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 designed.

The file sensor.txt (available in the e-learning platform) contains all the data collected from several sensors. The format is as follows:

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

The data is ordered by its timestamp.

You are encouraged to improve each UC namely providing additional properties which can influence any Quality Attribute.

The rules defined for PA1 are also

3 Use Cases (UC)

General Requirements

- A Kafka Cluster, beyond Zookeeper, includes:
    - 1 to several Brokers
    - 1 Topic named as Sensor
    - 1 to several Partitions
    - 0 to 2 Replicas
- Beyond the Kafka Cluster (Brokers and Zookeeper) there are two additional processes:
    - PProducer
        - responsible for reading sensor data (records) from the file sensor.txt and sending it to the Kafka Cluster according to the requirements of each UC
        - there can be only one Thread responsible for reading the data from sensor.txt
        - may include one or more Kafka Producers
        - each Kafka Producer must be implemented as a Java Thread

        - the interaction between Threads must be implemented as Shared Regions
        - all Java Threads (producers) must run inside the process PProducer
        - PProducer contains one process only: PProducer.java
    - PConsumer:
        - responsible for reading records from the Kafka Cluster and to process them according to the requirements of each UC
        - may include one or more Kafka Consumers
        - each Kafka Consumer must be implemented as Java Thread
        - may include one or more Group Consumers
        - one GUI by Kafka Consumer and/or by Group Consumer
        - all Java Threads must run inside the process PConsumer
        - PConsumer contains one process only: PConsumer.java
- All Thread Classes must start with the letter T, such as TCustomer
- All Monitors must start with the letter M, such as MFifo
- Each Monitor must be an individual java class
- All Interfaces must start with the letter I, such as IConsumer
- The access to Monitors must be through Interfaces provided their methods
- Threads (Active Entities) that access Monitors must receive as argument the correspondent interfaces
- All variables must be private (if not, a justification must be provided)
- Whenever possible, Constructors must be declared as private (use a *getInstance* method to instantiate classes and return an interface)
- Variables whose initial value does not change must be declared as final
- ReentrantLock is much more flexible than synchronized methods and provide additional functionalities. The usage of ReentrantLock is mandatory. Nevertheless, **you must know** how synchronized methods work.
- Methods' and variables' names must be semantically oriented whenever convenient
- Each Use Case must be implemented in a way it can be demonstrated separately from the remaining UCs
- Create one Java project only (PA2_GXX, XX is your group id) and for each UC:
    - create one Package named as UCn (n is the UC id)
    - inside UCn create three packages: Producer, Consumer and Scripts
- In each Kafka Producer class and in each Kafka Consumer class it is mandatory to instantiate a Java Properties object and set into it <u>all</u> the required properties
- If a property (from a Kafka Producer or Kafka Consumer) is important you must <u>set its value explicitly</u> even if you intend to use the default value
- You cannot use: idempotence property and Kafka Transactions.
- The solution for each UC must be considered the optimal one
- You must use the KAFKA libraries published in the e-learning platform
- You cannot use MAVEN

UC_1

Objective: learn how to use Kafka

Kafka Cluster:

- Three Brokers
- Three partitions

PProducer:

- one to three Kafka Producers
- three sent modes: fire and forget, asynchronous and synchronous (the same for all Producers)
- several properties are configurable
- GUI:
    - button to start
    - combo box to define sending mode
    - combo box to define the number of Producers
    - components to define the values of several Producers' properties
    - display sent records
    - display total time to send all records

PConsumer:

- one Kafka Consumer
- GUI: display received records

UC_2

Objective:

- check kafka high-availability
- the primary Broker goes down but the system continues working

Kafka Cluster:

- Three Brokers:
    - one primary and two replicas
    - ack=-1, retries=5, in-sync=-1
- one partition

PProducer:

- one Kafka Producer
- three sent modes: fire and forget, asynchronous and synchronous
- GUI:
    - button to start
    - combo box to define sending mode
    - display sent records

PConsumer:

- one Kafka Consumer
- GUI: display received records and total number

UC_3

Objective:
- kafka scalability and performance
- use of partitions and partitioner policies

Kafka Cluster:
- 3 brokers
- 3 partitions

PProducer:
- 3 Kafka Producers
- three sent modes: fire and forget, asynchronous and synchronous (the same for all Producers)
- use the patitioner policies
- GUI:
  - button to start
  - combo box to define sending mode
  - combo box to define the partitioner policy
  - display sent records by Producer

PConsumer:
- three Kafka Consumers
- one consumer for each partition
- GUI: for each Kafka Consumer, display received records and total number

UC_4

Objective:
- kafka high-availability
- use of Kafka Consumer Groups
- use of commit modes
- use of Group Rebalance Listener

Kafka Cluster:
- 3 Brokers
- 3 partitions

PProducer:
- 1 Kafka Producer
- GUI:
  - Button to start
  - display records by producer and total number

PConsumer:

- one Consumer Group with 1 to 3 Kafka Consumers
- Common GUI:
    - combo box to define the number of Consumers
    - define the commit mode
    - use or not use of Group Rebalance
- GUI for each Kafka Consumer:
    - display the received records and the total number
    - button to kill a Consumer
    - display total time to receive all records from the first one

4 Assignment Submission

- Project and folder root name must be: PA2_GXX, where XX is your group number;
- Each project must be compressed in **7z** format and sent to omp@ua.pt till 2023 May 1, 09:00;
- Do not send Kafka libraries but send any additional library required to run your project;
- Report (PDF):
    - For each UC:
        - what is implemented and working correctly
        - what is not implemented or not working correctly
- Assessment Rules:
    - requirements
    - Java Doc
    - Report