Design and Implementation of Dynamic Network Task Management System

Arun S. Karjol
5thSem MCA, RVCE
Bangalore

Bharath R.V
5thSem MCA, RVCE
Bangalore

Dr. Anupama Kumar
Associate Professor
Department of MCA
R.V. College of Engineering

Abstract - Network task management system deals in managing the remote tasks to be carried by various client systems in the same network. The requested tasks by various clients are stored in the database with the IP Address, task name and time. The task requested by the client is completed when the server pings the requested client at the specified time. The server again checks for anymore tasks that are remaining to be performed. The database is updated after the tasks are completed. Once the requested tasks are completed the server keeps on examining for any other requests in the database. If the tasks are not performed because of any network issues like timeout then these tasks are again tried to get connected for five times if not it picks the next task to be performed, so that server should not be idle and increases the performance. Searching is provided in the project to verify the completed and incomplete tasks in the application. The system is implemented in C# on .Net framework with MySQL database. The application is built as a user friendly windows application. The client server connection is established dynamically using the IP address and credentials, the tasks are performed and acknowledged, the status of the tasks are updated in the server database and data base is refreshed for avoiding the overhead of the server.

Keywords: Client, Server, Task, remote, IP address

I. INTRODUCTION

Task management is the process of managing a task through its life cycle. The management involves planning, testing, tracking and reporting of the tasks in a proper manner [1]. The network task management system aims at establishing client server architecture between remote systems and facilitates system tasks among the clients. The tasks include opening/ performing of single/multiple tasks as requested by the client at other remote client locations. The server establishes a dynamic connection with the clients at the requested time, completes the task requested and the connection is terminated after the completion of the task. The data base is refreshed automatically once the task is completed to ensure faster access and remove system overhead.

The scope of the project lies in establishing dynamic Client-Server architecture. This is implemented with a scope of single-server and many clients with tasks that are done with multi-client/tasks architecture.

The project defines two types of clients:

(I) Clients which define the tasks and

(2) Remote clients which performs the tasks.

Automatic updating and restoring of the data is carried out to ensure elimination of system overheads.

The requested tasks by various clients are stored in the database with the IP Address, task name and time. The task requested by the client is completed when the server pings the requested client at the specified time. The server again checks for anymore tasks that are remaining to be performed. The database is updated after the tasks are completed. Once the requested tasks are completed the server keeps on examining for any other requests in the database. If the tasks are not performed because of any network issues like timeout then these tasks are again tried to get connected for five times if not it picks the next task to be performed, so that server should not be idle and increases the performance. Searching is provided in the project to verify the completed and incomplete tasks in the application.

II. LITERATURE SURVEY

Task management can help either individuals achieve goals, or groups of individuals collaborate and share knowledge for the accomplishment of collective goals. Tasks are also differentiated by complexity, from low to high. Task management may form part of project management and process management and can serve as the foundation for efficient work flow in an organization. Effective task management requires managing all aspects of a task, including its status, priority, time, human and financial resources assignments, recurrence, notifications and so on [9]. These can be combined together broadly into the basic activities of task management. Network task management system is implemented using client server architecture. A client server application is a piece of software that runs on a client computer and makes request to a remote server[2]. In this paper the client server architecture is implemented dynamically between a remote server and multiple remote clients. Synchronizing between the client and the server is a network task which is time consuming if not connected properly. In [3] the author has described the need of an IP network like TDM/SDH. He has also explained the communication between the client and the server using alternate routes when the network node is affected.

III. METHODOLOGY

This proposed system implementation involves a methodology which begins with the process of understanding the client Server Architecture and Connection establishment between the Client-Server models. The business logic for the requirements is implemented using C# .Net and MySQL Database. The project is implemented as three different modules and integrated together to make an application. The waterfall model has been followed to develop the system.

The module 1, Network Connection Module is designed to connect the client and server dynamically using the IP address and credentials. The requested task by various clients is stored in the database with the IP address, task name and time. To perform the task on client machine it takes the username and password of that machine for authentication. Whenever specified time is equal to server time, server picks all the necessary information from the database and performs the task.

The module 2, Performance module is designed to perform the tasks requested by the clients is completed and acknowledged by the server. The module has been developed using PSEXEC tool[4] to perform the specified task in the remote client machine. The server collects the tasks to be performed by the remote clients and keeps checking the database for upcoming task. At the specified time, it pings to the required client. The server pings the client for connection for a maximum of 5 times in case of a network failure and then reports to the requested client if it is not connected. In case of the connection made, the tasks will be completed. If two more tasks are requested by two different clients at the same time, the server pings the remote requested client machine and performs the task. If two different remote clients are requested to perform multiple tasks at the same time, the task are completed by the server. The server acknowledges to the client machine regarding the task completion.

The last module, named as status update module is implemented to update the status of the tasks in the server data base. The module provides an option to search for the completed and incomplete tasks. A summary of the status is also available for analysis. The server database is dynamically refreshed whenever the tasks are completed.

IV. IMPLEMENTATION

The figure 3.1 displays the User Interface of the application. It displays the various options available in the project to connect to the client, add tasks, add machines, view summary.

Fig-3.1: Home page

The following figure 3.2 adds a new client machine to the network along with the required credentials.

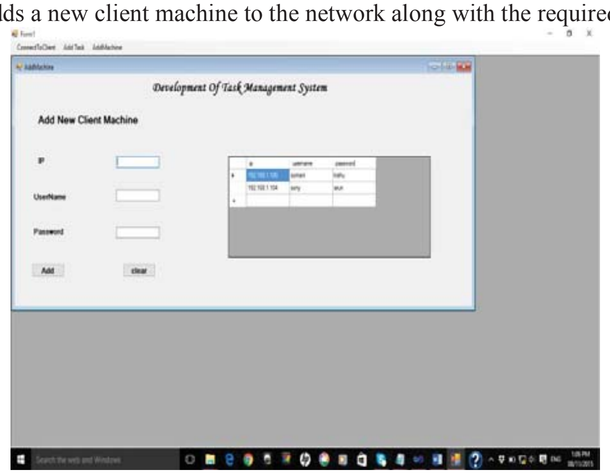

Fig-3.2: Add New Client

The figure 3.3 explains the addition of a new task to the clients at a specific time. Based on the IP address of the requested client, the tasks will be added and stored in the server data base.

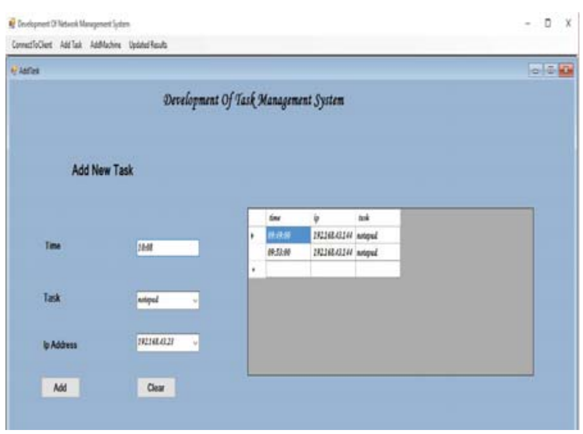

Fig-3.3: Add New Task

The figure 3.4 displays the dynamic connection establishment between the client and the server. The server provides two options: Once the start server option is enabled, a connection is established and the task is performed. If there is a failure, the ping option can be used to manually connect the client.

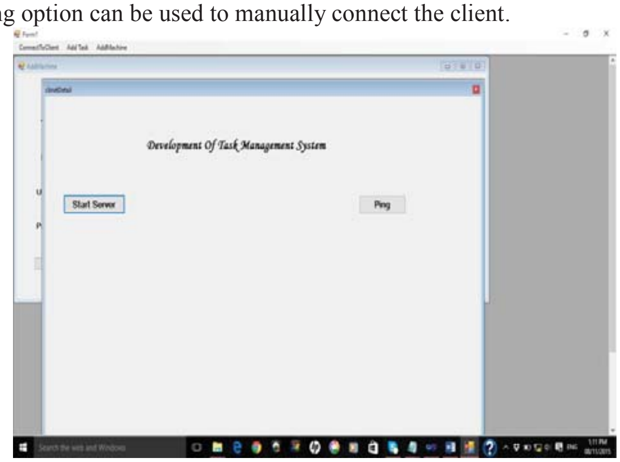

Fig-3.4: Start Server Page

Successful execution is explained in figure 3.5 where a Client request is being executed with a command prompt as a task to be performed at Client machine at a specified time that they had requested.

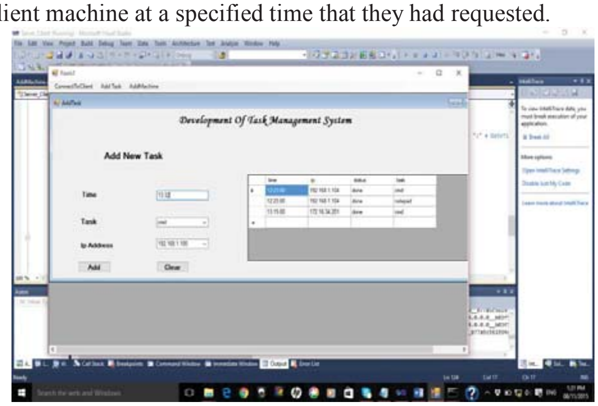

Fig-3.5: Executing Client Task

The figure 3.6 shows the execution of Client request to open a notepad at the specific time.

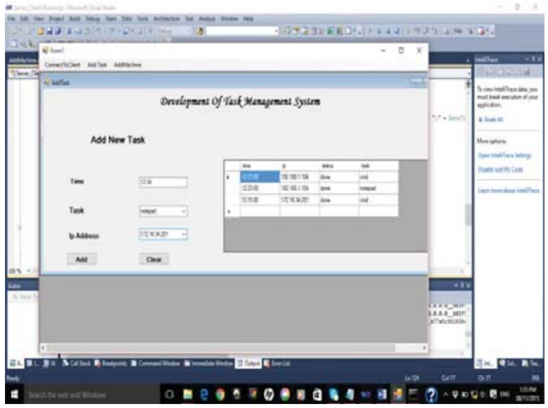

Fig-3.6: Executing Different Task.

In the below figure 3.7 shows the successfully executed output at a Client machine where a notepad task has been performed at a requested time.

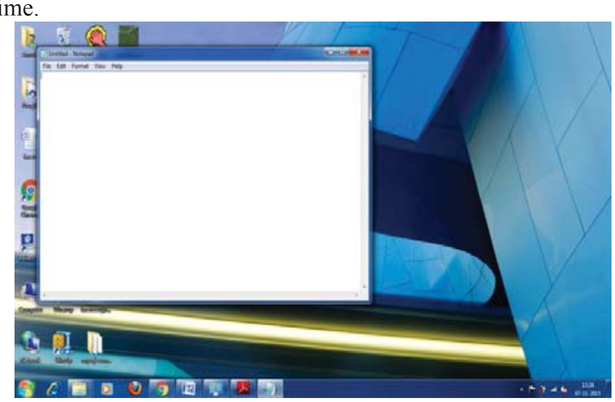

Fig-3.7: Successfully Executed Task.

V. CONCLUSION

The project Development of a network management system has been implemented successfully using c# with .Net platform. A client server system has been established in the first phase and the communication is authenticated with the help of the IP address. The data is also stored in the data base with MySQL as a backend. The various system tasks to be performed by various clients requested by the clients are stored in the server data base. The server verifies the date and the time of the performance of the task and forwards to the required client by establishing the connection dynamically. If the connection could not be established, the server tries it for a maximum of five times and reports non connectivity to the data base. The clients acknowledge the completion of the task to the database once it is completed. The data base is automatically updated and refreshed after the completion of the tasks so that the server overhead is reduced. Reports were generated for the success and unsuccessful completion of tasks. In future, the project can be extended to connect devices for Internet of Things

REFERENCES

[1] Maus, Heiko, M.P. van der Aalst, Wil, Rickayzen, Alan, Riss, Uwe. V. “Challenges for Business Processes and Task Management,” Journal of Universal Knowledge Management. Volume 0, Issue 2, 2005.

[2] D. Njoku, K.I. Nkuma-Udah and C.G. Onwugbufo “Client – Server Architecture: An Overview ”Department of Computer Science, Imo State Polytechnic, Umuagwo, Imo State, Nigeria. IEEE paper on Decenber 10, 2013.

[3] J.M. Suri,DDG(I),TEC, and B.K.Nath,Dir(I)TEC "Time Synchronization In IP Address " by TEC Khurshid Lal Bhawan Janpath, New Delhi-1.
[4] https://technet.microsoft.com/en us/sysinternals/psexec.aspx May 2, 2014 – PsExec.
[5] NET 4.5 Programming 6-in-1, Black Book. By Kogent Learning Solutions Inc ISBN-9789350045107.
[6] "Supported Platforms: MySQL Database". Oracle. Retrieved 24 March 2014. "Downloads". *MySQL*. Retrieved 2014-08-03.
[7] "Waterfall Software Development Model". 5 February 2014. Retrieved 11 August 2014.
[8] Harper, Douglas. "server". Online Etymology Dictionary. Retrieved 30 November 2013.
[9] Manufacturing & Production - s4gv*s4gv.com/s4gv-solution-1/manufacturing-production.html* by S4GV_Executive Team on February 07, 2014

Acknowlegement : We would like to acknowledge our advisor Mr. Elvis Antony, Architect, Siemens Technology and Services Private Limited" for his constant guidance for the project.