International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Analytics of Application Resource Utilization within the Virtual Machine

Priyanka H

Department of Computer Science and Engineering, Nandi Institute of Technology and Management sciences

Abstract: ***Cloud environments are becoming increasingly prominent and are hosted in large data enters. These large data centres have large number of physical servers which in turn are virtualized into large number of Virtual Machines (VMs) which are orchestrated by cloud infrastructure like openstack, cloud stack etc. These cloud environments support large number of applications which consume resources in a varied sizes and are supported on these VM environment. These cloud infrastructures will need to monitor and manage these applications which are consuming resources in a wide variance for efficiency and effectiveness. This necessitates the cloud infrastructure to balance the utilization of the resources by applications in different VMs and attach new resources as needed or redistribute application to the VMs as relevant. To effectively manage these virtual systems and utility clouds, operators must understand current system and application behaviours. This requires continuous monitoring of resources consumed by the application along with analysis of the data captured by the monitoring system. This aims to monitor the resources utilized by application in the respective VMs and determine decisions for either migration of application or addition/reduction of resources based on configurable thresholds. This approach is leading from infrastructure based resource monitoring and management to application centric monitoring and management in the VMs.***

Keywords: Analytics, VMS , Applications monitoring, Cloud

1. Introduction

Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services. Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users.

Cloud computing is a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications.

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.
Cloud Environment consists of number of servers, storage and network component interconnects them. In cloud infrastructure there is a software layer running on the servers which will utilize these storage and network components. It has hypervisor layer like kvm which will help to create virtual machines in a cloud environment.

The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing have led to a growth in cloud computing.

In a cloud data centre, there are physical servers with a large number of virtual machines. These virtual machines are hosted with many applications. In order to optimize the utilization of computing resources, the applications running on the virtual machines need to be monitored. Identifying when it is best to migrate an application in a virtual machine has a direct impact on resource optimization. Performance optimization can be best achieved by an efficiently monitoring the utilization of computing resources. So, we need intelligent monitoring agent to analyse the performances of virtual machines.

This paper proposes an agent based resource monitoring system that depicts the CPU and memory utilization. The monitoring agent collects the virtual machine resource utilization and displays in a dashboard. Dashboard displays the key performance metrics such as CPU and memory utilization. The analysis report of dashboard provides information to administrator for resource optimization

Cloud infrastructures are hosted as large number of physical servers. Hypervisor layer of these servers creates n number of virtual machine. To effectively manage these virtual systems and utility clouds, operators must understand current system and application behaviours. This requires continuous monitoring of resources consumed by the application along with analysis of the data captured by the monitoring system. This necessitates the cloud infrastructure to balance the utilization of the resources by applications in different VMs and attach new resources as needed or redistribute application to the VMs as relevant .A good understanding of the resource usage of applications in the system is necessary to drive linear performance characteristics for these applications. This also facilitates on whether and when new applications will need to be scheduled to run.

2. Related Work

Monalytics: Online Monitoring and Analytics for Managing Large Scale Data Centers[1]

In this, studies have been performed in terms of monitoring, it has created scalable methods for real-time data collection and aggregation to support efficient online queries that answer questions like "which machines have CPU utilization above 90%?"
Analysis-focused research has drawn from areas like data mining, machine learning, and statistics to create techniques that assist in or automate problem diagnosis with high accuracy and significantly reduced human intervention.

Monitoring has been shown feasible at scale and in real-time, analysis is typically performed after a volume of monitoring data has been written to disk-resident logs, or in a central location, which impedes the scalability of on-line monitoring and analysis tasks. Further, due to lack of underlying infrastructure support, analytics often require global data – over time and space – making it difficult to use them on-demand and in real-time. Finally, in modern virtualized utility or cloud computing systems, operators or administrators have limited visibility into the virtual machines running on data center machines. This prevents them from using problem diagnosis methods that require such insight.

To address these challenges, they have come up with the system integrating monitoring with analytics, which can capture, aggregate, and incrementally analyze data on demand and in real-time and to the extents needed by intended management actions. They use „analysis" and „analytics" interchangeably accommodate the changing and diverse characteristics of analytics, with cost-effectiveness in large-scale data centers. This presents the design and evaluations of flexible architecture built upon dynamic distributed computation graphs (DCGs), providing the following technical contributions:

- Pro re nata (PRN) deployment: an important property of our system is its instantiations of analytic functions only where and when they are needed. In other words, it must have capabilities for dynamically zooming into „interesting" locations and periods of time. Such capabilities can also benefit scalability by substantially reduced costs compared with systems forced to „watch everything all the time". We validate the PRN deployment in two realistic use cases and compare them with traditional brute-force approaches.
- Reducing „Time to Insight" (TTI) and cost: a vital metric for assessing the performance of monitoring/analysis actions is Time to Insight(TTI) capturing the total delay between when „interesting" events occur until they are recognized (i.e., after analysis is complete). Using this metric and also assessing the costs incurred along with different values of TTI, we evaluate the cost-effectiveness of alternative topologies used to construct DCGs, and validate our novel flexible hybrid DCG design [1].

Fast and Scalable Real-time Monitoring System [5]
In this work they have discussed about monitoring and analyzing the applications using a design called SCMS. Fast real-time monitoring of system information is important to the understanding of parallel system especially for a large cluster system that appeared recently. Making the system fast and scalable at the same time is still a challenging task. This presents the design and implementation of a fast and real time monitoring system called SCMS/RMS. This system is a part of more comprehensive cluster management tool called SCMS [6]*. SCMS/RMS is designed to be flexible, highly scalable, and efficient. Many techniques that are used to increase the monitoring speed and to achieve high scalability have been described in this paper. The experiment has been conducted on a 72 nodes Beowulf Cluster and the results show that SCMS/RMS is very fast and highly scalable.

System performance monitoring is important since it allows system administrator to understand system behavior or, in many cases, predict the malfunction earlier. Moreover, performance information can help many important subsystems such as batch scheduler to make a better decision about system resource allocation. The implementation of fast and efficient real time monitoring is a challenging task especially for the recent large-scale clusters with thousands of node. Many works related to system monitoring appeared in the literatures. Many tools such as CIS[8], ClusterProbe[9], GARDMON[10] , PARMON [11], Co-Pilot [12], are built specifically for system monitoring. Moreover, many monitoring systems appear as a part of system management tools such as SCMS and VACM . The design of each tool is rather different due to the goal and complexity of the monitoring subsystem itself. For example, many tools rely on usual system interface such as /proc in Linux to access operating system performance data. But some tools such as CIS develop their own kernel probe to increase the speed of access and to reduce the level of intrusiveness. Scalability is also a major issue being addressed by many works. Many monitoring subsystems [1][3][4][6][5]* are still based on centralized daemon or applications that collect the information from the distributed agents running on every node in the system. This obviously limits the scalability of the system.

Hierarchical monitoring is employed to enhance the scalability in Cluster Probe. A performance study of this is presented in [6]*. Cluster Probe uses advanced techniques such as data filtering and merging to further reduce the monitoring traffic. The tool called CIS introduces the technique of adjusting the monitoring frequency to reduce the impact. This technique and more are also supported by our implementation. Most monitoring is based on their own protocol over TCP/UDP link except in [7]* which builds a large scale monitoring based on SNMP protocol

A Scalable, Commodity Data Center Network Architecture [2]

In this work they discussed about migration of application in the virtual machines and behavior of applications within VMs,

Virtual machine (VM) technology has recently emerged as an essential building block for data centers and cluster systems, mainly due to its capabilities of isolating, consolidating and migrating workload [2]. Altogether, these features allow a data center to serve multiple users in a secure, flexible and efficient way. Consequently, these virtualized infrastructures are considering a key component to drive the emerging Cloud Computing paradigm [2]. Migration of virtual machines seeks to improve manageability, performance and fault tolerance of systems. More specially, the reasons that justify VM migration in a production system include: the need to balance system load, which can be accomplished by migrating VMs out of overloaded/overheated servers; and the need of selectively bringing servers down for maintenance after migrating their workload to other servers. The ability to migrate an entire operating system overcomes most difficulties that traditionally have made process level migration a complex operation [3]*. The applications themselves and their corresponding processes do not need to be aware that a migration is occurring. Hypervisors, such as Xen [1] and VMware, allow migrating an OS as it continues to run. Such procedure is termed as \live" or \hot" migration, as opposed to \pure stop-and-copy" or \cold" migration, which involves halting the VM, copying all its memory pages to the destination host and then restarting the new VM. The main advantage of live migration is the possibility to migrate an OS with near-zero downtime, an important feature when live services are being served [3].

B. Proposed System

Monitoring: System performance monitoring is the act of collecting system performance parameters such as node‟s CPU utilization, memory usage, I/O and network rate, and present them in a form that can be easily understood by the system administrator. This service is important for the stable operation of large clusters because it allows the system administrator to spot performance of all the VMs. Moreover, other parts of the systems software can also benefit from the information provided.

Analytics: System Performance is analysed by monitoring CPU and memory usage continuously for the applications running within the VM.

Analytics used in developing project is as follows.An agent which continuously runs for specified time period for several iteration, collects the application resource utilization such as CPU , Memory etc This Resource utilization is saved, which when requested by the Central Management System, it is transferred. The Analytics is carried out in Central Management System, where we developed the code which checks for the application which is continuously utilizing Memory and CPU over the specified threshold for all iteration sent by an agent. The resource utilization detail which is fetched from agent is analysed. Memory and CPU utilization of each application is analysed and looked for those applications which are continuously utilizing resources over the specified threshold. If the average of application resource consumption is crossing the upper limit of the threshold then this application is candidate for either to be moved to another VM or to be monitored and acted upon.

3. System Architecture

Architecture of the system is shown below in the figure. 1. An agent which collects application‟s resource utilization and saves it. These data is transferred to central management system on request. Here the application‟s resource utilization are analysed and then appropriate graphs are drawn and it‟ll be checked which application continuously utilizing more resource like memory and CPU and whole system information is checked and results are displayed. This information is to make decisions for either migration of resources or addition/reduction of resources within the VM.

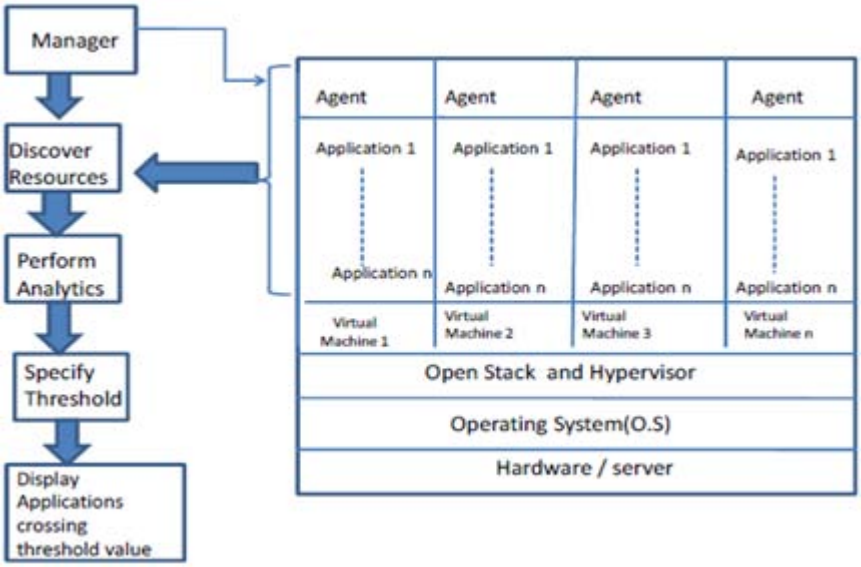

Figure 1: Architecture of Virtual machine application

4. Implementation

Agent

/proc/mem info library system used for reading memory usage of Applications ./proc/stat library system used for reading CPU usage of Applications. Ps aux is Agent which pulls the information from above libraries Agent initiates and collects resource utilization of the VM, where it is installed Once resource monitoring is started by Manager, Agent collects Memory and CPU utilization of each Application running in the VM for 10 iterations when requested by the Manager, the agent transfers resource utilization details to Manager. The Agent is made to run continuously and it collects resource utilization and stores the details. CPU and Memory information is obtained.

Algorithm 1: Agent

```
Timer time = new Timer();
ScheduledTask st = new ScheduledTask();
SheduledTask class
time.schedule(st, 0, 10000);
for (int i = 0; i <= 1; i++) {
Thread.sleep(10000);
p = runTime.exec("ps aux");
while (line != null)
{l
ine = bufferedReader.readLine();
process += line + "|" ;
}
```

Manager

Manager Component runs on the management station. When initiated by the user, Manager component transfers and installs an agent to a machine specified by the IP address When resource monitoring is requested, it initiates the collection of resource utilization details by an agent Manager collects resource usage in different virtual Machines which are running for different agents Manager when requested for resource utilization details, it collects the data from the VM where agent is installed, analyses the same and provides analytics results.

Algorithm 2 Manager

```
JSch jsch = new JSch();
session =
jsch.getSession(SFTPUSER,SFTPHOST,SFTPPORT);
session.setPassword(SFTPPASS);
session.connect();
ChannelSftp sftpChannel = (ChannelSftp)
session.openChannel("sftp");
sftpChannel.connect();
sftpChannel.cd(SFTPWORKINGDIR);
```

Analytics

The resource utilization detail which is fetched from agent is analyzed. Memory and CPU utilization of each application is analyzed and looked for those applications which are continuously utilizing resources over the specified threshold. Analysis is done as follows: Accept the configurable threshold value for memory and CPU for the applications running on the VM. Compare the application resource utilization with the threshold. As part of comparison, calculate the average utilization of 10 iterations to determine which application is crossing threshold i.e. Average of app(Memory usage > Threshold)= (app-itr1+app-itr2+....+app-itr10)/10 iterations

Average of app(CPU usage > Threshold) = (app-itr1+app-itr2+....app-itr10) / 10 iterations .If the average of application resource consumption is crossing the upper limit of the threshold then this application is candidate for either to be moved to another VM or to be monitored and acted upon.

Algorithm 3 Analytics

```
appStr = {};
strVal = [];
a = 1;
for i = 1:numel(App)
if App(i) == ' '
a = a + 1;
strVal = [];
else
strVal = [strVal, App(i)];
appStr{a} = strVal;
end
```

5. Results

The figure 2 shows the results of application which are crossing threshold values. These applications can be scheduled to another VM using suitable scheduling algorithm

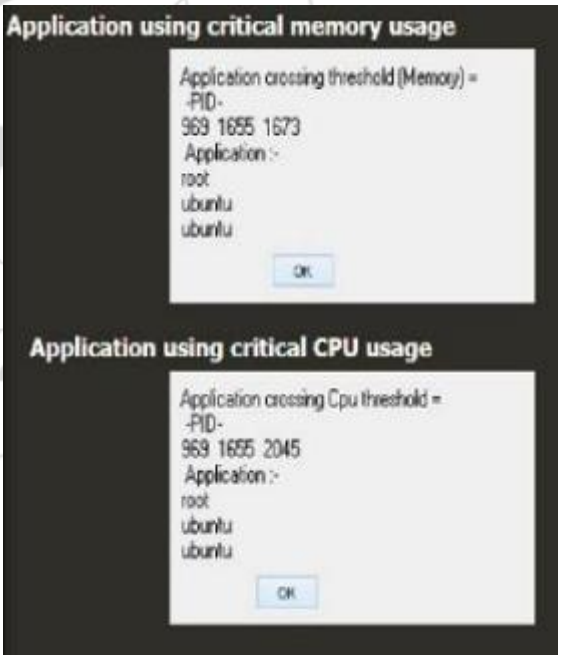

Figure 2: Displaying Applications crossing threshold values by installing Agent to VM

In figure 3 it shows the application consuming more cpu usage, I.e crossing the threshold value. The figure 4 it clearly shows the PID of application consuming more memory usage crossing threshold value. With this results it is more easy to detect application consuming more resource usage and accordingly it can be migrated into another Virtual Machine when needed.

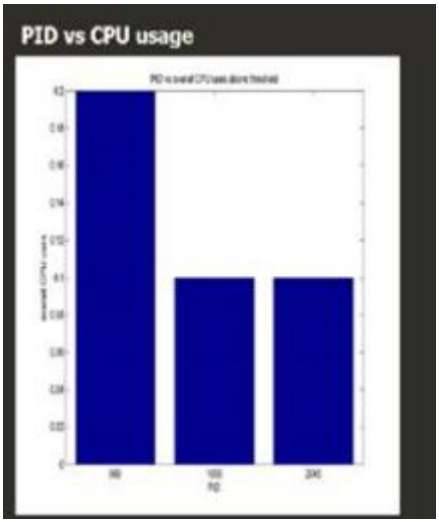

Figure 3: Displaying PID of Applications crossing threshold values of CPU

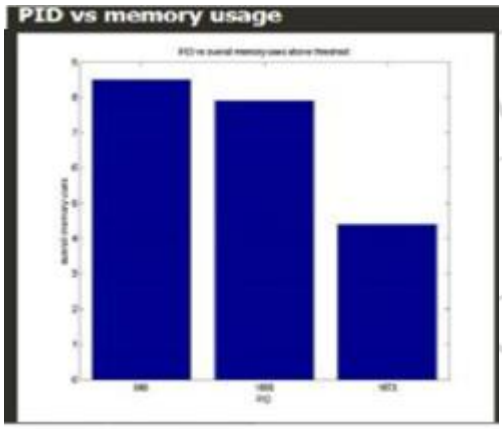

Figure 4: Displaying PID of Applications crossing threshold values of memory

6. Conclusions

In this paper, architecture is presented for resource monitoring using agent based system. Since, agents can be installed anywhere it is easy to use in a cloud environment for monitoring purposes. The proposed does the monitoring of applications within the virtual machine and using analytics checks for resource utilization of each application running within the VMs.

The continuous evaluation helps to check whether the applications running in the virtual machine are performing as expected, for doing this the resource utilization is considered and checked if there is enough resources for new application to run. This paper primarily aims to monitor the resources utilized by application within the VMs and determine decisions for either migration of applications or addition/reduction of resources based on configurable thresholds

In Future this can be extended to monitor and analyze storage and network resource consumption of applications. By Keeping all of the resource consumption profile, scheduling algorithm can be used to schedule new incoming application within virtual machine.

References

[1] Monalytics: Online Monitoring and Analytics for Managing Large Scale Data Centers Mahendra Kutare College of Computing Georgia Institute of Technolog Atlanta, GA 30318, USA imax@cc.gatech.edu
[2] A Scalable, Commodity Data Center Network Architecture, Praveen Yalagandula HP Labs Palo Alto, CA, USA
[3] A Flexible Architecture Integrating Monitoring and Analytics for Managing Large-Scale Data Centers
[4] P.Yalagandula and M. Dahlin. SDIMS: A Scalable Distributed Information Management System. SIGCOMM
[5] Fast and Scalable Real-time Monitoring System for Beowulf Clusters Putchong Uthayopas, Sugree Phatanapherom Parallel Research Group, CONSYL ,Department of Computer Engineering , Faculty of Engineering, Kasetsart University,
[6] Real-time End-to-end Network Monitoring in Large Distributed Systems , Han Hee Song University of Texas at Austin Austin, TX, USA
[7] Gupta G, Younis M. "Performance evaluation of load-balanced clustering of networks,". In: Proc. Of the 10th Int'l Conf. on Telecommunications (ICT). IEEE Press, 2003. 1577−158
[8] J. Astalos, "CIS - Cluster Information Service". http://ups.savba.sk/parcom/cluster/
[9] Z. Liang, Y. Sun, and C. Wang. "ClusterProbe: An Open,Flexible and Scalable Cluster Monitoring Tool", Proceedings of the First International Works hop on Cluster Computing,.
[10] R. Buyya, B. Koshy, and R. Mudlapur, "Gardmon: Ajavabased monitoring tool for gardens non-dedicated cluster computing", In Proceedings of Workshop on Cluster Computing Technologies, Environments, and Applications, Monte Carlo Resort, Las Vegas, Nevada, USA,
[11] R. Buyya, "PARMON: A Portable and Scalable Monitoring System for Clusters", International Journal on Software: Practice & Experience (SPE), John Wiley & Sons,
[12] Silicon Graphics, Inc, "Performance Co-Pilot". http://oss.sgi.com/projects/pcp