

Remote Monitoring System

Pranjal Mittal¹, Kavya Goel²

¹B.Tech (Final Year), Information Technology, ABES Engineering College, Uttar Pradesh, India

²B.Tech (Final Year), Information Technology, ABES Engineering College, Uttar Pradesh, India

Abstract - The Remote Monitoring System is used to monitor system resources that are connected in local network in real time. The Remote Monitoring System organizes and presents various performance factors like CPU Usage, Disk Usage, Memory Usage, Process list based on Memory Usage and Process list based on Disk Usage. These resources are represented in the map form, tabular form and graphical form. Here we are using flask as a platform on server side script, ReactJS frontend tool on client side, firebase for authentication and mongo dB as our database. Therefore we develop python script file that needs to be installed on every machine that is connected on the network be it LAN, Wi-Fi or any other that fetch resource information on real time basis. Data gets refreshed with the help of APIs on continuous basis. This helps to organize and maintain all the resources as well as user system information. The objective of the invention is to provide a means of selecting and grouping both historical and real time information into task packages to present it on a computer display of the host system.

Key Words: CPU Monitoring, Disk Monitoring, Memory Monitoring, User Interaction, Process Monitoring, Graphical user interface (GUI), Linux System.

1. INTRODUCTION

The variety of ways we use computers in every industry is hidden from none. Be it healthcare, media, government, researches, entertainment or space, every field is associated with heavy usages of computer systems. We use computers non-stop without even giving them any attention. We keep on using our computers without even thinking of their performance issues like CPU usage, memory usage, disk usage, per application memory consumption, per application space consumption, speed of the system etc. In this fast evolving world we cannot even imagine of slowing our system and neither do we want to give it a regular check for every system individually, so that is why we have designed this system to check all these measures of all the subsequent systems connected on the network at real time. One of the main advantages is that there is no specific limit set for the number of devices whose performance measures are to be fetched, but the basic requirement is a robust host machine which can withstand the bulk of data coming from various machines at regular intervals.

We have conducted a study of the window management practices for designing our system and design in such a way so that it can be used to monitor multiple devices at a time. After discussing related work and some definitions, we forth present the tool we developed for the performance measurement. We will then present the statistics we will derive from the data generated by the system. Finally we will discuss how these findings will be beneficial for the end users and conclude that there are much more enhancements and up-gradations that are required for remote monitoring of performances.

2. LITERATURE SURVEY

Under literature survey we studied the various research paper related to Remote Monitoring System.

In [1] The task manager window gives us the information related to the operations happening in the telecommunication network. Moreover the task operator has the privilege of accessing all the data associated with a particular task in a single display. The tasks comprises of single events which are completed in an iterative step by step method. In order to manipulate these events or data simultaneously instead of sequential pattern we take help of Task Support Window. In their work they represent following terms:

- 1) Switching Windows and Taskbar Usage.
- 2) Amount of Time that Windows Are Active.
- 3) Number of visible windows
- 4) Empty Space
- 5) Monitor Usage

In [2] Earlier the network operator had to make extra efforts for locating the tasks corresponding t current activities and find out bits of information to solve the problem. But now the scenario has changed and the operator can access all the information related in a task. In our research we are using VIBELOG as a windows operating logging tool. This tool is used to maintain logs of events where each log comprises of window for every activity like opening, closing, activating, sizing, moving, minimizing and restoring a window. It is important to distinguish between the two terms i.e. window visibility and active window state. Window visibility refers to all the windows

visible at a particular instance of time while only one window can be active at a time which is called the active window.

3. DEFINITION

The purpose of Remote Monitoring System is to monitor the performance of Agent Machines. It will make it easy to access the performance data of various machines on a single machine. Takes the complexity out of monitoring systems and simplify the process that allows CPU, memory and I/O Processes helps us to visualize core physical and metrics.

Monitoring activities should be performed to overcome the hardware and software issues. So, Performance Monitoring System used to provide a continuously updating display of the system's resources. These real-time monitoring utilities are good for multiple machines to see consumption of System resources. As there is no tool for Linux to monitor all system that are connected on network to monitor resources we develop a tool to monitor CPU Usage, Memory Usage, Disk Usage of each system real-time. Hence we decide to study the concept to develop tool for this requirement.

4. BACKGROUND STUDY

We take 4 weeks of study of their activity. Our first focus is how to get desired data from the Linux system and transform in our way after our detailed study decided to develop a bash operating file and using PSUtil library with OS library to take required parameter and send data to server so that it display on main dashboard.

For database we decide to use non-relational database Mongo DB to store system resources activity at regular interval of time and use this data to take record of system past activities as well. With the help of data and Ajax we are able to display and refresh at regular interval of time. Using same data we are able to represent in different form like in Map view, Tabular view, Gauge view.

At Back-end, the system has a Flask server which encapsulates the APIs and communicates with the Agent machines to gather the performance data.

After lot of discussion choose React JS as our front end tool as it allow us to develop our tool attractive as well as efficient. As react develop single page application and render at server it is very fast and reliable. We decide to use lot of inbuilt react functionality to incorporate with our tool and display our systems data at a single window with multiple functionality. At front-end the system user react library to dynamically generate views. Bootstrap is used to make the web interface responsive.

AJAX would be used to send the requests to APIs on the server. At agent machines, the system has a micro-service installed which communicates with the server to send its performance data.

For authentication decided to use Google Authentication as well as though user database first system is authenticate as well as user needs authentication to sign-in to view data of all the system that are authenticated as well as connected on same network.

5. DETAILED DESCRIPTION OF TOOL

We were interested in underlying how people used their system resources. Here we present information on how people use their system CPU Usage, Memory Usage, Disk Usage in different representation form.

1. Linux OS will be supported on the agent machines
 2. Users and admin can view performance details in three views
 - o Tabular Representation
 - o Graphical Node(Spatial) Representation
 - o Map Representation
 3. Admin and User Interface are on the web-based app.
 4. Admin can monitor its current machine's performance if the machine is running a Linux operating system.
 5. Users can only view the performance data from all the agent machines.
 6. Users can be authenticated using Google, Facebook or Custom Credentials.
 7. Users can reposition the node on the canvas.
 8. Admin can only be authenticated using system credentials of the server system.
 9. Admin can add/delete the agent machines.
 10. Node's connection state is indicated on the graph view by green and red indicators (Active/Inactive).
 11. Agent machine's performance details are shown (in Percentage) on hover on the nodes in the canvas view.
 12. Performance data would be stored in the database and historical data would be stored in a spreadsheet (Excel) file.
- a. **Map View:** This Screen will display all the active and inactive nodes which are connected to the central server.
- (i) The active nodes can be identified by the green colour indicator displayed in the upper right corner of every node.
 - (ii) The inactive nodes can be identified by the red colour indicator displayed in the upper right corner of every node.

On hovering the mouse icon on any node in this screen we can see the current CPU, Memory and Disk utilization of that particular node.

The screen will contain icons to go to other screens.

To go to the tabular data display screen click on the table icon

To go to the gauge display screen click on the double circle icon

To go to the node mapping screen click on the arrow icon.

- The screen will contain a button to export Performance data of every node to the database.
- The screen will contain an option to hide the MAC Addresses of all the nodes.

b. Tabular View: This screen will display the performance data coming from various nodes connected to the central server.

- The various performance data displayed for each node will be :

- 1) Mac address of the System.
- 2) Individual Core CPU Utilization Percentage.
- 3) Total Memory Utilization Percentage.
- 4) Total used Disk Percentage.

- The screen will contain a button to export Performance data of every node to an Excel File (Spreadsheet File).
- The screen will contain icons to go to other screens:

To go to the tabular data display screen click on the table icon

To go to the gauge display screen click on the double circle icon

To go to the node mapping screen click on the arrow icon.

c. Gauge View: This screen will display the performance data of every node connected to the central server in graphical form with the help of gauges.

- For each node, the following gauges will be displayed alongside their respective MAC Addresses:
- Gauge for displaying the total CPU percentage of every node.
- Gauge for displaying total RAM utilized.
- Gauge for displaying total disk space utilized.
- The screen will contain a button to export Performance data of every node to an Excel File (Spreadsheet File).

This screen will also contain icons to go to other screens:

- (i). To go to the tabular data display screen click on the table icon
- (ii). To go to the gauge display screen click on the double circle icon
- (iii). To go to the node mapping screen click on the arrow icon.

(iv). The screen will contain a button at the lowermost right corner of the screen to export Performance data of every node to the database.

(v). The screen will contain an option in the upper right corner of the screen to hide the MAC Addresses of all the nodes.

6. PER SYSTEM ANALYSIS

Our tool helps us to inspect disk management, memory management, CPU usage of each individual system. We designed a visualization tool to inspect patterns of visibility for each individual system.

In order to keep your system fit for purpose, monitoring activities should be performed to overcome the hardware and software issues. So, Remote Monitoring System used to provide a continuously updating display of the system's resources. These real-time monitoring utilities are good for multiple machines to see consumption of system resources:

- Monitoring CPU usage.
- Disk Usage.
- Memory Usage.
- Top 5 Processes consuming Highest CPU.
- Top 5 Processes Consuming Highest Memory.

a. CPU Usage: By using bash file using PSUtil and OS library get the data of CPU usage as well as no of core and there usage and send it to server so that it can display on our tool. We can get data after regular interval and we can alter time duration as well. By this we are able to monitor real time CPU Usage and display on administrator screen to easily monitor and take action accordingly.

b. Disk Usage: We can get data of disk usage of the system available space and total space in system in real time. With the help of this we are able to know if system need extra memory or is there any problem in the system. We can also get process name that consume high disk area so that able to take required action on that process if required.

c. Memory Usage: We can get data of volatile memory (RAM) consumption of the each system in real time basis and send it to the server so that it can display on our tool. We can get data after regular interval and we can set refresh timer according to our need. By this we are able to monitor Memory Usage and display it to easily monitor and take action accordingly.

d. Top 5 Processes consuming Highest CPU: In our tool we display top 5 Processes name with their consumption units, its make easy for administrator to monitor such processes and take required action when needed. We can get updated at

regular interval so that if new process consume more than previous one update in the list we can set refresh timer according to our need by default it get refresh in every 5 millisecond.

e. Top 5 Processes consuming Highest Memory: In our tool we display top 5 Processes name with their consumption units, its make easy for administrator to monitor such processes and take required action when needed. We can get updated at regular interval so that if new process consume more than previous one update in the list we can set refresh timer according to our need by default it get refresh in every 5 millisecond.

7. TOOL IMAGES

| Sr. No: | MAC Address | CPU Usage (%) | | | | Memory Usage (%) | Disk Usage (%) |
|---------|-------------------|---------------|-------|-------|-------|------------------|----------------|
| | | Core0 | Core1 | Core2 | Core3 | | |
| 1 | E4:02:9B:1A:A0:C7 | 34 | 23 | 54 | 12 | 56 | 72 |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |

Fig. Tabular View



Fig. Map View

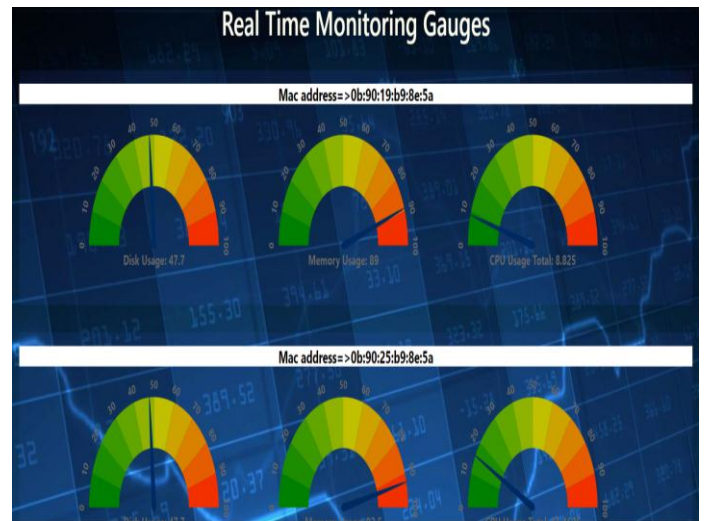


Fig. Gauge Screen

8. CONCLUSION

The usage of system display is likely to increase in the coming scenario. This is mainly because of two reasons. The first reason being the reduction in the prices of display and the second reason is the increase in power of the processors and video cards. In order to understand how to evolve new design techniques with the changing requirements of people, we firstly need to have an intense knowledge of how windows are managed, how the interaction between the users and windows takes place and how the present systems are different from the previously evolved systems. So the overall conclusion we have investigated and presented in this research paper we have some starting points, similarities and dissimilarities between present and past models, their merits and demerits. We also present in our research paper several techniques on how a user can deal with different systems.

9. FUTURE SCOPE

Our work demonstrates a robust tool to measure the importance of a window. This robust tool is computer usage analytics and visualizations. These visualizations can be fruitful in many other ways as well, such as which activity uses how much system resources, which activity the user was working on lately, we can compare various activities based on their performance, utility and resource usage. Also we create different classes of users based on their window switching mechanisms.

Where work on systems usage tends to focus on the resource usage by different systems. Our work mainly focuses on representing the consumption of system resources of all the systems that are connected on network in a single window data which is refreshed or updated at regular intervals. If

users are provided with more display space then it would require less window switching to access all the connected systems.

Windows are generally active for short periods of time, but looking at visualization of active window visibility. Future modification in the tool are as follow.

* Compatible with mac OS.

* Build Desktop Application

* Integrate with Android/IOS.

* Notify Users consuming resources more that 75% though mail.

REFERENCES

- [1] M. Novel interaction techniques for overlapping.
- [2] Feiner, and Bell Dynamic space management.
- [3] Rosenberg, J. K, Bly and S. A. A comparison of tiled and overlapping windows.
- [4] J. G, Funke, D. J., Neal and Paul, R. D. An approach to automated Window management.
- [5] Gaylin notes on creating empirically-based windowing benchmark task.
- [6] S. K. Rooms, Henderson, D. A. Jr. and Card: The use of virtual Workspaces to reduce space contention in a window-based graphical user Interface.
- [7] Hutchings, D. R. and Stasko, J. Revisiting display space management.