**Northeastern University – Silicon Valley**

CS 6650 Scalable Dist Systems

**Homework Set #4 Assigned: 2/2/20 Due: 2/14/20** [100 points]

***INSTRUCTIONS: Please provide clear explanations in your own sentences, directly answering the question, demonstrating your understanding of the question and its solution, in depth, with sufficient detail. Submit your solutions [PDF preferred]. Include your full name. Do not email the solutions.***

For I below, tudy **Chapter 6 from** Coulouris Book

1. Answer the following questions using explanation and diagrams as needed. No implementation needed.
2. 6.8 [10 points]

Suggest a design for a notification mailbox service that is intended to store notifications on behalf of multiple subscribers, allowing subscribers to specify when they require notifications to be delivered. Explain how subscribers that are not always active can make use of the service you describe. How will the service deal with subscribers that
crash while they have delivery turned on? page 245

Answer:

The Mailbox service will provide an interface so that a client can register interest in another object.

The client needs two things: (1) *RemoteEventListener* for getting notifications from Mailbox service and (2) a means to interacting with the Mailbox service for turning delivery on and off.

The *RemoteEventListener* is provided by the Mailbox service so that notifications may be passed from event generators to the *RemoteEventListener* and further to the client.

To enable client to interact with the Mailbox service for the on and off of delivery, we could define register as follows:

*Registration register()*…

The result is a reference to a remote object and this object whose methods enable the client to get a reference to *RemoteEventListener* and to turn the delivery on and off.

To use the Mailbox service, the client registers with it and receives a *Registration* object, which it saves in a file. It registers the *RemoteEventListener* with all of the *EventGenerators* whose events it wants to have notification of. If the client crashes, it can restore the Registration object when it restarts. The client can turn the delivery on and off as it desires.

The server needs to have up-to-date lists of the clients’ callback objects, but clients may not always inform the server before they exit, leaving the server with incorrect lists. To solve this problem, the design should make it possible to specify a lease for each subscriber/ client.

1. 6.14 [10 points]

**Consider the version of the FireAlarm program written in JMS (Section 6.4.3). How would you extend the consumer to receive alarms only from a given location?** page 261

**Answer:**

The header of the message consists of properties which are all user-defined and can be used to associate other application-specific metadata elements with a message. For example, if implementing a context-aware system, the properties can be used to express additional context associated with the message, including a location field.

It is possible to associate filters with message consumers by specifying a message selector (a predicate defined over the values in the header and properties part of the message). By using SQL to specify properties, we can filter messages from a given location in the context-aware system.

1. 6.15 [10 points]

Explain in which respects DSM is suitable or unsuitable for client-server systems. page 262

Answer:

DSM is in general less appropriate in client-server systems, where clients normally view server-held resources as abstract data and access them by request (for reasons of modularity and protection). It is unsuitable for client-server systems as it is not conducive to heterogeneous working. For security, we would need a shared region per client, which would be expensive.

However, DSM may be suitable for client-server systems in some application domains such as when multiple clients share server responses.

1. Implement a simple JMS Queue which can be useful for the notification mailbox discussed in Question 6.8 above. See the reference provided. [30 points]

Why do we need MyListener and MyReceiver differently in JMS Queue?

1. See the Coding Tutorial PDF provided, and the below references. Do your own study of RMI Java examples to implement this. Implement a Java RMI Application in which the Client object is sending a list of 10 integers to the Server, and a remote method ['sort()'] o the server returns a sorted version of the same list back to the Client. [40 points]

**References**

JMS

https://howtodoinjava.com/jms/jms-java-message-service-tutorial/

Java RMI

<https://www.cs.uic.edu/~troy/fall04/cs441/rmi/calc/index.html>

<https://www.cs.uic.edu/~troy/fall04/cs441/rmi/calc/index.html>