Testing Kafka Streams Applications

Lee Dongjin | dongjin@apache.org

Overall

- Introduce how to run [unit|integration] test for Kafka Streams Application.
- Problem: So unstable …
  - Slides: based on 1.1.0
  - Example: based on 2.0.0

Problem: Testing Kafka Streams Application

- Using running Kafka instance
  - **DON'T DO IT - WASTE OF RESOURCES & ENERGY.**
  - Hard to deploy, Non-isolated, Slow, and Non-repeatable (side effects!)
- The other strategies (From 1.1.0)
  - `EmbeddedKafkaCluster`
  - `TopologyTestDriver`
  - `MockProcessorContext`

Testing Strategies: Overall

- `EmbeddedKafkaCluster` strategy
  - An In-memory, embedded instance of a Kafka cluster.
  - Identical to the real instance - you can use it with embedded zookeeper, schema registry instance.
- `TopologyTestDriver` strategy
  - An official test utility for `Topology`.
  - Very similar to the real instance, but not identical.
- `MockProcessorContext` strategy
  - Provides (almost) complete checkup for `Processor` instance.
- Each strategy has its advantages & disadvantages.

Testing Strategies: Comparison

	`EmbeddedKafkaCluster`	`TopologyTestDriver`	`MockProcessorContext`
Reality	Best	Bad	Good
Granularity	Good	Bad	Best
Speed	Bad	Good	Best
Difficulty	Easy	Easy	Hard

How to use `EmbeddedKafkaCluster` (1)

1. Add Kafka Streams test dependency
    - `org.apache.kafka:kafka-streams:${kafka.version}:test`
2. Create EmbeddedKafkaCluster instance & Topics.
3. Run test by starting Kafka Streams instance.
4. Next slide!

How to use `EmbeddedKafkaCluster` (1)

```
import org.apache.kafka.streams.integration.utils.EmbeddedKafkaCluster;
import org.apache.kafka.streams.integration.utils.IntegrationTestUtils;

...

public class WordCountTest {

    @ClassRule
    private static final EmbeddedKafkaCluster CLUSTER = new EmbeddedKafkaCluster(1);

    @Before
    public void before() {
        CLUSTER.createTopic("inputTopic");
        CLUSTER.createTopic("outputTopic");
    }
}
```

Requires `org.apache.kafka:kafka-streams:${kafka.version}:test` dependency

Create Embedded cluster instance & Add topics

How to use `EmbeddedKafkaCluster` (2)

3. Test with `IntegrationTestUtils`

```java
@Test
public void test() {
    // Create KafkaStreams instance
    kafkaStreams.start()

    // Send records to the input topic synchronously
    IntegrationTestUtils.produceValuesSynchronously(inputTopic, inputValues,
                                                    producerConfig);

    // Retrieve records from the output topic synchronously
    List<KeyValue<String, Long>> actualWordCounts =
        IntegrationTestUtils.waitUntilMinKeyValueRecordsReceived(consumerConfig,
                                outputTopic, expectedWordCounts.size());
    assertThat(actualWordCounts).containsExactlyElementsOf(expectedWordCounts);
}
```

How to use `TopologyTestDriver` (1)

1. Add dependency
   - `org.apache.kafka:kafka-streams-test-utils:${kafka.version}:test`
2. Create `TopologyTestDriver` instance
3. Run test with `TopologyTestDriver#pipeInput`, `TopologyTestDriver#readOutput`

How to use `TopologyTestDriver` (2)

```
    ...

    @Test
    public void test() {
        // driver = new TopologyTestDriver(topology, props);
        driver.pipeInput(recordFactory.create(INPUT_TOPIC, "key1", "value1"));
        driver.pipeInput(recordFactory.create(INPUT_TOPIC, "key2", "value1"));

        final ProducerRecord<String, String> record = driver.
            readOutput(OUTPUT_TOPIC, STRING_DESERIALIZER, STRING_DESERIALIZER);

        assertEquals(...);
    }
}
```

Feed input with `pipeInput` method, one by one.

Retrieve output with `readOutput` method, in order.

Summary

1. Choose an adequate strategy for your case.
    - Tradeoff: Reality but Slow vs. Non real but Fast
2. `EmbeddedKafkaCluster` approach is better for the integration test, while `TopologyTestDriver`, `MockProcessorContext` approach is better for unit test.
3. However, If you have insufficient experience, test with `EmbeddedKafkaCluster` first (i.e., identical to the real cluster) and add other tests later.

Questions?

- Slides: speakerdeck.com/dongjin
- Example project: github.com/dongjinleekr/kafka-streams-example