Performance Monitoring System for Virtual Machines

Sri Upanya B[1], Uthra.V[2], Dr.B.Monica Jenefer[3]
[1&2] Final Year Students,[3] Head of the Department,
Department of Computer Science and Engineering,
Meenakshi Sundararajan Engineering College,
Chennai,India

Abstract

Various virtualization technologies are adopted to reduce the cost while maximizing the productivity, flexibility, responsiveness, and efficiency. In this paper a virtual environment is proposed where there are multiple virtual machines including several virtual clients and a single virtual server. The virtual clients are monitored by a centralized server. A machine's resource utilization depends on the applications that are running on it. Thus the system metrics indicate the excessive use of a machine. Processor performance, memory utilization and network properties are fetched from the client and communicated to the server machine using ZeroMQ, a message queuing technique. ZeroMQ is an asynchronous messaging library where the API provides sockets for communication. This data that is received by the server is logged and evaluated. If certain resource utilization exceeds a threshold, an email notification is triggered. This system provides a highly secure and efficient environment with limited memory overhead. It is especially useful for applications such as military and medicine where confidentiality is a prime concern.

Keywords - *virtualization, virtual machine, message queues, performance monitoring, ZeroMQ*

I. INTRODUCTION

Virtualization is the foundation of cloud computing. The hypervisor is a program that enables hosting of several virtual machine on a single hardware. It divides the hardware resources across the various virtual machines. By separating the operating system and the applications from the underlying physical hardware, it is able to provide advantages like ease of deployment, ease of management, reduced cost, portability and optimal utilization of resources. However, it is prone to risks such as virtual machine failure, virtual machine separation and other kinds of issues. To protect the VMs in an efficient way, a monitoring system was first proposed by Jiangyong Shi , where the virtual machines memory, network and file systems were monitored. By assigning higher privileges to a single centralized virtual machine, the various performance metrics of the virtual machines may be gathered and monitored. The performance of virtual machines memory, processor and network are monitored. In the first part of the paper the methodology for gathering this data is discussed. Message queues are used for communicating performance data. The details are discussed in the second part of the paper. This communicated data are stored in the central server for analysis. The monitored results are shown in the third part of the paper. Additionally, an environment which makes use of this monitored data is proposed.

II. RELATED WORK

Jiangyong Shi provided an introspection based virtual machine monitoring system that used the VMI technology to monitor the virtual machine one the hypervisor layer. The semantic information was analysed for intrusion using software tools like Snort, OSSEC and Volatility. The VM's memory, file system and network semantics were monitored. However, the passive technique is vulnerable to transient attacks.

Kenichi Kourai compared the performance of the virtual machine with that of the physical machine and provided methods by which the virtual machine could perform better than the physical machine. Though performance enhancement methods were provided, they were only applicable for certain configurations of the virtual machines and worsened the performance for other setups.

Hafiz ur Rahman monitored the virtual machine's performance by applying complex and heavy workload. But, the results were inconclusive because of the limited number of virtual desktops that were implemented.

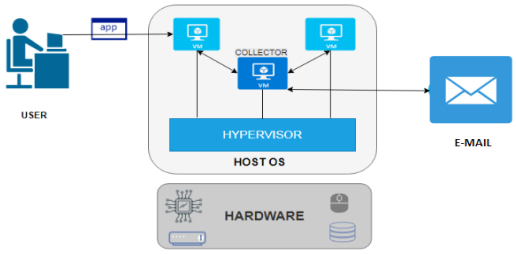
Swati Pawar compared the performances of virtual machines that were full virtualized and para virtualized. A variety of semantics were monitored but no analytics on the data was provided.

Aravind Menon used Xenoprof tool to analyse network performance in Xen hypervisor.

Our monitoring system evaluates multiple semantics such as CPU performance, memory utilization and network performance to provide a comprehensive approach to virtual machine monitoring. Further a method to utilize this study is provided.

III. METHODOLOGY

The framework for this monitoring system is provided in figure 1. It mainly consists of two types of system – central server and client VMs. The central server is a virtual machine that has been assigned higher privileges. This server monitors the performance of the client VMs. The performance data is collected from the client and communicated to the central server. The data is collected and output such as a response or an alert is generated by the central server.



Here two Type-2 hypervisors are used- VirtualBox and VMware Workstation. Type-2 hypervisors run over the existing host's operating system. The operating system used is the Ubuntu LTS 18.04.

A. *Processor Monitoring*

The processor can be monitored by using the information stored by the operating system. This information consist of CPU statistics such as number of processors, processor speed, load, idle time and active time. The data is stored in a structure and communicated to the central server. This data received by the server is analysed. A load greater than a threshold will trigger an output such as a response or an alert. This response may be a system log or an e-mail trigger.

Here the CPU load average is analysed. A load of 1.0 denotes 100% utilization of a single core. If the load is greater than 1.0 it indicates that processer resource is insufficient. This results in contention among the various processes for the core. Example, if the load is 1.75, then it means that on an average, 1.75 processes are trying to execute parallelly.

Similarly for a 2-cored system, the load average of 2.0 denotes 100% utilization and so on.

B. *Memory Monitoring*

Similar to processor monitoring, the memory is monitored from the RAM information provided by the operating system. By accessing this data, we can compute the amount of memory that is utilized by the various processes that are running on the system. This data is acquired in a structure and communicated to the server continuously and it is analysed. When the memory utilized crosses the threshold value a will trigger an output such as a response or an alert. This response may be a system log or an e-mail trigger based on the application of the monitoring system.

C. *Network Monitoring*

The network is monitored using network monitoring tools such as Monitorix, iftop or NetHogs. These tools help in obtaining the information about the number of packets sent and number of packets received. The data is communicated to the server . This data received by the server is analysed. Any spike in the network performance will trigger an output such as a response or an alert which can be a system log or an e-mail trigger based on the application of the monitoring system.

D. *Communication*

The communication between virtual client and the server works under the following conditions: Firstly, the entire information about a particular metric is sent as a single entity. Secondly, multiple virtual client should be able to communicate with the server at a given instant. Communication mechanisms are built using message queues. ZeroMQ [6] is an asynchronous messaging library where the API provides sockets for communication. It provides a method to send atomic messages over various transports like in-process, inter-process, TCP and multi-cast with very low latency.

The server is able to connect to multiple clients through multithreading. Multithreading is the ability of a processor to execute multiple processes or threads concurrently. This allows several VMs to be monitored at the same time by a single server. Since a two way communication is established between the server and the virtual clients, the server is able to issue commands and alerts to the clients.

IV. RESULT

By analysing several processes, the average load for a typical single cored system is obtained. These values are plotted to form a graph.

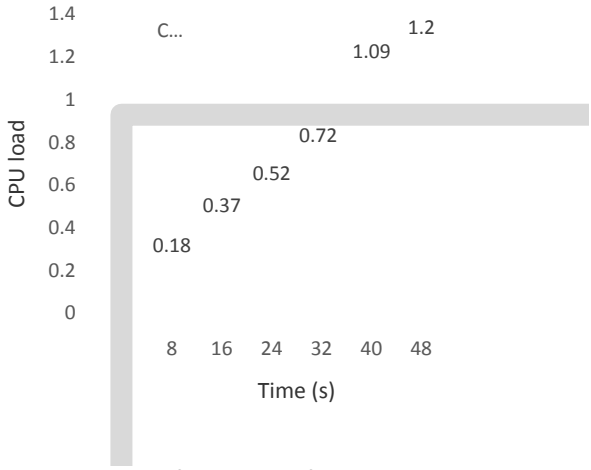


Fig. 2 A graph of processor load for a single cored system.

Here, 1.0 is chosen as the threshold for load since it represents 100% utilization of a single cored system. Any value greater than 1.0 denotes that the processes require more than the available CPU resource due to which a contention takes place.

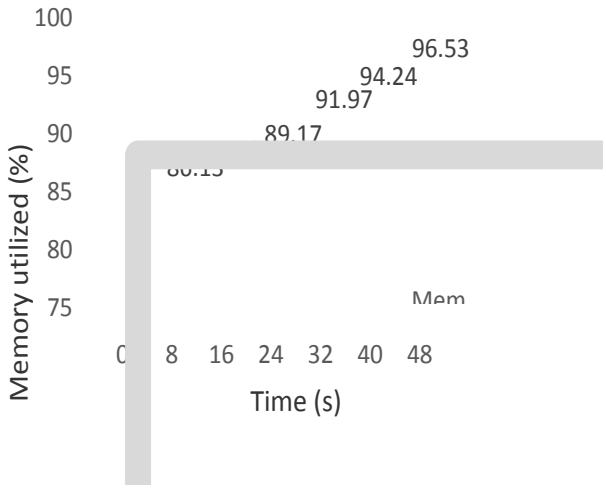
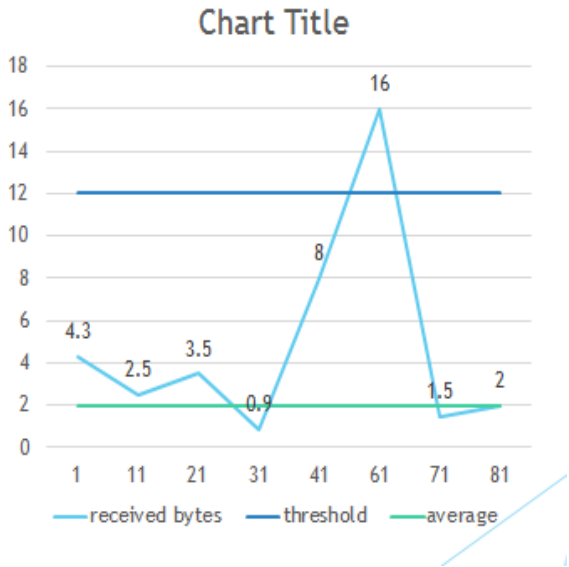


Fig. 3 A graph of memory utilization for a single cored system.

The memory utilization is calculated using the total memory and the memory utilized. Here, the threshold is considered to be 90%, which is the optimal usage of system RAM. When we run more processes at a time, there will be more load on the memory which in turn will increase the memory utilized. When there is utilization of RAM more than 90%, the threshold is surpassed and an alert is triggered.



Similarly, when a system uses extra network bandwidth by sending or receiving more packets of data than that of the optimal or threshold considered, an alert is triggered by the server.

V. CONCLUSION

In this paper, we have discussed the benefits of virtual machines and provided a monitoring system for the virtual machines. The monitoring system protects the VMs in an efficient manner by monitoring semantics such as processor performance and CPU work load, RAM utilizated by the processes running on the system and network bandwidth utilized by the client for sending and receiving packets. This monitoring is done by the single centralized virtual server. This server gathers the various performance metrics from the clients with the help of ZeroMQ message queuing technique which provides efficient communication between the client and the server using sockets. These data are logged, analysed and the results are provided in the form of a response or alert. This response is generated when the value of the data analysed crosses the threshold. Thus a comprehensive approach for performance monitoring the virtual machines is generated with minimal overhead.

REFERENCES

[1] Jiangyong Shi,Yuexiang Yang "Design Of A Comprehensive Virtual Machine Monitoring System" 2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems

[2] Xen,Kenichi Kourai, Riku Nakata "Analysis of the Impact of CPU Virtualization on Parallel Applications in Xen"

[3] Hafiz ur Rahman, Farag Azzedin, Ahmad Shawahna, Faisal Sajjad, Alyahya Saleh Abdulrahman, "Performance Evaluation of VDI Environment" Information and Computer Science (ICS), KFUPM,Dhahran, KSA

[4] Swati Pawar, Sarvesh Singh, "Performance Comparison of VMware and Xen Hypervisor on Guest OS" International Journal of Innovative Computer Science & Engineering ;Volume 2 Issue 3; July-August-2015; Page No. 56-60

[5] Aravind Menon, Jose Renato Santos, Yoshio Turner, G. (John) Janakiraman, Willy Zwaenepoel, "Diagnosing Performance Overheads in the Xen Virtual Machine Environment"

[6] http://zguide.zeromq.org/page:all

[7] https://www.vmware.com/solutions/virtualization.html

[8] Sonam Srivastava, S.P Singh "A Survey on Virtualization and Hypervisor-based Technology in Cloud Computing Environment" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5 Issue 2, February 2016