Performance Analysis of Downtime in VM using Control Groups for RAM Crash and CPU Overhead

Deepika Shekhawat, Reema Ajmera

Abstract: Cloud computing is one of the offered technology for various services whether it is platform, software, infrastructure and many more but providing services and providing them in efficient way are two different things. Downtime is a frequent problem that occurs due to which users are not able to access the services to overcome this, in this paper the major two problems were analyzed that may be responsible for the downtime and they are RAM Crash and CPU overhead and then problem is overcome by executing Control Groups on Red Hat Enterprise Level(RHEL)

Index Terms: Cloud computing, Virtual machine, Downtime, Control Groups

I. INTRODUCTION

Cloud Computing is a term that has been used for the delivery of variety of hosted services over the internet and its domain. The recent developments in this field have allowed users to access a new paradigm, known as Virtualization. The principle of Virtualization is based on the usage of virtual machines or VMs that have an encapsulated software layer that provides the same behavioral properties like inputs & outputs like an actual physical machine. This encapsulated layer is often surrounded by an operating system and acts as the software equivalent of the host. This concept of virtualization was evolved to minimize the cost of the infrastructure of a cloud-based system. A virtual machine does not depend on the state of the physical hardware or the physical machines. Multiple virtual machines can be installed and used on a set of hardware which makes it advantageous. But with increasing complexity in this infrastructure, further developments are required in order to reduce the factors that give rise to hindrance in high-performance such as downtime. Downtime is the term that refers to the periods when the system remains unavailable or crashes due to high workload. In this duration, the system stops and is unable to perform its primary functions. There can be several reasons or events related to the occurrence of the system's downtime. Some of the events or reasons are planned such as routine maintenance, that requires the system to stop working or unplanned events such as a communication failure or excessive workload that caused it to crash.

Revised Manuscript Received on April 07, 2019.

Deepika Shekhawat, Department, of Computer Science Engineering, JECRC University, Jaipur, India.

Dr. Reema Ajmera, Department, of Computer Application and Information Technology, JECRC University, Jaipur, India.

Downtime in Vms causes a bad user experience along with the risk of loss important data, time & money as the stakes are high. The majority of existing work that builds on live migration of VMs simply assumes some fixed (in many cases arbitrary) duration of the live migration and the downtime involved by it. [1] According to the experiments presented in this paper, such an assumed value has to be chosen very carefully since migration time as well as downtime can vary by an order of magnitude or more, depending on the memory workload. It is the goal of this paper to systematically investigate the factors determining the causes and analysis of causes of downtime in VM**.**

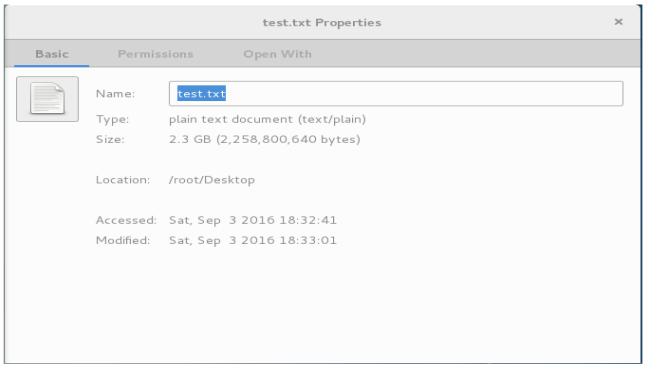
II. ANALYSIS AND METHDOLOGY

Analyzing the various security implication that may be responsible for the downtime includes the load on RAM and CPU due to various factors for example the large number of processes running simultaneously on the machine or on the virtual machine if that happens the machine will go down or its performance definitely get affected due to which the downtime occurs and the processes will unexpectedly be stopped. Hence it is required to limit the resources for the process so that it will use only that allowed and allotted resources.For the above-mentioned objective, I have designed a front end that will reflect the usage of the RAM and the CPU for this I have used various technologies to design the same for example AJAX, JScript, the back end is done by using the programming language “python CGI”& PHP. Platform to achieve the above-mentioned objective I have used is RHEL 7 and to analyze the performance of the system and to get graphs a tool named as Monitorix is used. Limiting the RAM and CPU and utilizing them efficiently was a major concern. By limiting, the performance of the system can be increased and all the processes can utilize the resources, to achieve this one of the concepts that are used is control groups (cgroups).

Control Groups (cgroups)

Cgroups allow you to allocate resources such as CPU time, system memory, network bandwidth, or combinations of these resources among user-defined groups of tasks (processes) running on a system. You can monitor the cgroups you configure, deny cgroups access to certain resources, and even reconfigure your cgroups dynamically on a running system.

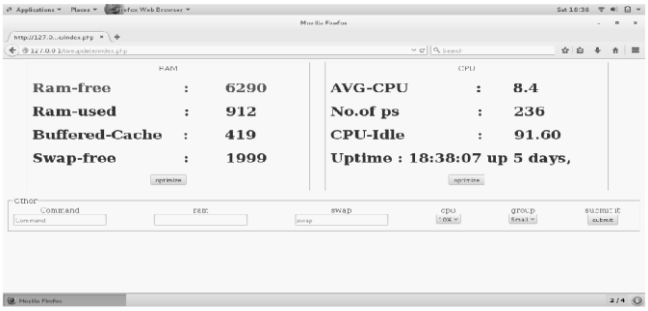
The cgconfig (control group config) service can be configured to start-up at boot time and re-establish your predefined cgroups, thus making them persistent across reboots.



By using cgroups, system administrators gain fine-grained control over allocating, prioritizing, denying, managing, and monitoring system resources. Hardware resources can be appropriately divided up among tasks and users, increasing overall efficiency.

III. PROBLEM IDENTIFICATION

Reasons for the Downtime

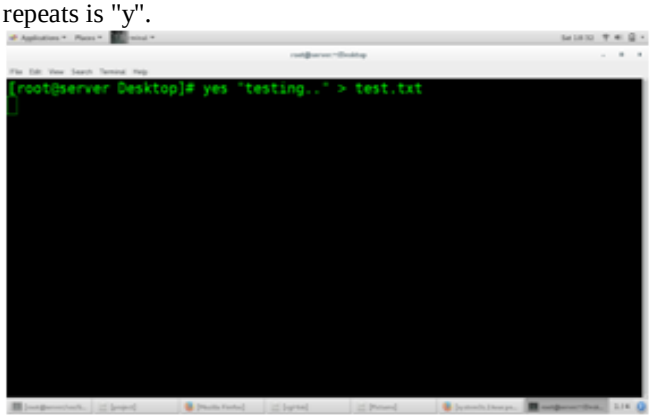


There are major two reasons that may be responsible for the downtime of the virtual machine i.e. due to RAM crash, CPU overutilization or both and that in turn may occur due to high load of the processes on the system to limit them there are several countermeasures that we will discuss one by one.

A. Due to RAM Crash

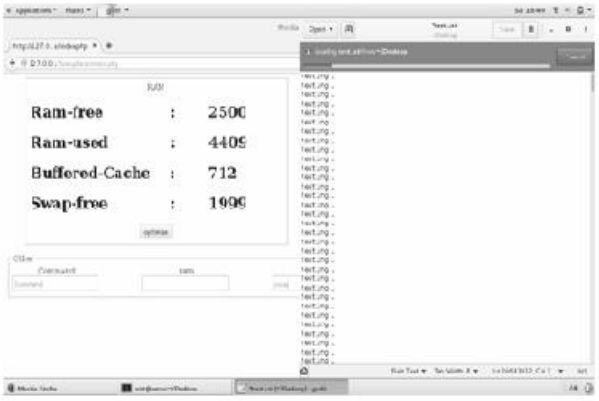
To analyze the reasons for the downtime due to which RAM may crashes I used one of the commands of Linux "YES" command that will print the text written in double quotes and will redirect the print message in a file with extension .txt. The yes command is used to output "y", or whatever word you choose, forever.

The yes command outputs the same string, STRING, in a constant stream. If STRING is not specified, the word it repeats is "y".



As shown in a snapshot that reflects the property of the file we can see this particular text file is taking 2.3GB of RAM and opening this fill with Gedit will result in degradation in free amount of RAM that in turn will result in abruptly stopping of the remaining process running on the system.

Initially, as we can see free RAM is 6290 and 912 is used by the system only at this stage we have swap memory free is 1999

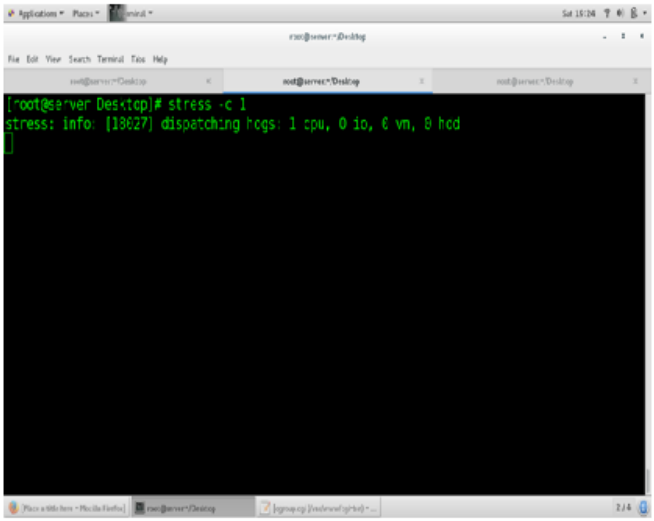


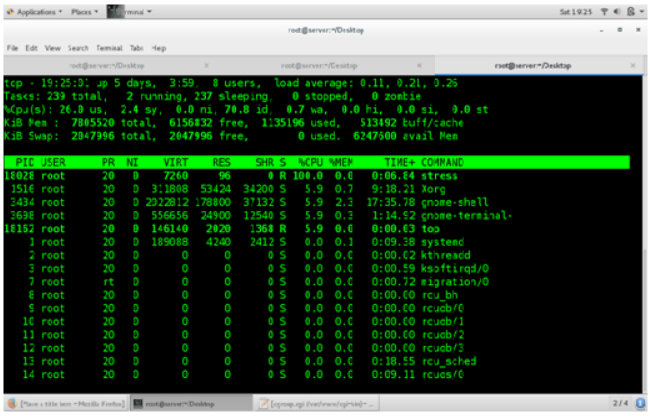
When Opening test.txt with gedit. As we can see, a window is opened in which test.txt is opened from gedit in which the text "testing.." is written and the free RAM is reducing.

While the system is executing the gedit the free RAM is decreasing very gradually and reached 2500 from 6290 it will further decrease but after this stage, the system will get hanged and getting the screenshot after this was not possible.

B. Due to overload on CPU

To analyze the reasons of the downtime of the CPU I have used one of the command of linux that is stress command it will increase the load on the CPU and a point will reach where stress command will utilize 100% of CPU .



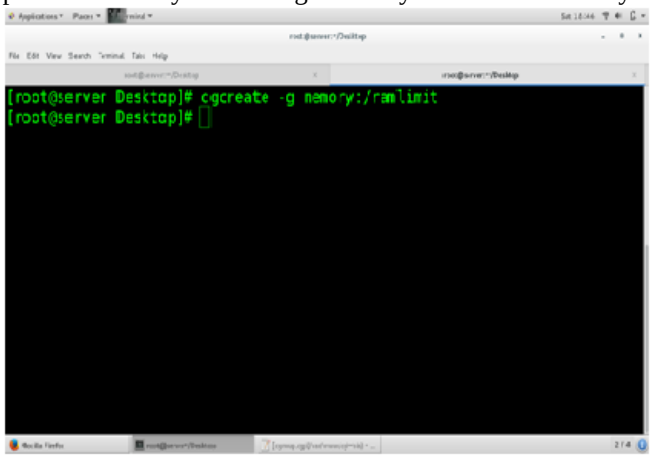


As Stress command is executed we can see the percentage of CPU it is utilizing is 100% that indicates its utilization is more and almost full as compare to other processes due to this a downtime will occur which will resume other processes running and a system will be hanged

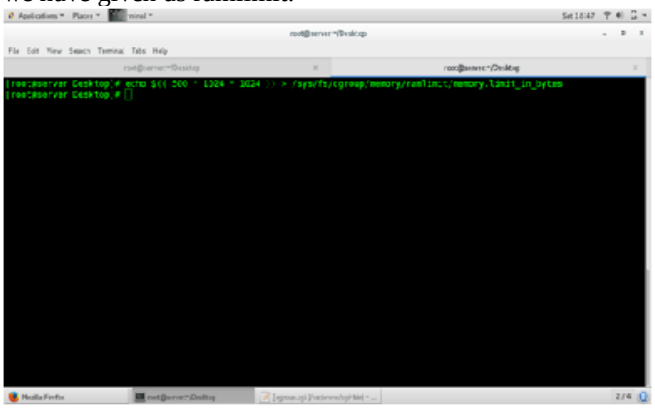
IV. .COUNTERMEASURES FOR PROBLEMS

A. *Countermeasure for RAM Crash:*

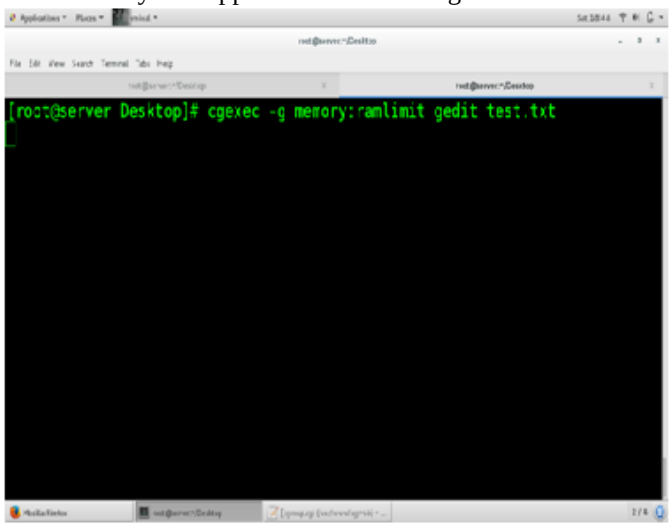
To avoid the unexpected stopping of the process we have to come with the countermeasure of the above problem. And the first countermeasure is creating the cgroups. By creating a cgroups we can limit the resources by giving the quota on RAM so that the given size of the RAM can only be used by the process and after that if a process tries to use the RAM that will be terminated but it will not affect the n number of process that may be running on the system simultaneously.



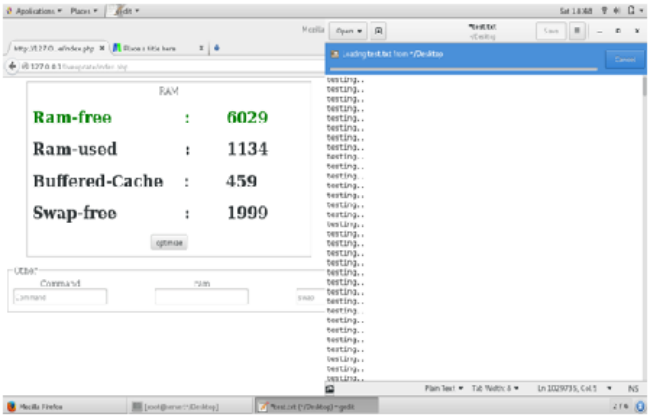
Control group can be created by the command cgcreate in the above snapshot a cgroup is created in memory whose name we have given as ramlimit.



RAM is imposed with a quota of 500 MB, now the process cannot take a size more than this the quota can also be imposed on swap memory so that after the limited size is utilized then swap memory can get started not only this as soon as the limited swap memory is used the process will automatically be stopped without affecting the others.



Finally cgroup is executed by using a command cgexec



Initially, RAM was 6029 and swap memory is 1999 as soon as the test.txt the limited RAM will get started to be used.As the limited RAM is used and the swap memory will be used since the quota is also imposed on the swap as soon as that quota is finished the process will stopped

B. *Countermeasure for CPU Overhead:*

To avoid the above situation we can create the control groups(cgroups) which will limit the usage of CPU so that limited percentage CPU will be given to the process and the system will run smoothly if any of the process is taking more CPU then they will be given the amount that is given in the cgroup.a cgroup is created using cgcreate command to limit the cpu. After creating the control group we are imposing the quota on CPU so that only the specified percentage is used by the process, if the process tries to use more it will be terminated by the above mentioned command we are making changes in /sys/fs/cpu file.

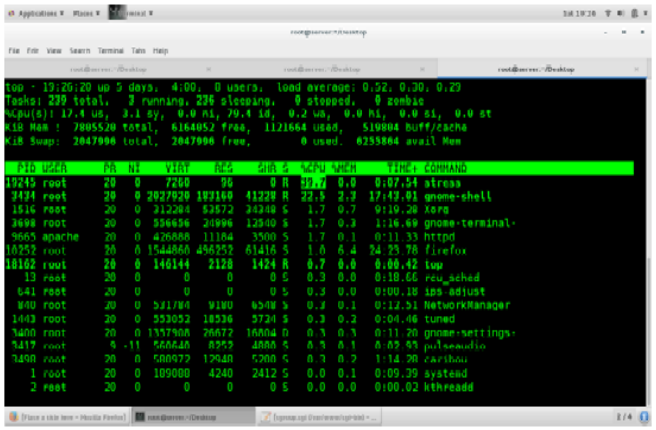
V. RESULTS

A. RAM crash

As we can see in the above graph a point is reached where the RAM usage was suddenly reached to 5GB and system get hanged due to this the system will go down and downtime occurs, all the processes running will abruptly and unexpectedly be stopped.

This graph shows how when we are not using control groups the process was trying to take the all-around 5GB but when the quota is imposed Ram will be utilized that much only after that the process will be terminated and will release the RAM due to this only that process that was responsible for the downtime of RAM will be terminated and the rest will work in an easy flow as they were.

B. *CPU OVERHEAD*



It results that only 39.7%is utilized by stress command previously when the quota was not imposed it was utilizing 100% of CPU so by creating the cgroups over the CPU we can limit its usage so that just because of the single process no other process has to suffer all will be working smoothly and results in Efficient utilization of CPU and RAM.

VI. CONCLUSION

There may be various reasons for the downtime of Virtual machines but considering the major two reasons. This paper presents performance analysis for the both and when the reason was RAM crash control groups were created and for CPU overhead cgcreate create is used to reduce the overhead and to increase the performance.

REFRENCES

1. Salfner F., Troger P., Richly M., "Dependable Estimation of Downtime for Virtual Machine Live Migration", *International Journal on Advances in Systems and Measurements*, **5(1,2)**, 70-88 , (2012)
2. Ramesh D., "Effective Memory Utilization in Cloud Using Domain Application demand Allocation", *Indian Journal of Applied Research*, **4(11)**, 1-3, (2014)
3. Mutlu O., Subramanian L.," Research Problems and Opportunities in Memory Systems", *open access at SuperFri.org*,**1(3)**,19-55, (2014)
4. Ousterhout J., Agrawal P., Erickson D.,Kozyrakis C., Leverich J.,Mazières D., Mitra S.,Narayanan A., Parulkar G.,Rosenblum M., Rumble S.M.,Stratmann E., and Stutsman R., "The Case for RAMClouds: Scalable High-Performance Storage Entirely in DRAM", *Appears in SIGOPS Operating Systems Review*,**43(4),**92-105, (2009)
5. http://blog.scoutapp.com/articles/2014/11/04/restricting-process-cpu-usage-using-nice-cpulimit-and-cgroups
6. https://access.redhat.com/documentation/enUS/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide/sec-Implications_for_Resource_Management.html

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication