Distributed System Design

COMP 6231

Instructor: R. Jayakumar

Distributed Appointment Management System using Java RMI

BY: Raviraj Savaliya (40200503)

Introduction: Distributed appointment management System (DAMS) for health care is a distributed system that has 3 hospitals namely Montreal (MTL), Quebec (QUE) and Sherbrooke (SHE). DAMS is managed and used by Patients and Admins. In this system Patient can perform few operations as listed below.

1. Book Appointment
2. Get Booked Appointment
3. Cancel Booked Appointment
4. List of all available Appointment

Whereas Admin can perform all patient's operation and some additional operation such as:

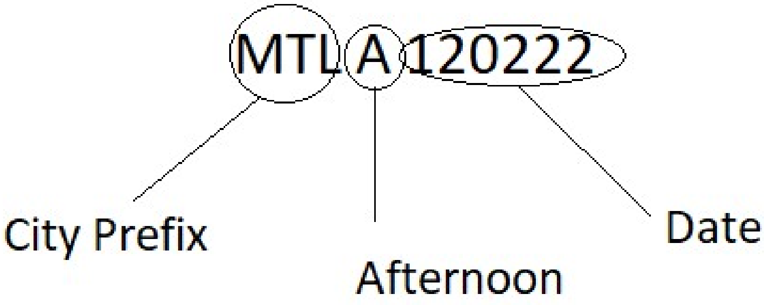
1. Add new Appointments
2. Remove Appointments

In DAMS, 3 servers are used for manage all clients requests. These servers name is:

1. Montreal
2. Quebec
3. Sherbrooke

Both patient and admins are identified by a unique adminID and patientID respectively, which is built from the acronym of their hospital's city and a 4 digit number (e.g. MTLA1111 for admin and MTLP1111 is for patient). We can identify the user by the 4th character of ID. If it is A then admin and if P then patient. DAMS system also maintains logs for server and client.

There are 3 types of admins for different server. They creates appointment slots of 3 different types such as Physician, Dental and Surgeon. There is three different time slot Morning(M),Afternoon(A) and Evening(E). Appointment ID is combination of city, time slot and appointment date (e.g. MTLM120222, QUEE120222).



Data Management:

The appointments data is stored in HashMap. Appointment type is key and Appointment Id

Is sub key and sub value is capacity of appointment.

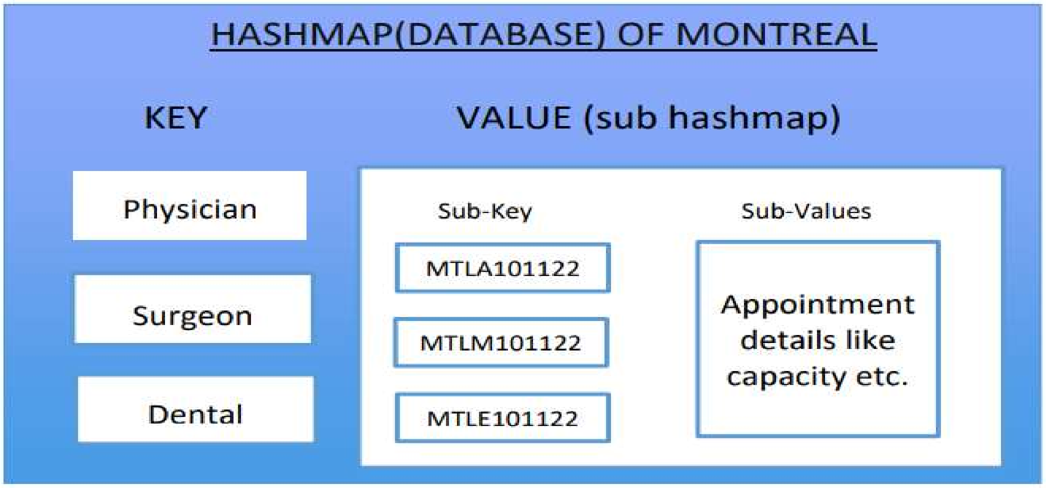


Fig. 1 Hashmap of a single hospital/city

System overview:

DAMS is designed in Java RMI. Java Remote Method invocation is mechanism to create distributed system. This mechanism allows an object to invoke methods on object running in another Java virtual machine[google].

In this assignment, RMI application and UDP are the major techniques. DAMS contains client which is connected to the to region wise servers. for example, If user is from Montreal then automatically Montreal server will be selected. Here, all servers are internally connected using the UDP. If Montreal’s user wants to book appointment of Quebec Hospitals, then Montreal server forward operation request to Quebec server. Quebec server perform that operation and send back the result to Montreal server.

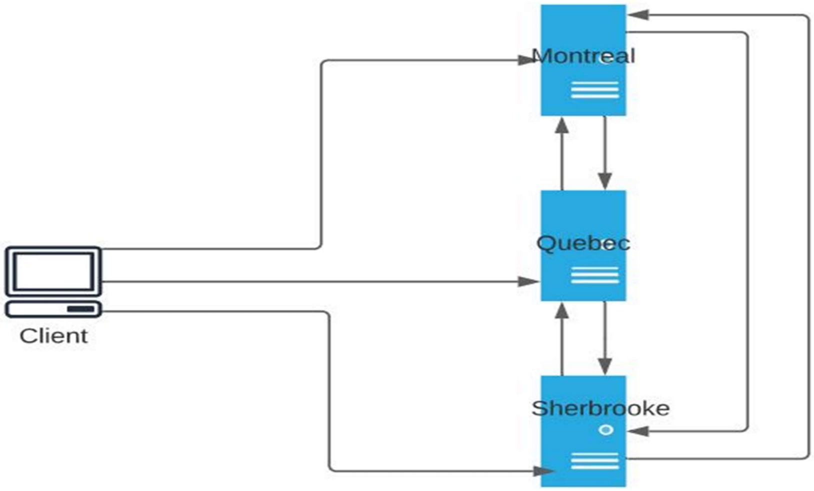


Fig.2 Server-client connection

As shown above, Server handles all request coming from the client and return a response. All server has their own HashMap to handle the data.

Flow of system:

Client requests the remote object through the RMI lookup. RMI Registry returns Server Stub. Now client invoke methods in the remote object. Remote object returns the response to client. [Source Transformation for Concurrency Analysis - Scientific Figure on ResearchGate]

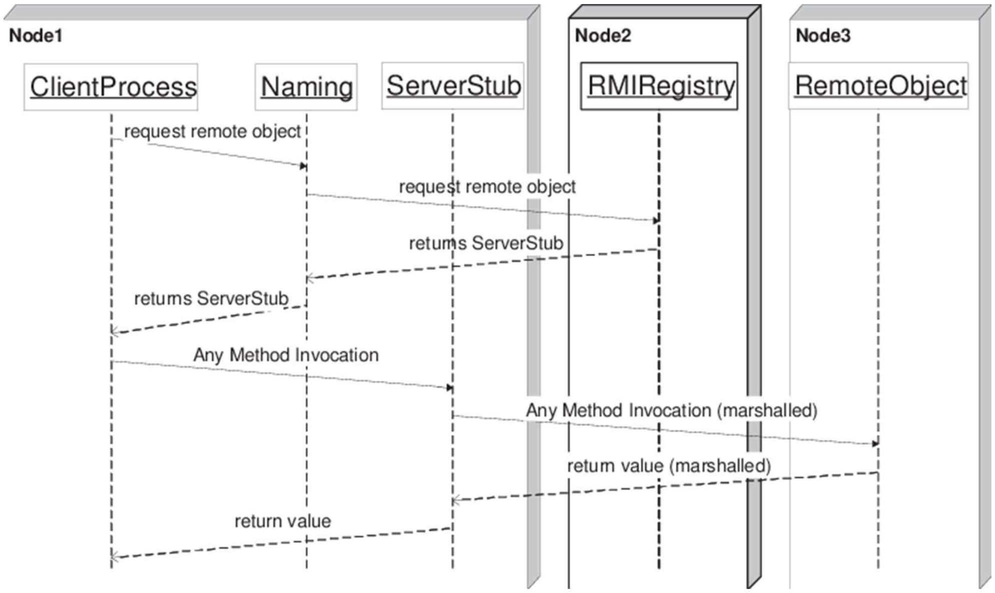


Fig.3 Flow of system

Test Cases:

Num	USER	OPERATION	TEST CASES
1	Admin/Patient	Login	1. Validate with valid UserID 2. UserID wise menus 3. Auto server selection from UserID
2	Admin	Add Items	1.Verify Appointment type 2.Verify Appointment id 3. Verify Appointment capacity 4. It will add a new item or change the capacity of current appointment
3	Admin	Remove items	1.Verify Appointment Id 2.Verfiy Appointment type 3.It will remove all appointments capacity
4	Admin/Patient	List of Appointments	1.It will display the all appointments from all servers.
5	Patient/Admin	Book Appointment	1. Verify AppointmentID 2. verify AppointmentType 3. Book Appointment using inter server communication. 4.If appointment already book will display try again. 5.Verify user can book one appointment once only 6. Change the available appointment capacity if booked appointment successfully.
6	Patient/Admin	Get Appointment	1.verify user has booked appointment 2.if not it will shows null 3.If appointment is booked then display list of appointments booked 4.Inter server communication
7	Patient/Admin	Cancel Appointment	1.check if appointment is booked by user or not 2.if booked, verify appointment id and appointment type then cancel the appointment 3.if user cancels the appointment capacity is updating. 4.If not booked, gives message to book appointment first. 5.Inter server communication
8	Patient/Admin	Logout	1.It will logout the user and ask for username to login