Distributed System Design

COMP 6231: Winter ’19

Instructor: R. Jayakumar

Distributed Library Management System (DLMS)

Using JAVA RMI and UDP

Documented By:
Name: Shresthi Garg
Student ID: 40105918

Overview:

The Distributed Library Management System (DLMS), aims to portray a system that is used by the Managers to maintain Books' records collectively and Users to access different library book and perform various action. The system allows the following two roles and their specific function:

Manager Role:

The operations that can be performed by a manager are the following:

- Add a book / Increase the quantity of an existing book
- Remove a book/ Reduce the quantity of an existing book
- List all the books available in his associated Library

User Role:

The operations that can be performed by a user are the following:

- Borrow a book
- Find a book with its name
- Return the borrowed book

The system provides the following functionalities/operations:

1) Server has 3 libraries viz. Concordia, Montreal and McGill.
2) Each server has 2 users i.e. 3X2 = 6 users in total
3) Each server has 2 manager 3X2 = 6 managers in total.
4) In total there are 7 books in different libraries.

Also, the system takes into account the following convention to distinguish managers and users of different library and the books in different library.

- **Library**:
  - Each of the three library has different acronym
    - Concordia - CON
    - Montreal - MON
    - McGill - MCG

- **Manager:**
  - Each Manager has a managerID which is a unique for each manager and constructed from the acronym of their library, a letter 'M' and a 4 digit number. eg CONM2233

- **User**:
  - Each User has a userID which is a unique for each user and constructed from the acronym of their library, a letter 'U' and a 4 digit number. eg CONU1234
- **Book:**

- Each Book has a itemID which is a unique for each book and constructed from the acronym of their library and a 4 digit number. eg CON1233

Description:

This System basically is designed to explain the entire functionality of the Java RMI. Below are the few points that should be known before understanding the project:

Definition - What does Java Remote Method Invocation (Java RMI) mean?

Java Remote Method Invocation (Java RMI) is a mechanism that allows one Java Virtual Machine (JVM) running object to invoke methods on an object running in another JVM. It facilitates the remote calling of Java object methods and sharing of resources and services.(Google).

Design Architecture

The system contains the following:
- One Client - It manages both the manager and user role.
- One Remote Interface - Defining method definition for the manager and user role.
- One Remote Interface Implementation - Defining method implementation for the manager and user.
- Three Server Implementation - Server Implementation specific to each library i.e., Concordia, Montreal, McGill.

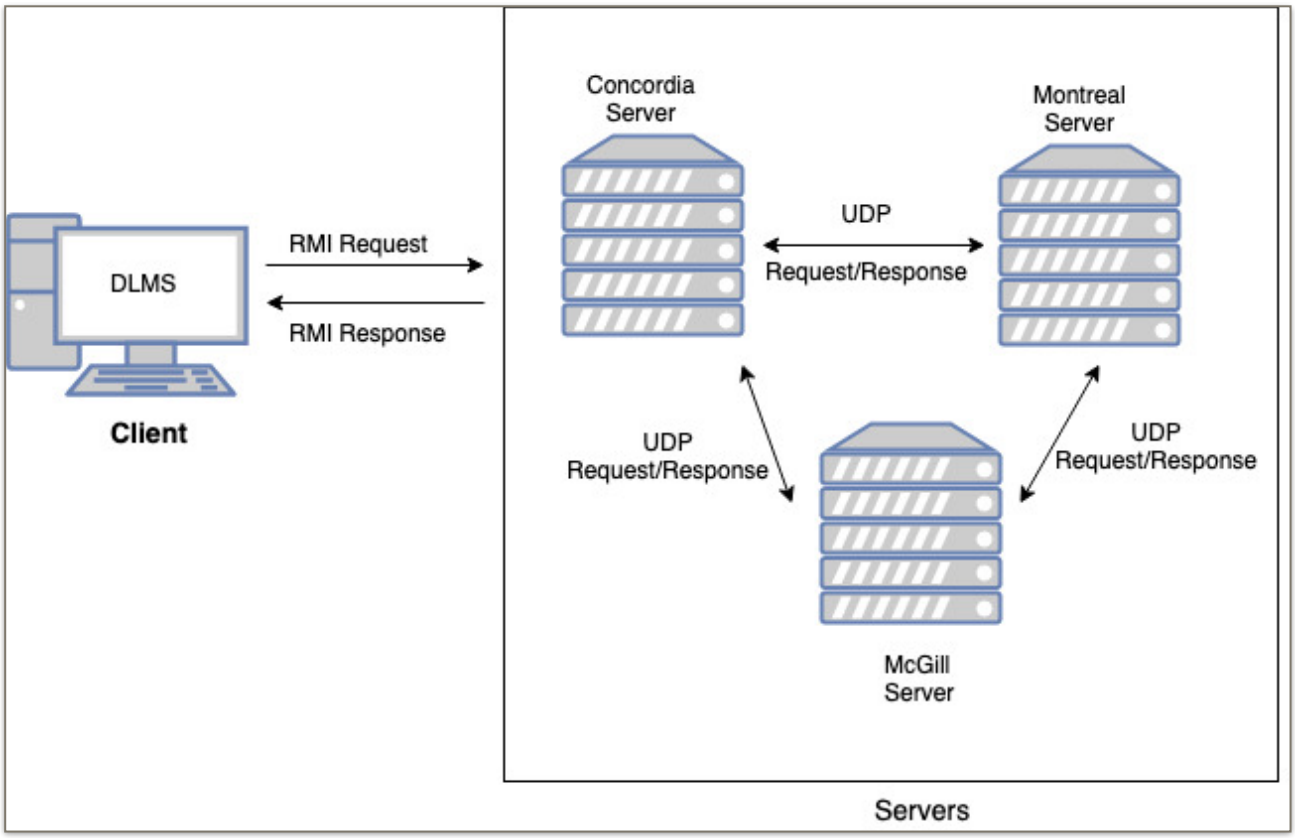

Fig: High level Architecture Diagram of DLMS

DLMS System Flow:

- System starts by running the servers which registers itself with the rmi registry and becomes active in the listening mode to receive request from the client.
- DLMS Application starts by running the client which presents a Welcome UI which prompts the end user to enter the Library ID.
- Based on the entered Library ID it distinguishes the user / manager and displays the functions with respect the role.
- Once the end user selects the required function, it makes a RMI(remote method invocation) request from the client, to the remote interface which in turn calls the remote implementation class to do the requisite function.
- This uses Java RMI to communicate between Client and Server.
- Once the request reaches the server, it performs inter server communication (if required) and returns the response from the called server to the calling server.
- The inter server communication in the above scenario is performed using UDP protocol.
- The server then returns the response back to the client as a RMI response.
- End user sees the desired response

Logging:

System maintains the following logs:

Client

It maintains separate logs for each Manager and User that logs in the system.Below are the details.

- Format: <UserID/ManagerID>.log
- Information: The logs records the following details:
  - Timestamp
  - Request Type and Parameters
  - Response received from Server

Server

It maintains separate logs for each Server (Library) in the system.

- Format: <Library>.log
- Information: The logs records the following details:
  - Timestamp
  - Request Type and Parameters
  - Request completed successfully or failed.
  - Response received from Server.

Test Scenarios covered:

Apart from the basic scenarios have covered the following test scenarios:

User:

- User can borrow multiple items from its own library but only one from each of the other library.
- User cannot borrow the same item again.
- User cannot opt to be the waitlist of the same book again.
- User can only return borrowed by him.

Manager

- Manager cannot perform user's function and vice versa.
- Manager cannot add a book of another library.
- Manager cannot access another library.