

Implementation of Distributed Programming Environment to support Distributed Object Over Internet

Priyanka Patil¹, Vidhi Vadgama², Dushali Khadke³, Manoj Ubhale⁴

UG Students, Department of Computer Engineering, SSBT's College of Engineering and Technology, North Maharashtra University, Jalgaon, Maharashtra, India^{1, 2, 3, 4}

Abstract: With the explosive growth of internet and network services, there is always a proliferation for distributed application that seeks to leverage the power of the internet. Remote method invocation is now increasingly being used in Internet based applications and hence it is important to study the performance parameters of RMI. RMI is the action of invoking a method of a remote interface on a remote object. The three methods of RMI namely General, Activation and Custom Socket Factory are to be evaluated empirically using parameters like Round trip time, Latency and Packets per Unit time. The graph plotted allows us to gain an insight into the performance aspects and other tradeoffs of RMI. The General method deals with invoking any method directly from memory of the remote machine. RMI Activation allows passive objects to be brought into the active state by instantiating it on an as needed basis. Custom socket factory method allows customization of socket depending on the type and amount of data to be transferred over the channel. Search mechanism is also to be implemented in the GUI, which has user-friendly access to the three methods of RMI and displays the results of the evaluation.

Keywords: Compactness, Proliferation, RMI.

I. INTRODUCTION

Data uncertainty is common in emerging applications. Data uncertainty can be caused by various factors including measurements precision limitation. It is inherited in various applications due to different reasons such as outdated sources and imprecise measurements. Now a day, huge amount of historical data handling is very important. The data mining is very effective tool for extracting knowledge from historical information. Classification is one of the most effective techniques in data mining to handle an uncertain data. Classification means separation or ordering of objects into classes. The classification involves three data mining techniques the Naive Bayes, the Neural Networks and the Decision Tree algorithms. 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. Some projects have addressed the problem of object caching, but the proposed solutions are not targeted to the Java environment. The objective is to provide hybrid and adaptive caching, which manages both page caches and object caches, according to an object's behaviour. Other environments, such as the Object Store database management system, use object caching to meet scalability requirements. Javanese is distinguished from these

environments by providing an RMI-based solution that relies on cluster caching for general-purpose applications.

Motivation

As a result, Java-based middleware servers 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.

II. LITERATURE SURVEY

The content of the paper focuses on the research and contributions of various sources. These include:

[1]This paper describes the basic concepts of RMI Interface to provide web services. Many distributed programming environments require distribution or sharing of objects over Internet. Java Remote Method Invocation support client server applications where distributed objects reside on servers, which execute all methods invoked on the objects. Remote Method Invocation provides object caching and the local access it provides.

[2]This paper describes the basic concepts of client server communication by using web services. Web service like C, C++, JAVA, C#, .NET program execution as per client request.

[3]The paper describes the basic concept of program compilation and successful execution of program. Paper also describes the details of how client request for web service through Remote Method Invocation Interface and get web service.

III. PROPOSED SYSTEM

It is proposed to evaluate the performance of the three methods (General, Activation and Custom Socket Factory) by transferring different types of files such as text, image and video .These files are transferred using the three invocation methods and their performance is evaluated by analyzing the parameters: round trip time, latency and transmission rate. The observations are tabulated and graphs are drawn to facilitate the comparison of the three methods of remote invocation. This offers the user an insight into the best suitable method.

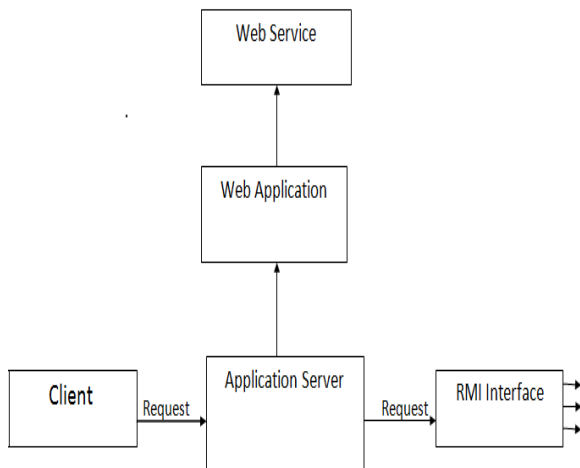


Figure1: Architecture of Proposed System

A Search mechanism is proposed to be implemented to locate methods when the host does not know where the method actually resides. This system is categorized into Keyword based search and Extension based search. A user friendly GUI that will hide the low level details is to be developed which will provide easier access to the three methods of invocation and will display the results of the actions performed.

Problem Definition

Collaborative and remote applications provide sharing of distributed resources and services on demand. Object caching is critical to distributed applications, especially over the Internet, where latency and bandwidth are highly variable. We have developed a configurable and efficient remote method invocation mechanism that provides the same interface as Java RMI, while extending its functionality so that shared objects can be cached on the accessing nodes.

IV. IMPLEMENTATION

Implementation of the proposed system involves the environment in which the system is implemented and the overall system development. The overall development of the proposed system requires suitable environment and proper resources for its successful completion. The proposed system is developed for a client-server communication. At the server, all language compilers are installed. The server contains Java Compiler, C Compiler, CPP Compiler and C# Compiler.

Model has been used to develop the web application in which View is represented by java server pages, Controller is represented by Servlet and Model is represented by java beans. MySQL is the database management system used for controlling the database. The break down structure of system is given below and also detailed implementation of each module is given.

Test case ID	Test Case Name	Test Case Description	Step	Test Data	Expected Result	Actual Result	Test Result (Pass/Fail)
1	Password	To verify password	Enter the password	Abcd/1234	Correct password	Correct/Incorrect password	Pass/Fail
2	Connection Establish	To verify connection	Change the connection	Data stored	Accepted	Accepted	Pass
3	C Program Execution	To verify code	Enter the C program	C program	Correct/Incorrect code	Correct/Incorrect code	Pass/Fail
4	C++ Program Execution	To verify code	Enter the C++ program	C++ program	Correct/Incorrect code	Correct/Incorrect code	Pass/Fail
5	JAVA Program Execution	To verify code	Enter the JAVA program	JAVA program	Correct/Incorrect code	Correct/Incorrect code	Pass/Fail
6	C#.net Program Execution	To verify code	Enter the C#.net program	C#.net program	Correct/Incorrect code	Correct/Incorrect code	Pass/Fail

Fig: Test cases

The above figure shows the test cases involved in our project and results of that cases.

(A) Breakdown Structure (modules):-

- 1) Designing of GUI: Description of AWT, Swing and code of Some Method.
- 2) Implementing Business Logic: Description of Core java and code of Some Method.
- 3) Implementing Event Handling: Description of Event Handling and code of Some Method.
- 4) Design for Database: Description of JDBC and code of Some Method from Database Connection. Java class Tomcat comes from the higher level. Both Bottom-up and Top-down integration tests were carried out.

(B) Server, Client Browser:-

After selecting create account or login operation, separate windows for both gets displayed on screen. It will show that the connection between client and server gets successfully created. Otherwise connection gets failed.

(C)Tomcat server, Client Browser and Authentication:-

User can create account, login and logout after integrating above modules. Information regarding new user stored into the database.

- 1) Input-: Creation of user account.
- 2) Output-: information regarding user get stored into database and username, password.
- 3) Validation-: user get successfully login.

- 4) Input:- User login name and password.
 - 5) Output:- Profile information regarding user get displayed in the browser.
 - 6) Validation:- User can successfully interact with system.
- (D)Tomcat server, Client Browser, Authentication, accessing Profile:-
- 1) Input -: upload file.
 - 2) Output -: File gets stored into the database.
 - 3) Validation:- File gets stored into database in a specified directory with same name as uploaded file.
 - 4) Input -: Download file.
 - 5) Output:- previously uploaded file.
 - 6) Validation:- The retrieved file will store on local directory and its contents must exactly equal to uploaded file.
 - 7) Input -: delete files.
 - 8) Results were obtained
 - 9) Output -: Files blocks get deleted from database.
 - 10) Validation:- Information about File block will delete from database.
- (E)User Classes and Characteristics:-
- 1) Main Controller- it is used to control all the request coming from users. It forwards the request to different beans for processing.
 - 2) Login Bean-it is used to logging in and validate user.
 - 3) Register Bean-It is used for registration of user.
 - 4) CCompile-It is used to compile C Programs.
 - 5) CPPCompile-It is used to compile CPP Programs.
 - 6) JavaCompile-It is used to compile Java Programs.
 - 7) DotNetCompile-It is used to compile c#.net Programs.
 - 8) DatabaseConnection-it is used to get connected to database.

V. RESULTS

The Proposed System includes User Registration Form. It consists of UserId and Password. Once User is authenticated, new window is open which consist of different options for Programming Language. The proposed system provides better reliability, integrity and security as compared to the existing system.

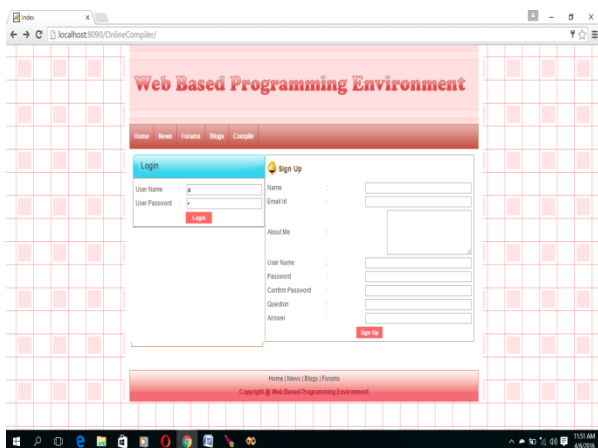


Fig 1: Registration

In above figure: 1, there is a registration process is to be done to provide web service as per client request.

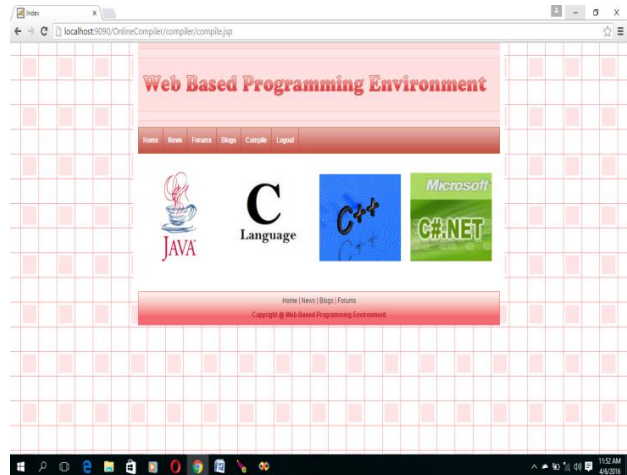


Fig 2: Web Services

In figure:2,web based programming environment is provided for the execution of JAVA,C,C++ and C#.NET programming languages.



Fig 3:Java Program Execution.

In figure: 3, there is a process for JAVA program compilation and execution.

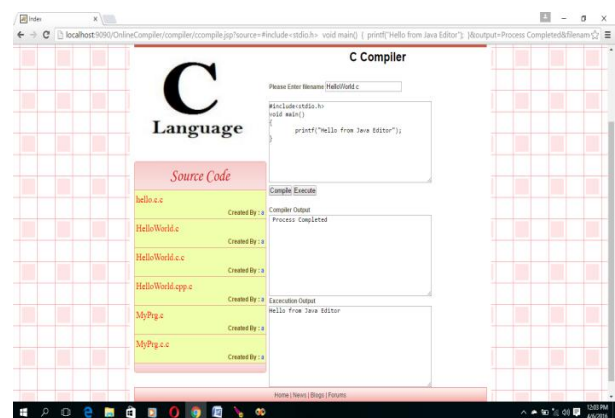


Fig 4: C Language execution.

In figure: 4, there is a process for JAVA program compilation and execution.

The same process is followed for C++ and C#.NET program compilation and execution.

VI. CONCLUSION

The goal of our project was to deliver a user-friendly product that will allow comparison of RMI methods. This has been achieved by evaluating the three Methods based on the various parameters such as round trip time, latency and transmission rate. As per the analysis done, The General method allows for faster retrieval of data than the other methods. The remote objects reside directly on the server side memory resulting in lower round trip times. The Activation method, though having higher round trip times, overcomes the memory constraint faced in the General method, by instantiating objects only on request. In the Custom socket factory method application specific sockets are created to allow implementation of security algorithms. The encryption and decryption thus performed result in an increase in the values of round trip time. The user interface was created using Java Swing allowing easier access to the invocation methods and search mechanism. It also displays the list of files on the remote and local hosts and the results of the evaluations performed.

REFERENCES

- [1] Daniel Hagimont and Fabienne Boyer” A Configurable RMI Mechanism for Sharing Distributed Java Objects” JANUARY FEBRUARY 2001.IEEE INTERNET COMPUTING.
- [2] Ahuja and Renato Quintao,,” Performance Evaluation of Java RMI: A Distributed Object Architecture for Internet Based Applications” 2000 IEEE.
- [3] Charles P.Pfleeger, Shari Lawrence Pfleeger., “Security in Computing”.
- [4] Cornell G., “Distributed Systems, Concepts and Design”, Addison-Wesley Publishing,1994.
- [5] Farley J.,”Java Distributed Computing”, O’Reilly Publishing, 1998.
- [6] Matt Welsh.: NinjaRMI : “A Free Java RMI” . September 1999.
- [7] Patrick Naughton Herbert Schildt., “The complete reference JAVA 2”.