

# Server Controlled Mobile Agent

Mohd Haroon  
Research scholar  
TMU Moradabad, India.

Dr Mohd Husain  
Prof Dept of CSE  
AIET Lucknow

Manish Madhav Tripathi  
Asst prof dept(CSE)  
Integral university

Tameem Ahmad  
Integral university  
Lucknow

Vandana Kumari  
Integral University  
Lucknow

## ABSTRACT

Today's agent incarnations can be characterized in a number of ways ranging from simple distributed objects to highly organized software with embedded intelligence. Mobile agent technology offers a new computing paradigm in which a program, in the form of a software agent, can suspend its execution on a host computer, transfer itself to another agent-enabled host on the network, and resume execution on the new host [1]. This paper is concerned with summing up a number of functionalities like node accessing, remote administration, software installation, and load balancing in a wireless network from server in real-time systems. As the sophistication of mobile software has increased over time, to have the associated threats to security, server could be able to manage and operate all the remote nodes in his network within range of the Access Point. Dynamically we can select any mobile computer and manage, operate and instruct through announcer.

**Keywords:** Mobile agents, Node Accessing, Remote Administration, Load balancing.

## 1. INTRODUCTION

### 1.1 Problem Statement

A Wireless network has no fixed networking infrastructure. It consists of multiple mobile nodes which maintain network connectivity through wireless communication, and it is completely dynamic, so such networks are easily deployable. Therefore Efforts have been put up to manage and monitor these remote nodes and utilize to balance the load of the network.

### 1.2 Motivation

The widespread use of Laptops and wireless mobile devices has generated the need to provide services which can overlook the whole network. Keeping the need in mind we develop software with security features that increases the usability of software, whose application and implementation will be discussed later on in this paper.

### 1.3 Approach

To manage the network through server we have to firstly establish a wireless network using Access Point/Access Router/ADSL Router (an ad-hoc network) and have to select a server from the network over which the administrative program will execute, that monitors the overall network. On rest of the nodes a hidden program will be reside to response that server.

## 2. SOFTWARE FEATURES

This paper consists of a number of features listed below:

### 2.1 Nodes Management

Server can be able to manage its all nodes through following options:

#### 2.1.1 Remote Monitoring

Server can monitor each and every remote node in his wireless network and can take snap shots of each node which can be

continuous according to the need. Here we used the concept of serialize and deserialize memory stream. We used a class called snapshot to capture the screen of the client. We stored the screen data in a byte array then we created a memory stream and then we serialize the image using binary formatter and memory stream, for this we used the namespace called System.Runtime.Serialization.Formatters.Binary which is provided by the .net. And we converted the screen data into the .jpg image. This image can be showed or can be saved according to the need. And for continuous capturing of data we used timer and have specified the time interval. For security we are tranfering the screen data as an object not as the image.

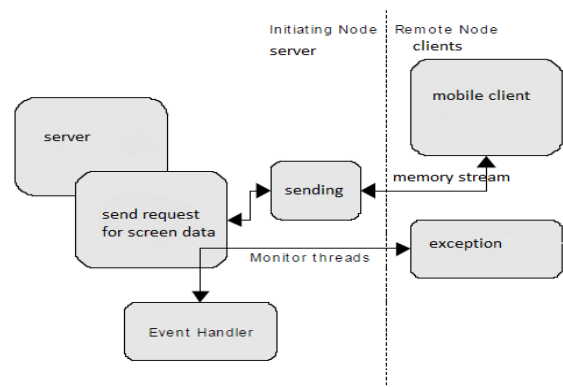


Figure 1: An overview of the node monitoring system

#### 2.1.2 Remote Software Installation

Here we are installing a software form server to client, but the limitation is that the software installation wizard should be a non GUI based. For this we have created software which captures the screen shot and saves the images for tutorial purpose. We have created a class called InstallFile which creates the constructor tc for TcpClient, and ns for NetworkStream which is used to get the stream form the TcpClient. Now after that we have BinaryReader br which is used to open the file (.exe) and setting its mode as open and FileAccess as Read. We have BinaryWriter sw which is used to write the stream in the network stream. Same method is applied to create the desktop icon at client site but instead of using the .exe file type we used app.ico type.

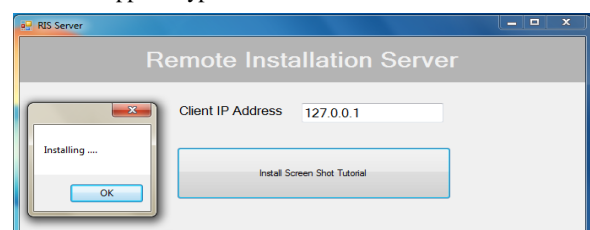


Figure 2: Screen Shot of Remote installation server

### 2.1.3 Remote Control

In remote control we have remote application invocation, remote shut down and restart. For this purpose we have used .net programming. Here one program is running on the server while the other one is running on the clients simultaneously.

This concept can be implemented using the namespace System.Net and System.Net.Sockets provided by the .net environment, i.e. socket programming and concept of multithreading.

Here we have used TcpListener (which listens to the request), Socket (to accept the socket), NetworkStream, StreamReader (to read the matter written on the network stream) and StreamWriter (to write on the network stream).

On clicking the buttons at server site it writes the name of process to be invoked at client site using StreamWriter and at the client site the program running there reads the StreamReader and invokes the remote applications on the client site using process.start.

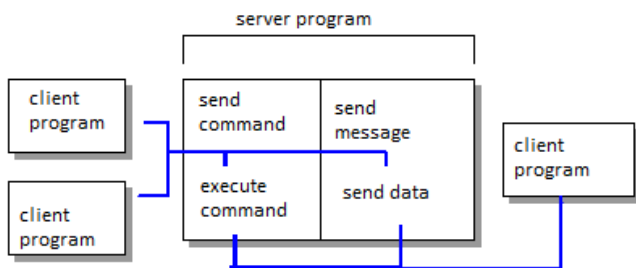


Figure 3: A diagram for remote control system

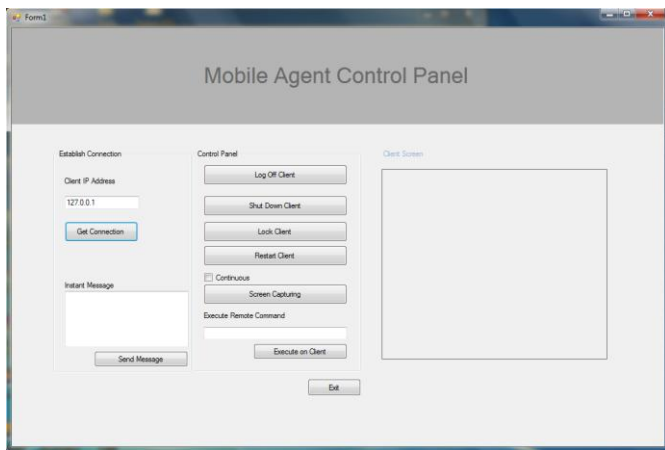


Figure 4: Screen shot showing control panel for remote control of clients

### 2.1.4 Announcer

This feature enables server to multicast/unicast messages to its clients. The messages are pop upped and appear in a message box and hence enable a server to instruct and pass message to its clients. The main concept used here is the multithreading, socket programming and windows API programming.

The Java Remote Method Invocation (RMI) application programming interface (API) enables client and server communications over the net. Typically, client programs send requests to a server program, and the server program responds to those requests [6].

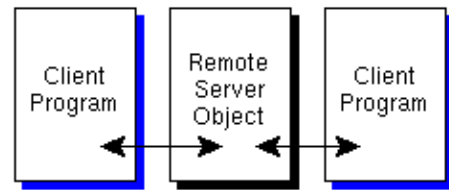


Figure 5: A diagram for RMI

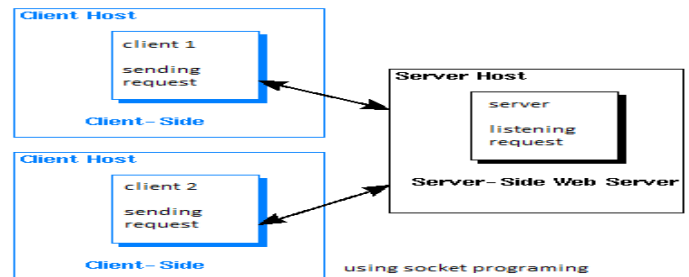


Figure 6: An overview of the announcer system

This concept can be implemented using the class System.Net and System.Net.Sockets provided by the .net environment.

Here we have created TcpListener (which listens to the request), Socket (to accept the socket), NetworkStream, StreamReader (to read the matter written on the network stream) and StreamWriter (to write on the network stream).

Here text box is provided where the administrator can write messages and can send it to its clients. On clicking the send button the message written on the text box gets write on the NetworkStream using StreamWriter and is send to the client site, which is there read by the StreamReader and is displayed in a message box.

## 2.2 LOAD BALANCING

Load balancing in terms of number of processes on the node and transferring of application load on least loaded node. For this we have calculated the process depth of each node. Each node calculates its process depth from time to time using process.GetProcess() and sends the information (process depth) to the server. Now server compares the process depth of each client and then determines the least loaded client and then redirect the job to the least loaded client i.e. whose process depth is minimum.

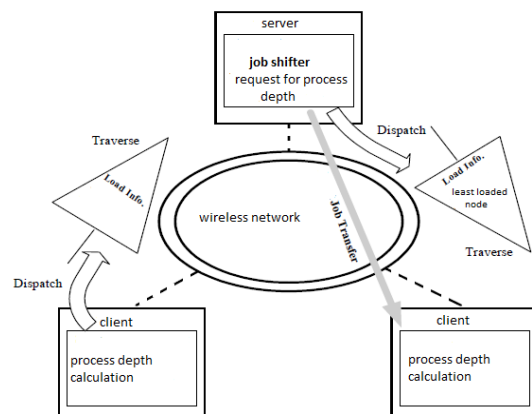


Figure 7: load balancing scheme for process execution

As the no. of nodes increases, the task for the load balancing may become quite difficult to manage at run time, in this case, in order to distribute the incoming load on participating nodes, we use mobile agent (MA). A value function which is sum of all three types of resources including CPU, Memory and I/O is considered. The MA chooses this value function as a trademark for load transfer. Different metrics are used to compare load balancing mechanism with the existing technology [3].

### 2.2.1 The Following Policies are Chosen for Mobile Agent Load Balancing-

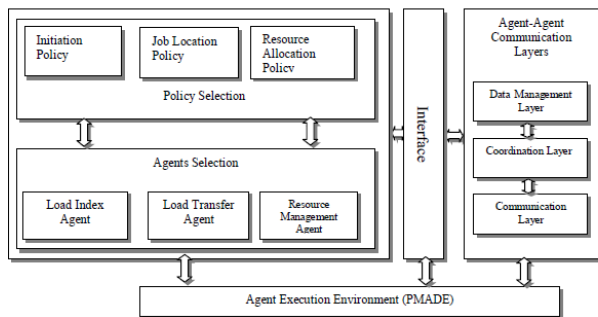


Figure 8. System Architecture for Load Balancing

**INITIATION POLICY** - It determines who starts the load balancing process. The process can be initiated by any node [3], in this research paper it is initiated by the server.

**JOB LOCATION POLICY** - It determines the location of job execution i.e. either of which client node [4]. If the enough resources are available for job execution at local site then it is executed at local node otherwise it is transferred at the global site by using MA. Suitable agent is chosen for job transfer [3, 4].

### 2.2.2 Agent Selection

The following agents are chosen for load balancing

**LOAD INDEX AGENT (LIA)** - Agent calculates the load index (LI) or process depth of each resource (CPU, Memory, I/O) on a particular client node. LI of CPU is the sum of remaining CPU lifetimes of job running on that node. LI of memory is the sum of page fault processing time of job on that node. Similarly LI of I/O is the sum of I/O processing times of jobs on that node. The sum of load index of each resource is the total VF for that node and is used for load transfer by the respective agent [3].

**LOAD TRANSFER AGENT (LTA)** - This agent is used for migration of job from one node to another. It executes two predefined policies namely –local and global. The policy is chosen according to the response time of job submitted for execution. If the response time of job at the local site is less than the global site then local policy is executed otherwise global policy is chosen [3, 4].

**RESOURCE MANAGEMENT AGENT (RMA)** - This agent is used for resource selection and allocation to the appropriate job in execution. Currently for the sake of simplicity we consider only three main resources in distributed system namely as CPU, Memory and I/O devices [3].

### 2.2.3 AGENT - AGENT COMMUNICATION LAYERS

Agents in the system communicate with each other via coordination and communicationLayers with users using mobile group approach for MAs. The request an agent receives from the communication layer should be explained and submitted to the coordination layer, which decides how the agent should act on the request according to its own knowledge. Agents communicate by exchanging messages through reliable communications channels,

i.e., transmitted messages are received uncorrupted and in the sequential sent order, as long as the message sender does not crash until the message is received [6]. Data Management Layers keep record of data associated with MAs [3].

The mobile-agent based approaches have the merits of high flexibility, low network traffic and high asynchrony [4].

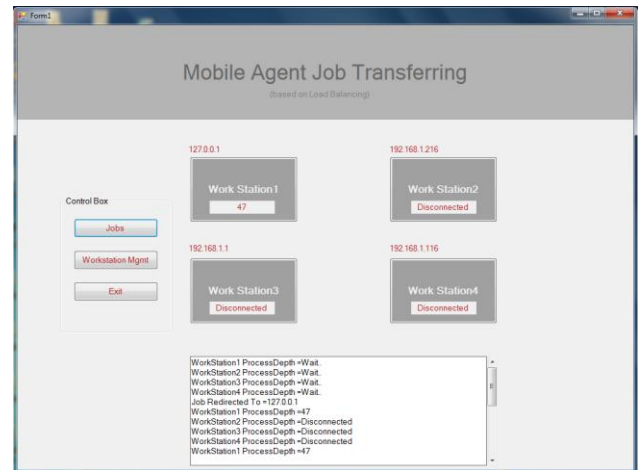


Figure 9: Screen shot of mobile Agent Job Transferring System

### 2.3 Security

Data center security continues to be a challenge as outside threats has become more sophisticated while, at the same time, the workforce has become more mobile. Devices at the edge of the network such as load balancers must provide security against outside threats such as denial of service attacks while authenticating valid users [5].

For security, server is password protected; the only nodes stored in the table maintained by the server can only connect to the server and not any other nodes.

A server has a privilege to add a node or to delete a node and to update the information of the client. Beside this for monitoring the client nodes the screen shot of client is transferred as an object and memory stream form client to the server. Which is where converted in the form of image.

In addition to this access point provide more security to it, as for connecting the node every node is require to provide the key only then it is connected to the server, it also checks MAC address of the client before a connection is made between the client and the server.

### 3. CONCLUSION

Our method works well for small network but as the no. of client increases, its performance decreases mainly in terms of load balancing. The method is simple and easy to handle, its complexity is low so it works faster. Simple and easy methods are used to monitor all the clients in terms of capturing the screen data of clients on regular interval, remote installation of softwares i.e. administrator can install the softwares directly to its clients through server, remote control which includes remote application invocation, remote execution of commands such as shut down, restart, lock etc. Announcer is used to multicast the messages to the clients.

Besides these, this paper is also concern with the incoming job transferring to the least loaded node i.e. system is redirecting the jobs on least loaded node using process depth computation; when

large no. of clients are used then more parameters can be added to select the least loaded client like memory utilization of clients etc.

#### **4. FUTURE ENHANCEMENT**

This project provides a basic idea to control the clients through the server; here we are using few clients at present which can be extended to n no of clients as per need. More no. of software can be installed in the client node by sitting on the server and can remove the limitation i.e. the installation wizard should not be a GUI based. For more security we can use complex cryptography encryption and decryption algorithms. In job transferring part we can apply the same concept and extend the project to shift the processes i.e. we can divide the processes and can send it to other clients for processing(in case of processing of bulk of data). A facility of chatting can be added as per requirement. We can use more parameters such as memory utilization to calculate the load of the client.

This research has been done on windowsXP, which can be extended to any operating system.

#### **5. REFERENCES**

- [1] Asoke K. Talukder, Roopa R Yavagal "Mobile Computing", Tata McGraw-Hill Publishing Company limited, New Delhi, 2009.
- [2] International Conference on Computational Intelligence and Multimedia Applications IEEE Computer Society 2007.
- [3] Neeraj Nehra, R.B.Patel, "Towards Dynamic Load Balancing in Heterogeneous Cluster using Mobile Agent", International Conference on Computational Intelligence and Multimedia Applications, 2007.
- [4] Jiannong Cao, Yudong Sun, Xianbin Wang, Sajal K. Das, "Scalable Load Balancing on Distributed Web Servers Using Mobile Agents".
- [5] LB System, "Load Balancing System Design", white paper, June 2008.
- [6] Raimundo, J., Macêdo A., Assis Silva F.M, "The mobile groups approach for the coordination of mobile agents. J. Parallel Distributed Computing", 2005, 65(1): 275-288.