COMP6231 Assignment2

Name: Kaichen Zhang **ID: 40000160**

Techniques:

In this assignment:

I used **Java IDL** definition to define the CORBA interface, used IDL compiler to generate: _ServerInterfaceStub, ServerInterface, ServerInterfaceHelper, ServerInterfaceHolder, ServerInterfaceOperations ServerInterfacePOA, which can realize platform transparency and language transparency. Such package can act like the interface to realize the communication between clients (passenger and manager) and servers (MTL,WDC,NDL) like RMI.

I used **UDP** to implement the communication between servers.

I used **HashMap** to store the passenger records and flight record.

I used **multithreading** technique to implement that multiple clients can act simultaneously.

I used **synchronization** technique to keep the integrity of data while modifying it, so the server can maximize the concurrency.

Design architecture:

clients package:

Client class defines the human machine interaction, by simulate both passenger client and manager client.

MultiThreadTest class uses multithreading to test multi clients interaction. Which can check the performance of synchronization.

Records package:

Defining the flight record data structure, passenger record data structure. Both are using hash map to store.

servers package:

Communications are made through ORB.

Corba Objects extend POA class and implement the methods

Containing the FlightServer class, providing remote method invocation by extends ServerInterfacePOA, implemented by MTL,WDC and NDL servers.

FlightServer class overrides the bookFlight method, but put a new passenger into the hash map using his first character of last name as key, and passenger records as value.

FlightServer class overrides the getBookedFlightCount method by sending UDP messages to other servers, and receive UDP counter reply messages from others.

FlightServer class overrides editRecord method by editing expected flight record field, giving record ID, filed name and new value.

FlightServer class overrides transferReservation method, by sending passenger record using UPD message from original city to other city. And making reservation at the other city, canceling passenger record at the original server automatically.

DFRSApp package:

Generated by using IDL compiler to compile DFRSApp.idl. It declares four methods: bookFlight, getBookedFlightCounts,editRecord and transforReservation.

Contains:

ServerInterfaceStub:The Java class HelloStub is the stub file, the client-side proxy, which interfaces with the client object

ServerInterface

ServerInterfaceHelper: The Java class HelloHelper provides auxiliary functionality needed to support a CORBA object in the context of the Java language.

ServerInterfaceHolder: The Java class called HelloHolder holds (contains) a reference to an object that implements the Hello interface.

ServerInterfaceOperations: It is known as a Java operations interface in general

ServerInterfacePOA: The Java class HelloImplPOA is the skeleton, the server-side proxy, combined with the portable object adapter.

(reference from COMP 6231, Fall 2016 Distributed Objects and CORBA. Prof. M.L. Liu, California Polytechnic State University)

Scenarios:

Firstly we should run the server of three different cities.

Human machine interaction scenario:

To simulate the passenger, we run the client class and choose to perform as passenger, and we can do the operations:

Entering valid information and booking a ticket.

To simulate the manager, we run the client class and choose to perform as manager, and we can do the operations like:

Entering valid manager ID.

1. Manage flight information:

1) return all available flights at the current server 2) edit flight information 3) back

2. Get booked flights counts by sending UDP

3. Transfer reservation by entering existing passenger ID and other city name.

4. Exit

Automatic Multi Client scenario:

Test how multi clients can perform the operations simultaneously and keeping data clean.

Important part/ Difficulty:

1. Setting IDL and ORB properties
2. In the last assignment I transfer java objects as value, but in CORBA if we want to transfer object we have to cast the data to byte array stream when we send it, and cast it back to object when the receiver receives it. To make such change is more complemented than changing the method paramaters, and using console to perform relevant scenarios.
3. For the log part I can log the performance and actions from human machine scenarios, but for the multi threading test we can only check it from conlose.
4. Also to understand all files generated by IDL, and how to fully setting CORBA is important.
5. It's slow and difficult to debug the system.