POLITECNICO
MILANO 1863

Apache Kafka

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

Rules

- Complete the java files as indicated by each exercise
  - Only add code in the locations indicated by TODO comments
- Complete the README.md file with
  - Number of your group (from the group registration document)
  - Name of each group member
  - Number of partitions allowed for each exercise (minimum, maximum)
    - Maximum can be set to N if no upper bound is required
  - Number of consumers allowed for each exercise (minimum, maximum)
  - Argument to start each consumer
- Create and submit a single zip file with your project
  - Rename the project NSDS_kafka_eval_groupXX (XX is the number of your group in the group registration document)
  - Name of the file: kafka-groupXX.zip
  - Submit from the contact email provided in the group registration document

Assumptions

- One instance of the Producer class publishes messages into the topic “inputTopic”
  - The producer does not fail
  - Message keys are String
  - Message values are Integer
- You may set the number of partitions for “inputTopic” using the TopicManager class
  - Indicate in the README.md file the minimum and maximum number of partitions allowed
- The consumers in the two exercises should be able to run simulataneously
- Consumers take in input one argument, representing their consumer group

Exercise 1

- Complete the implementation of AtMostOncePrinter
  - It consumes messages from "InputTopic"
  - It prints messages on the standard output
  - It prints only messages with a value greater than *threshold* (which is an attribute of the class)
  - It provides at most once semantics
    - All messages that overcome the threshold need to be printed 0 or 1 times
    - Messages may be missed only in the case of a failure
      - The trivial solution that never prints any message is not acceptable

Exercise 2

- Implement a PopularTopicConsumer
  - It computes the number of messages received for each topic
  - Upon receiving a new message, it prints the topic (or topics) that received the highest number of messages so far
  - It does NOT provide guarantees in the case of failures
    - Input messages may be lost or considered more than once in the case of failures
    - Thus, the computation of the number of messages may be incorrect in the case of failures