OTUS

Онлайн образование

otus.ru

Проверить, идет ли запись

Меня хорошо видно && слышно?

Тема вебинара

Kafka Streams

Непомнящий Евгений

Разработчик Java/Kotlin IT-Sense

@evgeniyN

Правила вебинара

Активно
участвуем

Off-topic обсуждаем
в TG

Задаем вопрос
в чат или **голосом**

Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения

Индивидуально

Время, необходимое
на активность

Пишем в чат

Говорим голосом

Документ

Ответьте себе или
задайте вопрос

Маршрут вебинара

- GlobalKTable
- ProcessorAPI
- Работа без БД

Цели вебинара

После занятия вы сможете

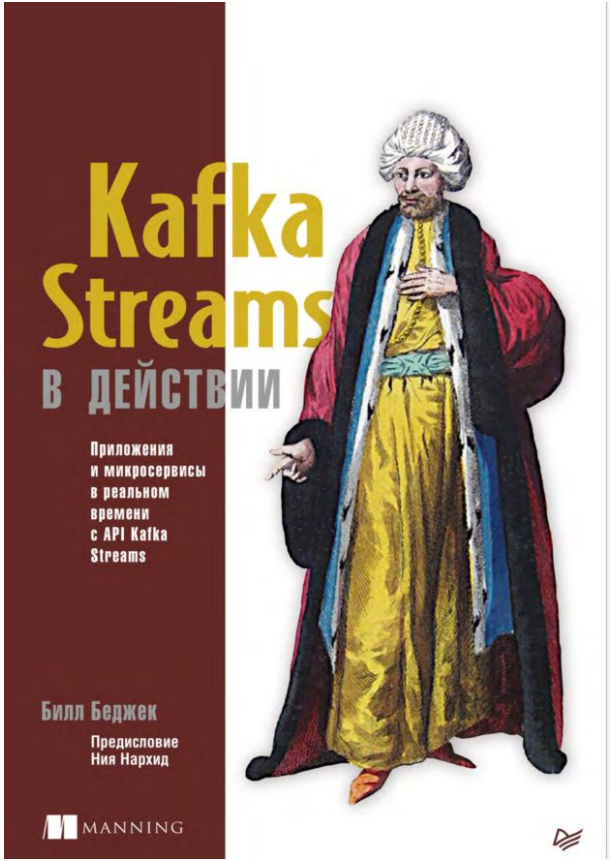
1. Использовать GlobalKTable
2. Использовать Processor API
3. Работать с данными от стримов без дополнительной БД

Смысл

Зачем вам это уметь

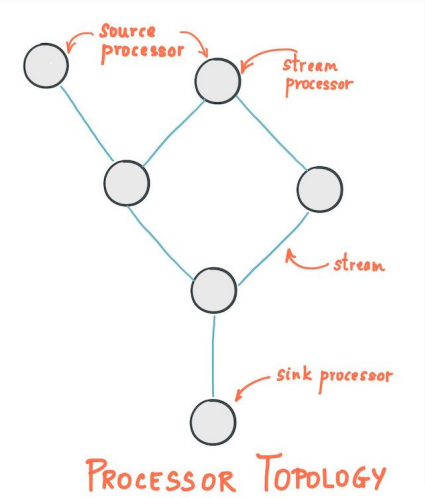
1. Kafka Streams позволяет избавиться от ручной работы в некоторых ситуациях

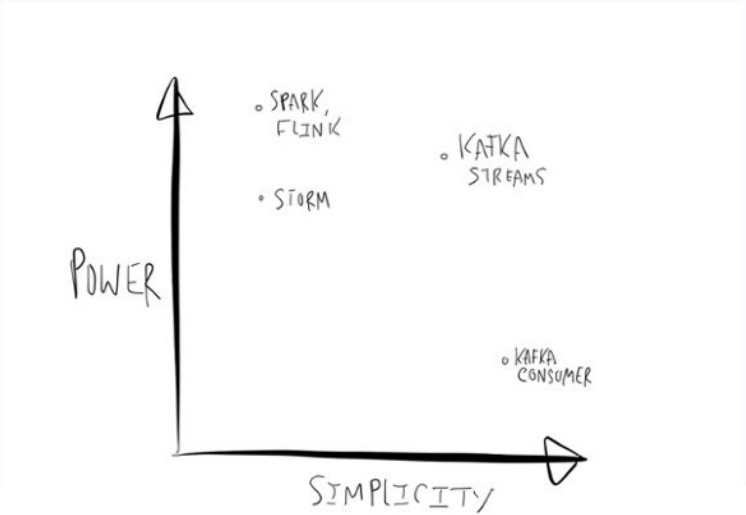
Литература


Kafka Streams
в действии
Приложения и микросервисы в реальном времени с API Kafka Streams
Билл Беджек
Предисловие Ния Нархид
MANNING

Kafka Streams

Kafka Streams - это про потоковую обработку событий. Вы получаете событие из одного или нескольких топиков, что-то с ними делаете и кладете в какие-то другие топики.
Kafka Streams позволяет работать не только с цепочкой обработчиков, а с DAG (направленный ациклический граф)





Подготовка

Задание 1

Запустите kafka

Выполните команду

docker compose up -d (в папке lesson-11/kafka, где лежит docker-compose.yml)

первый запуск может занимать много времени.

Запустите приложение

git clone https://github.com/OtusTeam/OTUS-Kafka.git

Далее зайдите в папку lesson-11 и запустите gradlew build - первый запуск может занимать много времени.

Остатки
с прошлого занятия

FixedSizedQueue serde

Ex8Aggregation

Статистика по окну

Допустим мы хотим посчитать кол-во операций для трейдера+тикета относительно окна

Ex9Window

GlobalKTable

Пример

Есть поток данных

`Industry: TransactionSummary (customerId, stockTicker, industry, summaryCount)`

Есть топики с информацией о

- компаниях (`stockTiker: name`)
- клиентах (`customerId: name`)

Мы хотим получить исходный поток, но с заполненными customerName и companyName.

Ex10GlobalKTable1 - доработайте код в **1**, **2** и **3**

Сроки выполнения: 5 мин

Давайте присмотримся

selectKey приводит к записи в топик и последующему чтению из него (репартиционирование).
У нас происходит три раза.

Это требуется делать, потому что данные в таблицах каждый таск получает только из одной партиции. И данные по транзакциям должны соответствовать данным по компаниям и клиентам.

Однако данных по компаниям и клиентам мало. Если бы мы прочитали вообще все партиции и имели бы все данные по компаниям и клиентам, нам бы не пришлось делать репартиционирования.

Ex10GlobalKTable2 - доработайте код в **1**, **2**

Сроки выполнения: 5 мин

GlobalKTable

Берет все данные из всех партиций топика и трактует их как “таблицу”.
Можно использовать для объединений и ручного чтения данных, более никаких операций нет.

Join

Левое соединение	Внутреннее соединение	Внешнее соединение
KStream-KStream	KStream-KStream	KStream-KStream
KStream-KTable	KStream-KTable	–
KTable-KTable	KTable-KTable	KTable-KTable
KStream-GlobalKTable	KStream-GlobalKTable	–

Processor API

Processor API

До сих пор мы использовали dsl для создания топологии - StreamBuilder … build().
Однако можно формировать DAG вручную напрямую через топологию

Ex11UpperCaseTransformerProcessor

Processor API

Можно совмещать с dsl - process, processValues

Ex4Reward

Punctuate

context.schedule(Duration, Type, Punctuate)

https://cwiki.apache.org/confluence/display/KAFKA/Punctuate+Use+Cases

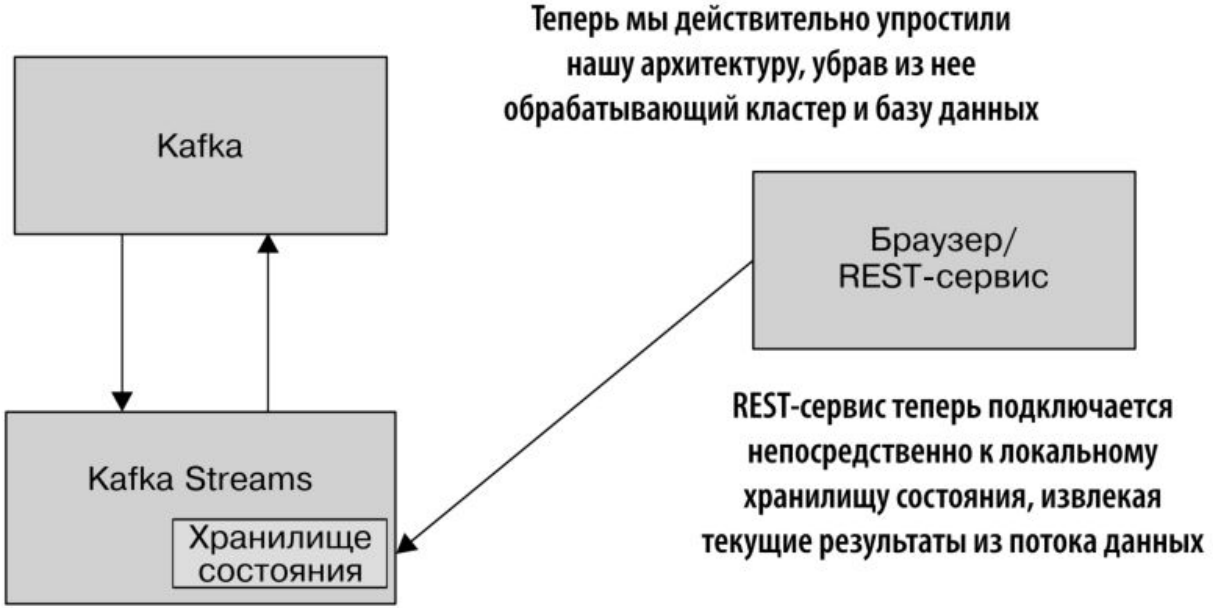
Ex12Punctuate

Запросы к локальным хранилищам состояния

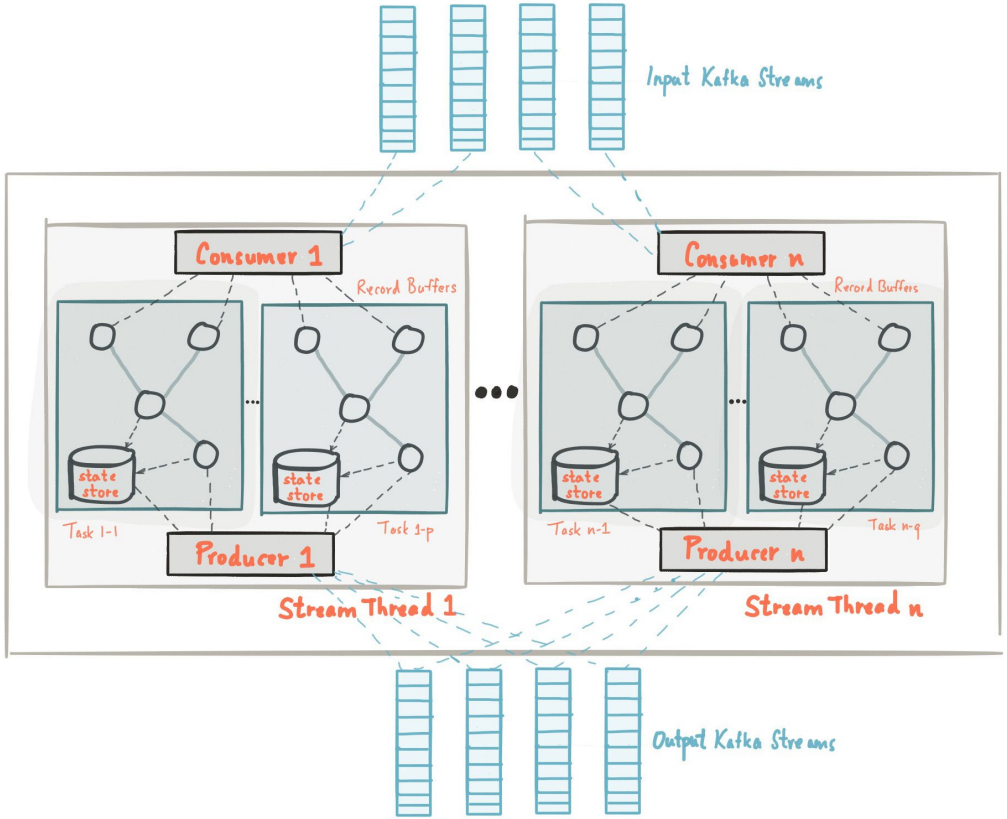
Состояние


Kafka
Браузер/
REST-сервис
Внешнее приложение
потребляет результаты
обработки и записывает их
в базу данных
Внешнее
приложение
Kafka Streams
Запуск Kafka Streams все упрощает,
поскольку отпадает потребность
в отдельном кластере
База
данных

Состояние


Теперь мы действительно упростили
нашу архитектуру, убрав из нее
обрабатывающий кластер и базу данных
Kafka
Браузер/
REST-сервис
Kafka Streams
Хранилище
состояния
REST-сервис теперь подключается
непосредственно к локальному
хранилищу состояния, извлекая
текущие результаты из потока данных
Приложение Kafka Streams потребляет данные
с брокера, а локальное хранилище состояния
захватывает текущее состояние потока

Однако хранилище состояния одно на task



Однако хранилище состояния одно на task

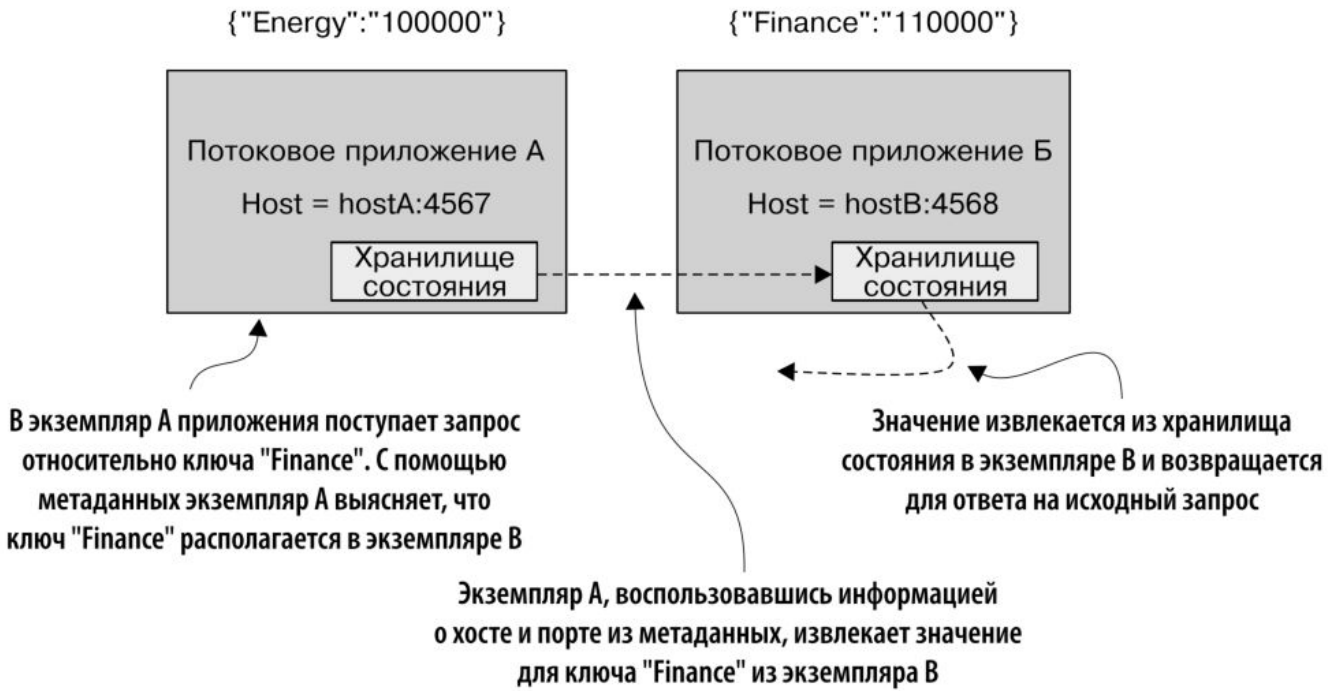


Схема работы

- Задаем APPLICATION_SERVER_CONFIG в каждом приложении (host:port)
- Обрабатываем входящий запрос
  - allMetadataForKey - возвращает host:port для заданного ключа
  - если это наш экземпляр - выдаем ответ на основе локального хранилища
  - если не наш - запрашиваем нужный экземпляр
- Kafka Streams не предоставляет готовый rpc

ex13.db

План демо

- Запускаем ManualPublisher
- Меняем логирование kafka на INFO
- Запускаем Application (-Dserver.port=8080)
  - http://localhost:8080/count?industry=a
  - http://localhost:8080/count?industry=d
- Запускаем Application (-Dserver.port=8081)
  - повторяем
- Гасим Application (8081)
  - ждем ребалансировки
  - повторяем

ex13.db

Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

Рефлексия

Ключевые тезисы

1. KStream - поток событий из топика
2. KTable - интерпретация топика как потока обновлений таблицы
3. GlobalKTable - берет данные из всех партиций
4. Processor API - низкоуровневое API прямой настройки топологии. Требуется редко
5. Можно запрашивать данные из локального хранилища, не используя БД

Заполните, пожалуйста, опрос о занятии по ссылке в чате

Спасибо за внимание!

Приходите на следующие вебинары

Непомнящий Евгений

Разработчик Java/ Kotlin IT-Sense

@evgeniyN