

# A Multipurpose Online Reservation System In Distributed Environment Using Middleware Architecture

Mohammad Badrul Alam Miah, Md. Abu Zakir Rizvi, Md. Ahsan Habib, Khandaker Hamidul Haque

**Abstract:** - The world is becoming smaller to smaller. People want various services at a time. For their expectation and choices, multi-server based services are provided. The proposed system is an online reservation system in middleware architecture distributed environment by using Java RMI and Server Socket. This reservation system contains several resource managers (RM), a middleware server and interactive clients. The middleware server is put between the resource managers and the clients. The clients can't interact directly with the resource managers. Instead, the clients interact with middleware server, which provides them a unique interface. The middleware server manages all the interaction with the resource managers to provide clients' requirements. The other ultimate goal of this reservation system which has been implemented is that to develop a multiple client, multiple server system. For handling multiple clients, this manages the middleware server such that it can receive concurrent requests from different clients and the structure does not damaged during the concurrent execution. The performance of the system is tested in different server and the experimental result shows satisfactory performance.

**Index Terms:** - Distributed System, Distributed System and Parallel Processing, Middleware, Online Reservation System, Parallel Processing, RMI Technology, RMI Technology with Middleware Server.

## 1. INTRODUCTION

The world is becoming smaller to smaller. People want various services at a time. For their expectation and choices, multi-server based services are provided. This proposed system provides three types of services at a time. It helps any client to go home/abroad with aircraft, home and hotel facilities with their belief. It gives their necessary information and integrating 3 services. Any client don't have hesitation after reaching there, how much distance he have to go for hotel, how he will go there, which type of hotel services he will get. The proposed system is fully based on an online reservation system. Various types of related system exist in the world. Bruno P. S. Rocha, Cristiano G. Rezende and Antonio A. F. Loureiro have done "Middleware for multi-client and multi-server mobile applications"[8], Mark D. Hanes, Stanley C.

Ahalt, Ashok K. Krishnamurthy have done "A Comparison of Java RMI, CORBA, and A Comparison of Java RMI, CORBA, and Web Services Technologies for Web Services Technologies for Distributed SIP Applications Distributed SIP Applications"[9], Denis Reilly have performed "A Dynamic Middleware-based Instrumentation Framework to Assist the Understanding of Distributed Applications" [10] and Seungwoo Son, Injoong Yoom and Changkap Kip have performed "A Component-based Client/Server Application Development Environment using java"[11]. But this proposed system gives better performance and security.

## 2. BACKGROUND AND MOTIVATION

Java-based middleware servers in particular, are rapidly gaining importance as a new class of workload for commercial multiprocessor servers. The middleware acts like a gateway for all clients to resource managers. This gateway process helps the system to control the access of clients to Resource Managers. Middleware technology helps greatly to implement distributed coordinated and scalable system. In a single server system, all pressure goes through the server [6], [7]. As a result, it increases the possibility of causing sever failure. It also reduces the ability to increase the scale the size of server as the clients increase by the time. Using multi-server system, server load is reduced by distributing the load among other server and that's the way to improve the server related major issues like failure possibility, scalability etc. A middleware based three-tier architectural system provides a more easy way to manage clients' request to different RMs/servers and to giving output back to clients from server. Ultimately, this approach provides benefits such as reusability, flexibility, manageability, maintainability, and scalability [1], [2], [3].

- 
- **Mohammad Badrul Alam Miah** is currently teaching at the Dept. of Information and Communication Technology as a full-time Assistant Professor in Mawlana Bhashani Science and Technology University(MBSTU).  
Email: [badrul\\_ict05@yahoo.com](mailto:badrul_ict05@yahoo.com)
  - **Md. Abu Zakir Rizvi** is a student at the Dept. of Information and Communication Technology in Mawlana Bhashani Science and Technology University. Email: [rizvi3389ict@gmail.com](mailto:rizvi3389ict@gmail.com)
  - **Md. Ahsan habib** is currently teaching at the Dept. of Information and Communication Technology as a full-time Assistant Professor in Mawlana Bhashani Science and Technology University.  
Email: [taregiut@yahoo.com](mailto:taregiut@yahoo.com)
  - **Khandaker Hamidul Haque** is a student at the Dept. of Information and Communication Technology in Mawlana Bhashani Science and Technology University. Email: [hshatu@yahoo.com](mailto:hshatu@yahoo.com)

### 3. METHODOLOGY/THE PROPOSED SYSTEM

The overall system architecture is shown in Fig 1. It provides a multiple server multiple client based architecture. One client can get triple services at a time. This proposed system can provide single, double or triple service for a large number of people at a time. It helps people by time consuming.

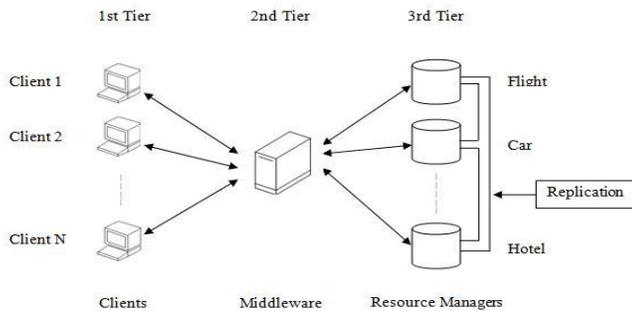


Fig. 1: Online Reservation System

RMI is a higher level abstraction than servers. In client-server computing, we typically develop an application level protocol to communicate between Java clients and servers, but with RMI we do not need to do this as RMI takes care of communication details for us. Using RMI as simple as invoking a method of an object [1], [4]. A client RMI call invokes the client-side stub (the proxy of the remote method that resides on the client's machine). The stub uses Object Serialization to marshal the arguments, i.e., render argument object values into a stream of bytes that can be transmitted over a network. The stub then passes control to the Java Virtual Machine's RMI layer. The skeleton on the server side dispatches the call to the actual remote object after unmarshal the arguments into variables in memory. The stub and skeleton programs are generated by the rmi c compiler. The Remote Reference layer permits various protocols for remote invocation, such as unicast point-to-point (the one currently has been implemented). Before a remote object can be accessed, it has to be registered into the naming server. The RMI framework provides a simple naming service [4]. Remote objects can register to the naming server using the java.rmi.Naming class with a URL-like naming scheme [1].

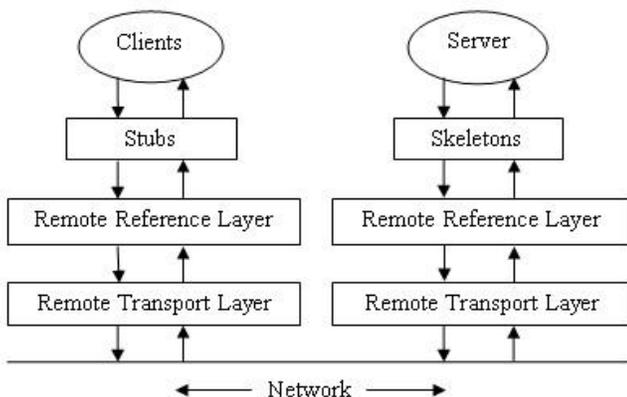


Fig. 2: Architecture of RMI [1]

Middleware is reusable software that resides between applications and the underlying operating system, network protocol stacks, and hardware. The purpose of using middleware is to isolate the application from the platform specific differences, both hardware and software, and provide facilities to hide the undesirable aspects of distribution. Properly developed and deployed middleware can reduce the task of developing distributed applications and systems by helping to [5]. It provides a set of capabilities closer to the application design level abstractions to simplify the development of distributed applications. It manages system resources by using higher-levels of abstractions. It avoids the use of the low level, tedious and error-prone platform details. It reduces system-lifecycle costs by building trusted reusable software patterns. It provides a wide array of the ready to use services for developers. It helps to ease the integration and interoperability of software over diverse heterogeneous and separated environments. It provides industry-wide standards for the higher levels abstraction of portable software[5].

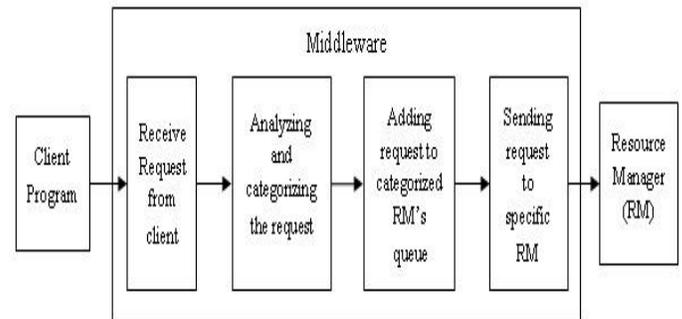


Fig. 3: Block Diagram of Sending Request.

### 4 EXPERIMENTAL RESULTS AND DISCUSSION

The proposed system has tested in lab environment for booking, withdrawal and searching query for flight, car and hotel properly. The performance rates are given below.

Table 1: Booking rate of the system for client.

Server	No. of client at a time	Success	Correct Booking Rate	Error Rate
Flight	40	38	95%	5%
Car	40	37	92.5%	7.5%
Room	40	37	92.5%	7.5%
<b>Total=</b>	<b>120</b>	<b>112</b>	<b>93.33%</b>	<b>6.67%</b>

Table 2: Withdraw of the system for client.

Server	No. of client at a time	No. of Success	Correct Withdraw Rate	Error Rate
Flight	40	37	92.5%	7.5%
Car	40	38	95%	5%
Room	40	36	90%	10%
<b>Total=</b>	<b>120</b>	<b>111</b>	<b>92.5%</b>	<b>7.5%</b>

**Table 3: Searching for availability and pricing of the system.**

Server	No. of client at a time	No. of Success	Correct Search Rate	Error Rate
Flight	40	38	95%	5%
Car	40	38	95%	5%
Room	40	36	90%	10%
<b>Total=</b>	<b>120</b>	<b>112</b>	<b>93.33%</b>	<b>6.67%</b>

**Table 4: Overall Performance of the System.**

Provision	Performance Rate	Error Rate
Booking Rate	93.33%	6.67%
Withdrawal Rate	92.5%	7.5%
Search Rate	93.33%	6.67%
<b>Average=</b>	<b>93.05%</b>	<b>6.95%</b>

Here for each type of experiment, the trial number was 360 times. It evaluates performance rate and error rate of the system. The following equation is used for evaluate performance rate.

$$\text{Performance rate} = (\text{No. of success}/\text{total no. of trial}) * 100 = (37/40) * 100 = 92.5\%$$

And find the error rate by using the following equation.

$$\text{Error rate} = (\text{No. of failure}/\text{total no. of trial}) * 100 = (3/40) * 100 = 7.5\%$$

## 5. IMPLEMENTATION

The proposed system has been implemented using JAVA, RMI technology and MySQL. JAVA is a powerful language with higher efficiency and a huge type of features and technology for computer networking, cloud computing and so on. RMI feature of JAVA is highly performable technology for multiple-instruction multiple data-stream (MIMD). RMI is highly used framework for cloud computing, Grid computing, Parallel Processing and distributed system. This is the main theme of this proposed system. Middleware based architecture is used for high quality service and other supports. Active Replication is used for fault tolerance. Proposed system can invoke/call method for various reservation services and get response quickly using JAVA RMI framework. This proposed system gives better performance because of using active replication system which is highly used all over the world. MySQL is open source database server which can perform up to 8GB memory space with efficient and better performance.

## 6. CONCLUSION

The great achievement of this project is finally the online reservation system is completely implemented with 93.05% correctness. By this project now it is possible for controlling and integrating multipurpose servers as their needful. In future work the fault tolerance algorithm will be updated to reduce error rate for better service.

## REFERENCES

- [1] Jia Weijia and Zhou Wanlei, "Distributed Network Systems from Concepts to Implementations", 1st edition Springer Science+ Business Media, Inc. ISBN: 0-387-23840-9, 2005.
- [2] Tanenbaum Andrew S., "Distributed Operating Systems", 5th edition, Pearson Education, Inc. ISBN: 978-81-7758-179-9, 2008.
- [3] Coulouris George, Dollimore Jean and Kindberg Tim, "Distributed Systems Concepts and Design", Third Edition, Pearson Education Limited, ISBN: 7-111-11749-2, 2001.
- [4] Holzner Steven, "JAVA 6 PROGRAMMING BLACK BOOK", 6th Edition, Kogent Solution Inc., ISBN: 9788177227369, 2007
- [5] Alrahmawy Mohammed Fathi, "Design Patterns and Component Framework for Building RTSJ-Based Real-Time Middleware", September 2010.
- [6] Macmillan, I., "Training: Structuring Distributed Real-time Systems." Copyright 1995 Architecture Projects Management Limited, 3rd October, 1995.
- [7] Wang Nanbor, Gill Christopher D., Schmidt Douglas C., Gokhale Aniruddha, Natarajan Balachandran, Loyall Joseph P., Schantz Richard E. and Rodrigues Craig "Middleware for communications QoS-enabled middleware" Copyright 2001 John Wiley & Sons, Ltd, 2003.
- [8] Rocha Bruno P. S., Rezende Cristiano G. and Loureiro Antonio A. F. "Middleware for multi-client and multi-server mobile applications".
- [9] Hanes Mark D., Ahalt Stanley C. and Krishnamurthy Ashok K. "A Comparison of Java RMI, CORBA, and A Comparison of Java RMI, CORBA, and Web Services Technologies for Web Services Technologies for Distributed SIP Applications Distributed SIP Applications", 2002
- [10] Reilly Denis, "A Dynamic Middleware-based Instrumentation Framework to Assist the Understanding of Distributed Applications".
- [11] Son Seungwoo, Yoom Injoong and Kip Changkap, "A Component-based Client/Server Application Development Environment using java".