REMOTE NETWORK MONITORING

Abstract:

Remote monitoring is real time monitoring tool for windows network and analysis through Images by throughput in the network. Here in this project the main concept is to monitor the client system. Through this we can access the client easily and can view the details open by the client in the client system. We can use to monitor the system when client access any unwanted we can monitor it easily.

For example in college if student try to access any unwanted application or websites means using this Remote monitoring administrator can access the student pc in the sitting place itself .

Here TCP protocol is used to implement this remote monitoring system. Access of client pc is done by the server. This remote monitoring is used in real time companies too.

Software Used:

Front End - JAVA

.

MODULES:

- **INITIALIZE SERVER.**
- **CONNECT CLIENT.**
- **CAPTURE THE CLIENT SCREEN AND TRANSFER.**
- **SERVER VIEWS THE CLIENT MONITER.**

INITIALIZE SERVER

Run the server side program **server initiator**. In server program server has to give port number. Port number given by the server, the same port number has to be given by the client.

CONNECT CLIENT

To connect client with server the client has to give the ip address of the server first, then client has to give the port number given by the server side. If both port number matches means Connection between client and server will be established.

CAPTURE THE CLIENT SCREEN AND TRANSFER

The capture of the client screen will be done continues and transfer to the server side. Through this transfer of client screen continuously the server can easily monitor the client screen. Application open by the client will be view by the server.

SERVER VIEWS THE CLIENT MONITER

In the server will be monitored by capture the client screen. The application open by the client will be monitored by the server. If client opens any unwanted application or web browser will be seen by the server directly. Server remote monitoring help them to monitor the client monitor in sitting place itself.

EXISTING SYSTEM:

In the Existing system the monitoring of Clients Pc is done only by directly viewing the client. Remote monitoring is not implemented here. Server find it difficult to monitor the client. In the existing system process is done in platform dependent.

PROPOSED SYSTEM:

In the Proposed system remote monitoring is done. Through the remote monitoring server can easily monitor the client in the sitting place itself in the server pc itself too. This process is done in platform independent.

TECHNOLOGY USED

Java

INTRODUCTION

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office) , was intended to replace C++, although the feature set better resembles that of Objective C.

Java, a platform independent programming language helps in building any kind of application of our interest; Java uses a compiler to convert the source code into architectural independent byte code. These are executed over a Java Virtual Machine (JVM), which is an idealized java processor chip usually implemented in software rather than hardware. Java was developed to include methods for Internet data manipulation. Java applications can be written once and run on any machine having a Java Virtual Machine as part of its operating system.

Features of Java

- Platform Independent
- Object-Oriented language
- Secure
- Portable
- Robust
- Image processing
- Web development support
- Supports Multithreading

SWING

Swing is a GUI toolkit for Java. **It is one part of the Java Foundation Classes (JFC).**Swing includes graphical user interface (GUI) widgets such as text boxes, buttons, split-panes, and tables. Swing widgets provide more sophisticated GUI components than the earlier Abstract Window Toolkit. Since they are written in pure Java, they run the same on all platforms, that is uniform behavior in all platform, unlike the AWT which Is tied to the underlying platform's windowing system.

Swing supports pluggable look and feel – not by using the native platform's facilities, but by roughly emulating them. This means you can get any supported look and feel on any platform. **Swing is a platform independent, Model-View-Controller GUI framework for Java. It follows a single-threaded programming model.**

Swing Features

- **Platform independence**
- **Extensibility**
- **Component-Oriented**
- **Customizable**
- **Configurable**
- **Lightweight UI**
- **Loosely-Coupled/MVC**
- **Look and feel**

REMOTE METHOD INVOCATION (RMI)

. Remote method invocation allows applications to call object methods located remotely, sharing resources and processing load across systems. Unlike other systems for remote execution which require that only simple data types or defined structures be passed to and from methods, RMI allows any Java object type to be used - even if the client or server has never encountered it before. RMI allows both client and server to dynamically load new object types as required. In this article, you'll learn more about RMI.

Remote Method Invocation (RMI) facilitates object function calls between Java Virtual Machines (JVMs). JVMs can be located on separate computers - yet one JVM can invoke methods belonging to an object stored in another JVM. Methods can even pass objects that a foreign virtual machine has never encountered before, allowing dynamic loading of new classes as required. This is a powerful feature

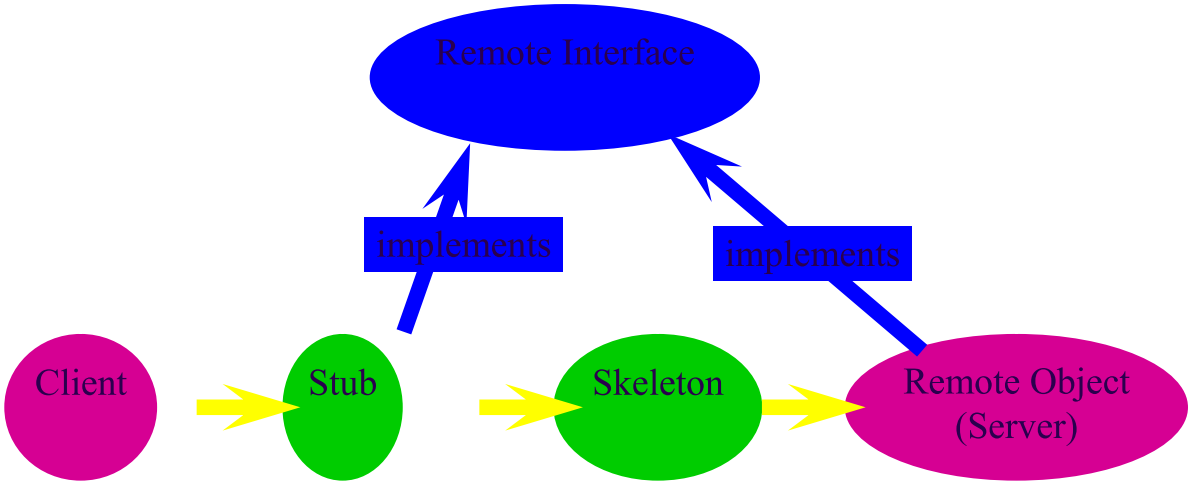
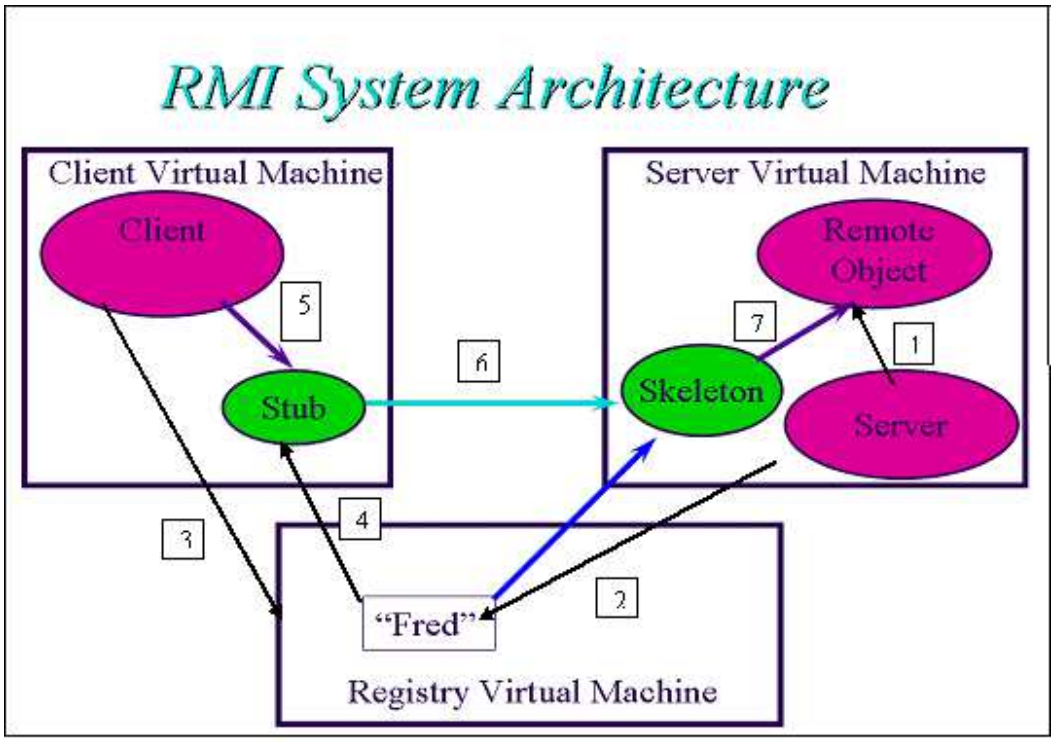


Figure 5.3.1 shows remote references and stubs

RMI Flow



RMI System architecture

1. Server Creates Remote Object
2. Server Registers Remote Object
3. Client requests object from Registry
4. Registry returns remote reference
   (And stub gets created)
5. Client invokes stub method
6. Stub talks to skeleton
7. Skeleton invokes remote object Method

Overview of the project

- Command Line
  - Run Command

    This enables the local user to open the required application in the remote machine.

  - Custom Batch

    This feature allows to create a batch which consists of few applications on the whole. We can create a maximum of 5 applications in a single batch. The execution of this batch opens all the applications that are created, simultaneously in the remote machine.

  - Clear Screen

    This feature helps the user to clear the current remote admin screen.

  - Clear Debug

    This feature helps the user to clear the Debug console area.

  - Login features

    The remote system can be restarted, shut down from the local system itself.

- File Browsers

This module provide platform to browse Files and directories of the remote system and perform different file operations. Those files available in the remote machine can be viewed from the local system itself. Those files and directories can be updated, deleted, edited, renamed, etc, also the local user can create new directories and also files for the directories in the remote machine.

- File Transfers

This module allows users to instant/delay Upload and download files in ASCII and Binary format. The files and the directories in the remote machine can be moved from the local machine to the remote system (i.e., uploading) or moved from the remote system to the local machine (i.e. downloading).

This transfer of files from either side can be done between different directories also. The transfer of files and directories can be done to move them within the same system but between the directories also.

- Connection Monitor

This module allows users to monitor the connection status of the remote systems. Restart/Refresh RMI connection between the systems is also done in this module.

SYSTEM IMPLEMENTATION

Server Side Installation

- Install the required software's to satisfy server configuration
- Copy the folder that contains server side programs to the system
- Set the class path
- Set the IP address of the client system in the RMIServer.java file
- Compile the Server side programs
- Run the RMIServer.class file for establishing RMI connection with the client connection.

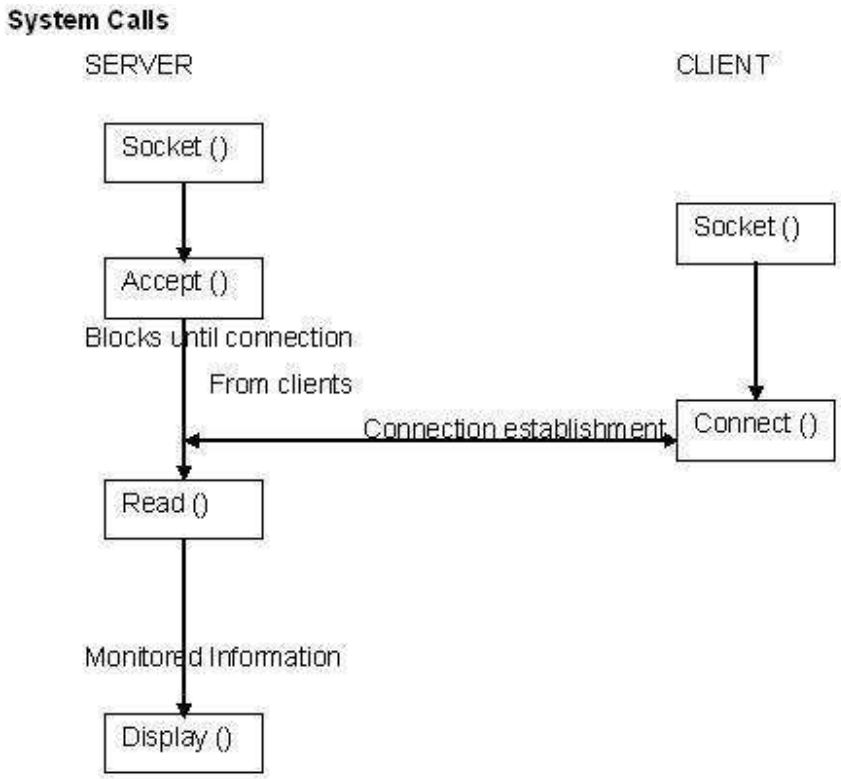
Client Side Installation

Install the required software's to satisfy client's configuration

- Copy the folder that contains client side programs into the system
- Set the class path

- Compile the client side programs
- Run the RMIClientGUI.class file for establishing RMI connection with the client connection.

6.3 Use Case Diagram



10. Development Environment

Server

Hardware Requirements

Processor : Intel Pentium III

RAM : 256 Mega Bytes

Hard Disk : 10 Giga Bytes

Operating System : Windows NT/2003 Server/XP, Linux

Software Requirements

Language : Java (j2sdk1.4.2_04)

Client

Hardware Requirements

Processor : Intel Pentium III

RAM : 128 Mega Bytes

Hard Disk : 10 Giga Bytes

Operating System : Windows NT/2003 Server/XP, Linux

Software Requirements

Language : Java (j2sdk1.4.2_04)

11. Implementation Environment

Server

Hardware Requirements

Processor : Intel Pentium III

RAM : 256 Mega Bytes

Hard Disk : 10 Giga Bytes

Operating System : Windows XP, Linux

Software Requirements

Language : Java (j2sdk1.4.2_04)

Client

Hardware Requirements

Processor : Intel Pentium III

RAM : 128 Mega Bytes

Hard Disk : 10 Giga Bytes

Operating System : Windows XP

Software Requirements

Language : Java (j2sdk1.4.2_04)

12. Conclusion

The project entitled "***Remote Network Monitoring"*** is developed and tested successfully. It meets the specified requirements. This feature of being independent of the remote system platform plays a good role in area where different OS is being used.

13.1 Future Enhancement

This tool can provide Application Programming Interface for programming plug-in and for integration with external system like mobile devices and embedded systems.

14. Bibliography

References

- William Grosso, "JAVA RMI", O'Reilly Media, Inc.; 1 edition, 2001

- Esmond Pitt and Kathleen McNiff, ”THE REMOTE METHOD INVOCATION GUIDE”, Tata McGraw Hill Publishing Company, 2001
- John Zukowski, “Definitive Guide to Swing for Java 2”,1999
- Tristan Richardson, Quentin Stafford-Freser,

  Kenneth R. Wood and Andy Hopper,”VIRTUAL NETWORK COMPUTING”
  Reprint from IEEE Internet Computing,
  Volume 2, Number 1
  January/February 1998
- Roger S.Pressman, “ SOFTWARE ENGINEERING –A PRACTIONERS APPROACH “, Tata McGraw Hill Publishing Company, 1997.

Websites

- www.java.sun.com
- www.samba.org
- www.remote-desktop-control.com