

CS 495: Software Engineering for Distributed Systems

Distributed Messaging System

Objective:

The objective of this assignment is to make use of the introduced concepts so far in the labs to build a miniature distributed system. These concepts include but not limited to “Socket Programming” & “Multi-threading”. The system in question here is a messaging system. Messaging systems are responsible for transferring data between different producers & consumers in a standardized way to remove the headache of having to deal with how multiple systems can share their data. If all producers write to the messaging system, then all consumers would be able to read them regardless of their source.

Requirements:

There are 3 types of nodes in the system

1. Broker Nodes: These are the servers that hold the data
2. Producer Nodes: These are the clients that connect to broker nodes to write messages
3. Consumer Nodes: These are the clients that connect to broker nodes to read messages

Broker Nodes:

There are 2 types of brokers. Leader Brokers and Partition Brokers. There can only be 1 leader in a system.

The Leader Broker is the coordinator of the system, and the node all clients should connect to. It also keeps track of the list of Partition Brokers and which messages they store.

Partition Brokers are only data storage nodes. They store the messages sent to them by the Leader Broker. A message can be any data string representing (A simple string, JSON, XML etc...).

Producer Nodes:

Producers are clients that put data into the system. Your producer should have a GUI to connect to a particular Leader Broker and a dialog to enter data the user wants to put into the system.

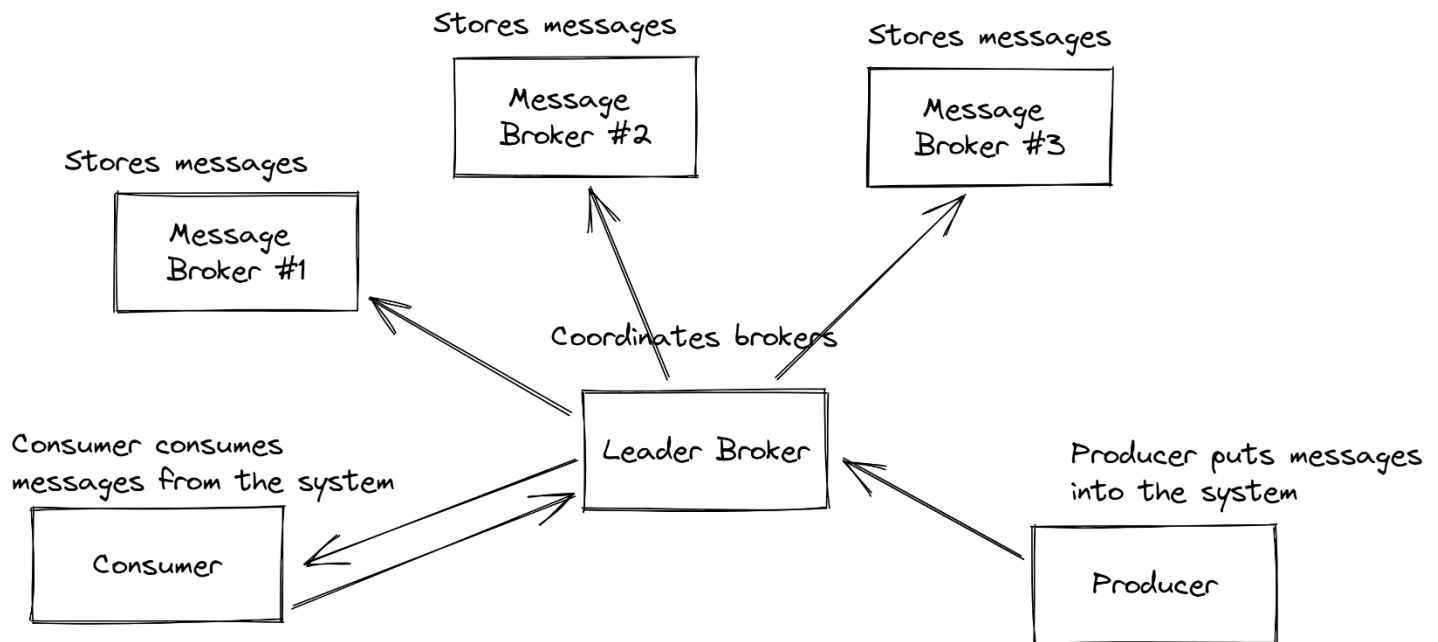
Consumer Nodes:

Consumers are clients that get data from the system. Your consumer should have a GUI to connect to a particular Leader Broker and a box that shows all the data it consumed. Your consumer should refresh every 30 seconds and get only the messages it didn't already fetch from the system.

Notes:

- All your nodes should be able to handle concurrent requests.
- Your system should be able to handle multiple partition brokers, consumers & producers at the same time.
- You are free to implement your own communication protocol for exchange between the nodes.

High Level System Overview:



Deliverables

1. Sequence Diagrams showcasing your application flow. These include but are not limited to
 - a. Your broker startup
 - b. Consuming messages
 - c. Producing messages
2. Your diagrams shouldn't only showcase the flow but also the message formats and communication protocol you created for your application
3. The source code for your developed application

Rules of Submission

1. Deadline is Saturday 9th of April (9/4/2022) at 11:59 PM through blackboard ONLY.
2. No email submissions will be accepted.
3. There are no late submissions.
4. The assignment is in groups of 2 (at most) from the same lab
5. Cheating is not tolerated and formal actions will be taken.
6. All of the deliverables should be bundled in a single ZIP file named "CS454_A1_<ID1>_<ID2>.zip"