POLITECNICO
MILANO 1863

Apache Kafka

Alessandro Margara
alessandro.margara@polimi.it
https://margara.faculty.polimi.it

Rules

- Rename the ConsumersXX.java file replacing XX with the number of your group
- Write in the comment on top of the class your group number and the name of all group members
- Write additional information on topic partitioning and consumer groups in the comment on top, as discussed in the following slides
- Submit only a single java file with your solution
  - Submitted from the contact email provided in the group registration document

Assumptions

- A Producer class publishes messages to inputTopic
  - You may set the number of partitions for inputTopic using the TopicManager class
  - Message keys are String, message values are Integer
  - You may assume that the producer never crashes

Assumptions

- Two consumers read from inputTopic

1. Consumer1 computes the sum of the values of all messages in a tumbling window of 10 elements*
2. Consumer2 computes, for each key, the sum of all the values of messages in a tumbling window of 10 elements*

*In a tumbling window of 10 elements, both the size and the slide of the window consist of 10 elements

Assumptions

- Consumer1 writes its results on topic outputTopic1
  - The key is always "sum"
- Consumer2 writes its results on topic outputTopic2
  - The key is the same as the input records

- The two consumers need to ensure the following guarantees

1. Consumer1 provides exactly once guarantees, meaning that messages are not lost or duplicated, even in the case of failure
2. Consumer2 does not provide any guarantee in the case of failures: messages may be lost or duplicated

Exercise

- Complete the implementation of Consumer1 and Consumer2 in the ConsumersXX.java file

- The Consumers main method takes in input two parameters:
  - The number of the Consumer to start (1 or 2)
  - The consumer group
  - You may add additional parameters, if needed

- Write in the comments on the top of the file
  - The minimum and maximum number of partitions allowed for each topic (inputTopic, outputTopic1, outputTopic2) to satisfy the requirements above
  - The minimum and maximum number of instances of Consumer1 and Consumer2 allowed to satisfy the requirements above
  - How you define the consumer groups for the various instances of Consumer1 and Consumer2

Hints

- You can add print statements to your consumers, to verify that they work as expected
  - The Producer generates deterministic values
- From a ConsumerRecord, you can extract:
  - The topic, with the topic() method
  - The partition, with the partition() method
- You can create a TopicPartition object using the constructor
  - TopicPartition(String topic, int partition)