CS 3311 Middleware Message Queues

Lab 6 Tue 14/11 **2017**

Student Name: **Id:**

Laboratory Goals / Objectives

Upon completion of the lab, the student should have extended the experience of using queues, by programming a distributed application that uses queues as supporting middleware.

Distributed application

Let's consider a distributed application made of 1, 2, 3 or more clients that avail of services provided by several servers. Each server provides a different operation. For example, for simplicity, we may consider that one service is finding patterns in a string, the second does implement a sorting algorithm, and the third multiplies matrices. Of course, more complex operations can be considered.

For supporting the client-server interaction, we consider queues and a convention for the message format (type, size, operands, client ID). There is a general queue where clients send their messages with the requested operation and the corresponding operands. We call this queue the Demand queue. The servers consume the appropriate message that matches each individual service, and will return the result as a message in another queue. We call this the Result queue. Alternatively, there can be a result queue for each client. The clients receive the results.

To do:

Task 1

Draw a diagram that shows the model of the application (clients and servers) and queues that you consider. Present the convention used for the message format. Write in your own words how the system works, for example how a server will select the appropriate message for its service, and how the client receives the right result.

Task2

Program in Python (or Java) the application (clients, servers and queues) and test its functionality. You can use the code developed in the previous lab.

Note: you can choose and program your own services, not necessarily those mentioned above.

Questions

The following questions are to be filled in individually by each student and returned by the due date.

1. What are the benefits of using message queues in respect to remote method invocation in a client/server application?

2. Provide the diagram corresponding to task 1 and your analysis regarding the format of the demand/result messages and client/server manipulation of messages.

3. Provide the code for task 2 and screenshots that show how the system works. Add comments in your code.

4. Your additional comments on implementing and testing your code.