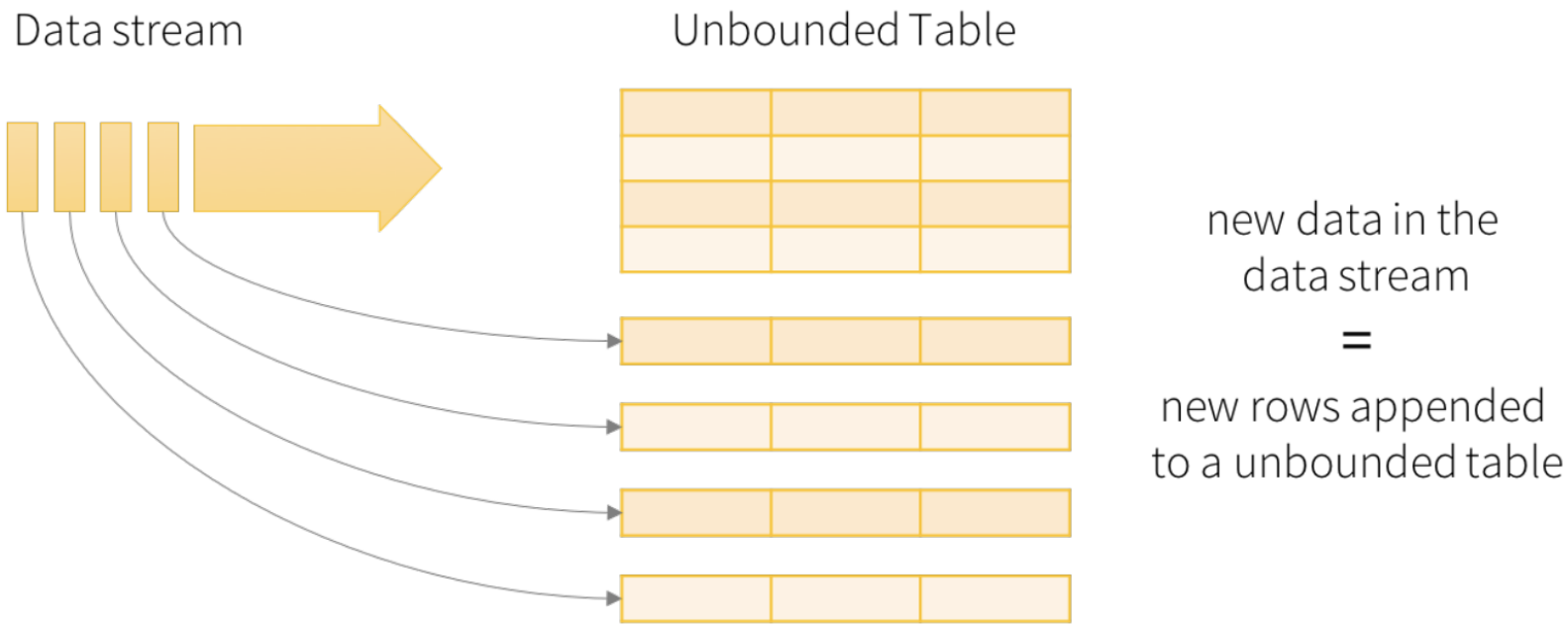


Spark Streaming

Micro batch streaming

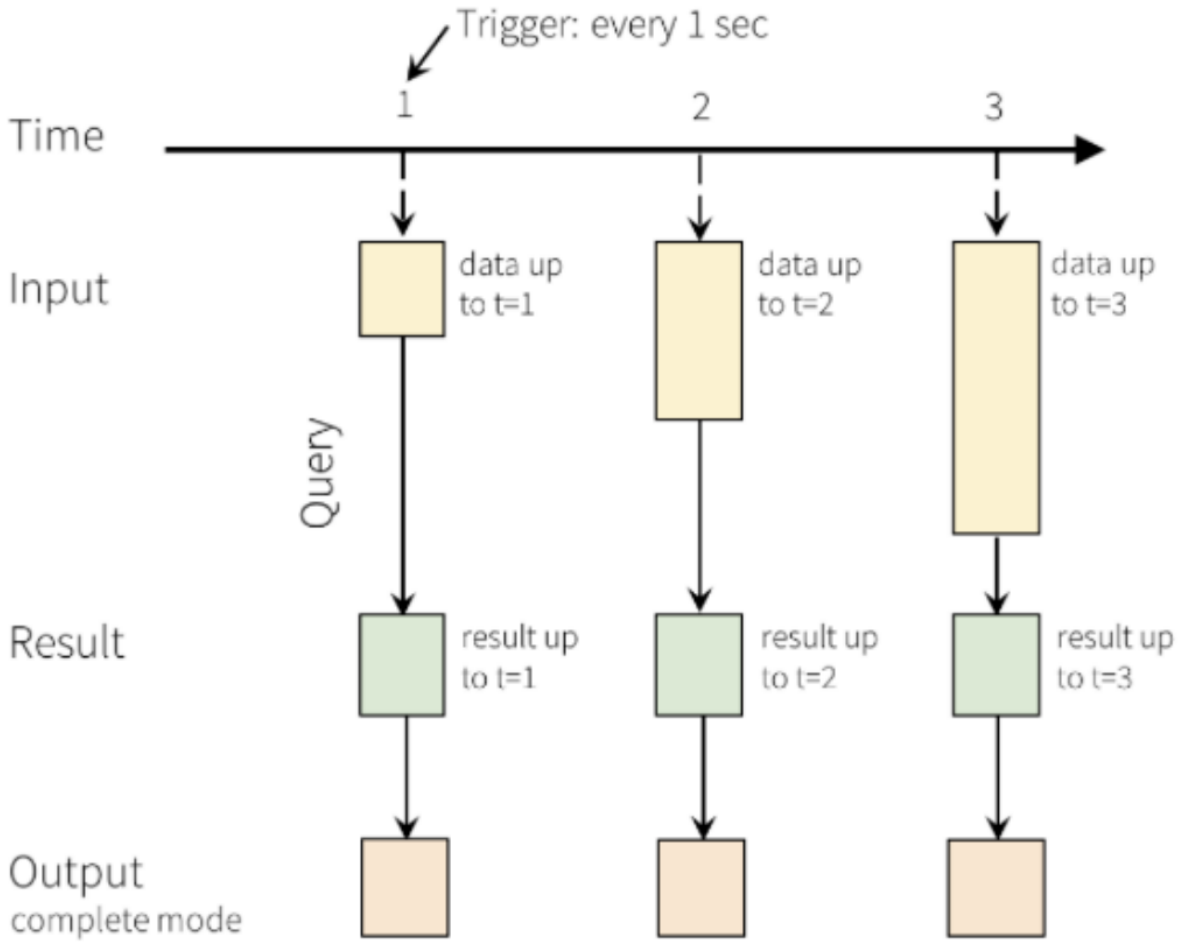
Подход к обработке потоковых данных, при котором данные обрабатываются пакетами фиксированного размера, называемыми micro-batches. Является компромиссом между традиционной пакетной обработкой данных и строгим потоковым (streaming) подходом.

Концепция Spark Streaming

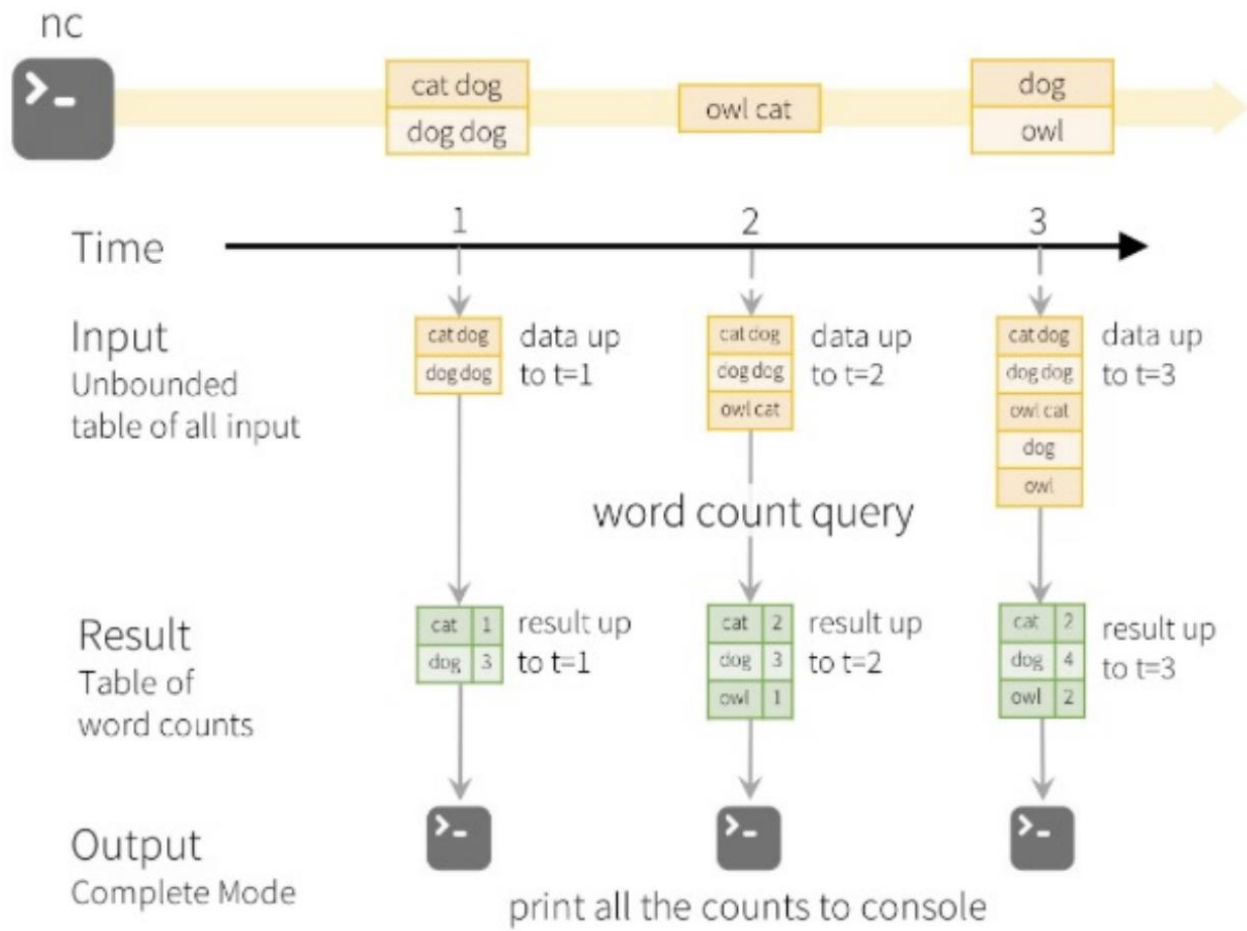


Data stream as an unbounded table

Концепция Spark Streaming - II



Интуитивный пример



Основные возможности

Достоинства

Единый API для пакетной и потоковой обработки

Толерантность к сбоям и обработка точно один раз

Обработка только 1 раз

Интеграция со стандартными источниками данных

Интеграция с экосистемой Spark/Hadoop

Поддержка SQLзапросов

Недостатки

Задержка (Latency)

Сложность масштабирования (обработка огромных micro-batch)

Меньше строк кода, чем SQL

Сложность управления состоянием

Необходимость знания экосистемы Spark

Ограничения по поддержке источников данных