

Exactly-Once обработка данных

Практический стенд этого урока автоматически запускает кластер Apache Kafka, состоящий из 3 брокеров, установленных на 3 виртуальных машинах. Каждый брокер адресуется как `node-{1-3}.<ваш логин>.slurm.io`

Заходим на наши брокеры в **3-х консольных окнах**, заменив плейсхолдер `<ваш номер студента>` на ваш личный номер студента:

```
ssh <ваш номер студента>@sbox.slurm.io
ssh node-{1-3}.<ваш номер студента>
export USER=<ваш номер студента>
```

1.

Клонируем репо и соберем тестовых клиентов **на всех брокерах**

```
git clone https://gitlab.slurm.io/edu/kafka.git
cd kafka
```

2. `mvn package`

Создаем топики для теста на любом из брокеров

```
/opt/kafka_2.13-2.7.0/bin/kafka-topics.sh --bootstrap-server node-1.$USER:9092
--topic transactions-input --replication-factor 3 --partitions 3 --create
```

3. `/opt/kafka_2.13-2.7.0/bin/kafka-topics.sh --bootstrap-server node-1.$USER:9092 --topic transactions-output --replication-factor 3 --partitions 1 --create`

4. Запускаем идемпотентного продюсера для записи случайно сгенерированных оплат в топик `transactions-input` на любом из брокеров

```
java -cp
test-clients/target/test-clients-1.0-SNAPSHOT-jar-with-dependencies.jar
io.slurm.kafka.TestProducer -b node-1.$USER:9092 -c 20000000 -t
transactions-input -i -s 100
```

5. Запускаем инстанс нашего exactly-once приложения, которое будет отфильтровывать оплаты с суммами свыше 90 долларов из топика `--in` и отправлять их в выходной топик `--out` на любом из брокеров

```
java -cp
test-clients/target/test-clients-1.0-SNAPSHOT-jar-with-dependencies.jar
io.slurm.kafka.ReadProcessWriteExactlyOnceApp -b node-1.$USER:9092 --group
my-processor --id 1 --in transactions-input --out transactions-output
```

Параметр `--id` должен быть уникальным для каждого из инстансов.

Попробуйте запустить несколько инстансов с одинаковым значением!

6. Убедимся, что отфильтрованные сообщения действительно пишутся в выходной топик на любом из брокеров

```
/opt/kafka_2.13-2.7.0/bin/kafka-console-consumer.sh --bootstrap-server  
node-1.$USER:9092 --topic transactions-output
```

7. Убедимся, что приложение действительно коммитит оффсеты на любом из брокеров

```
/opt/kafka_2.13-2.7.0/bin/kafka-consumer-groups.sh --bootstrap-server  
node-1.$USER:9092 --group my-processor --describe
```

8. Поднимем еще один инстанс* нашего приложения с другим `--id` на любом из брокеров

```
java -cp  
test-clients/target/test-clients-1.0-SNAPSHOT-jar-with-dependencies.jar  
io.slurm.kafka.ReadProcessWriteExactlyOnceApp -b node-1.$USER:9092 --group  
my-processor --id 2 --in transactions-input --out transactions-output
```

* консьюмер в приложении использует Cooperative Rebalancing, благодаря чему, избегается stop-the-world эффект при подключении нового инстанса!

9. Убедимся, что оба инстанса обрабатывают уникальные партиции входного топика

- В логе приложения видим “Committed offsets” для обрабатываемых топик-партиций

- Сделаем describe группы и увидим 2 живых инстанса:

```
/opt/kafka_2.13-2.7.0/bin/kafka-consumer-groups.sh --bootstrap-server  
node-1.$USER:9092 --group my-processor --describe
```

10. Попробуйте провести дополнительные операции с приложениями, например:

- Остановите один из инстансов: партиции будут переданы живому инстансу через `session.timeout.ms`

- Остановите оба инстанса: при перезапуске обработка будет восстановлена с последней закоммиченной позиции

- Консьюмер в приложении также сконфигурирован с функционалом Static Membership: попробуйте перезапустить инстанс в пределах `session.timeout.ms` (10 сек) и вне этого лимита

11. Попробуйте отключить Cooperative Rebalancing и Static Membership опциями `--no-cooperative-rebalancing` `--no-static-membership`, после чего остановите консьюмеры и запустите снова. Попробуйте останавливать один из инстансов и поднимать его обратно через 5-10 секунд — замечаете ли вы

разницу во времени ребаланса группы по сравнению со включенным Cooperative Rebalancing?*

*Если вы видите `InconsistentGroupProtocolException` — подождите `session.timeout.ms` (10сек) пока все инстансы будут помечены, как мертвые брокером.