Practical Assignment II (KAFKA)

1 Introduction

The practical assignment II is focused on using Apache Kafka´s Quality Attributes.

2 Description

The project is about a simulation of processing data from sensors in a Kafka cluster. There will be several Use Cases and for each one a solution must be built.

The file sensor.txt contains all the data collected from several sensors. The format is as follows:

XXXXX -> string containing the sensor ID
ZZZ.ZZ -> real number containing a temperature in ºC.
YYYYYY -> integer time stamp

The data is ordered by its time stamp.

3 Use Cases

General Requirements

- Kafka Cluster, beyond Zookeeper, comprises: 6 Brokers, one Topic named as Sensor, 6 Partitions, 3 Replicas and 2 min.insync.replicas
- Beyond the Kafka Cluster (Brokers and Zookeeper) there are at least three additional processes:
    - PSOURCE: responsible for reading sensor data (records) from the file sensor.txt and sending it to one or more Producers via Java Sockets
    - PPRODUCER:
        - responsible for receiving records from SOURCE and sending them to the Kafka Cluster according to the requirements
        - may comprise one or more Kafka Producers
        - at least one GUI is required depending on the use case
    - PCONSUMER:
        - responsible for reading records from the Kafka Cluster and to process them according to the requirements
        - may comprise one or more Kafka Consumers
        - at least one GUI is required depending on the use case
- Each Use Case (UC) must be implemented in a way it can be demonstrated separately from the remaining Use Cases
- Create one Java project only (PA2_GX, X is your group id) and for each Use Case:
    - create one Package named as UC_Z (Z is the UC id)
    - 3 processes: PSOURCE, PPRODUCER and PCONSUMER

- Whenever necessary, concurrency must be used to improve performance
- In each Producer class and in each Consumer class it's mandatory to instantiate a Java Properties object and set into it all the required properties
- If a property (from a Producer or Consumer) is important you must set its value explicitly even if you intend to use the default value.
- You cannot use: idempotence property and Kafka Transactions.
- You must use the KAFKA libraries published in the e-learning platform
- Do not use MAVEN

UC_1

PSOURCE:

Role:	records are read from sensor.txt and sent to PPRODUCER in accordance with the requirements

PPRODUCER:

Role:	records are displayed and sent to the Kafka Cluster
Performance:	maximum throughput, any latency
Data ordering:	must keep original order of all records
Data loss:	records of any sensor can be lost
Data duplication:	records cannot be duplicated

PCONSUMER:

Role:	records are read from the Kafka Cluster and displayed
Performance:	maximum throughput
Data ordering:	records processed in their original order
Data duplication:	records can be reprocessed

UC_2

PSOURCE:

Role:	records are read from sensor.txt and sent to PPRODUCER in accordance with the requirements

PPRODUCER:

Role:	records are displayed and sent to the Kafka Cluster
Performance:	maximum throughput, any latency
Data ordering:	must keep original order but by sensor ID only
Data loss:	records of any sensor can be lost
Data duplication:	records cannot be duplicated

PCONSUMER:

Role:	records are read from the Kafka Cluster and displayed
Performance:	maximum throughput
Data ordering:	records processed in their original order by sensor ID
Processing:	records can be reprocessed

UC_3

SOURCE:

Role:	records are read from sensor.txt and sent to PPRODUCER in accordance with the requirements

PPRODUCER:

Role:	records are displayed and sent to the Kafka Cluster
Performance:	minimum latency and try to maximize throughput
Data ordering:	must keep original order by sensor ID only
Data loss:	records of any sensor can be lost
Data duplication:	data cannot be duplicated

PCONSUMER:

Role:	records are read from the Kafka Cluster and displayed
Performance:	maximum throughput
Data ordering:	records processed in their original order by sensor ID
Data duplication:	records cannot be reprocessed

UC_4

SOURCE:

Role:	records are read from sensor.txt and sent to PPRODUCER in accordance with the requirements

PPRODUCER:

Role:	records are displayed and sent to the Kafka Cluster
Performance:	minimum latency and try to maximize throughput
Data ordering:	records can be reordered
Data loss:	minimize the possibility of losing records
Data duplication:	records cannot be duplicated

PCONSUMER:

Role:	records are read from the Kafka Cluster and the maximum and minimum temperatures are displayed and computed following the Voting Replication tactic
Performance:	maximum throughput
Data ordering:	records processed in any order
Data duplication:	records can be reprocessed

UC_5

SOURCE:

Role:	records are read from sensor.txt and sent to PPRODUCER in accordance with the requirements

PPRODUCER:

Role:	records are displayed and sent to the Kafka Cluster

Performance:	maximum throughput, any latency
Data ordering:	must keep original order of all records
Data loss:	records cannot be lost
Data duplication:	records can be duplicated

PCONSUMER:

Role:	records are read from the Kafka Cluster and the average temperature is displayed and computed following the Voting Replication tactic
Performance:	maximum throughput
Data ordering:	records processed in any order
Data duplication:	records cannot be reprocessed

4 Assignment Submission