ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (369-372), Month: April-June 2014, Available at: www.researchpublish.com

VIRTUAL MACHINE SCHEDULING TO AVOID DEADLOCKS

Rajeev Kumar, Rajiv Ranjan

Deptt.of Computer Science and Engineering, Arni school of computer science, Arni university, HP

Abstract: **The focus of the paper is to represent an algorithm that how the load to virtual machine can be deduced and avoid the dead lock while allocating the processes to virtual machine. In VM while processes are allocate they executes in queue, the first process get resources, other remains in waiting state. As rest of VM remains idle. To utilize the resources, here we purposed an algorithm that will avoid deadlock and utilize each VM successively.**

Keywords: **VM (Virtual machine*).***

1. INTRODUCTION

Cloud computing has attracted attention as an important platform for software deployment, with perceived benefits such as elasticity to fluctuating load, and reduced operational costs compared to running in enterprise data centers. While some software is written from scratch especially for the cloud, many organizations also wish to migrate existing applications to a cloud platform. A cloud environment is one of the most shareable environments where multiple clients are connected to the common environment to access the services and the products [2]. A cloud environment can be public or the private cloud. In such environment, all the resources are available on an integrated environment where multiple users can perform the request at same time. In such case, some approach is required to perform the effective scheduling and the resource allocation.

2. VIRTUAL MACHINE RELATES TO CLOUD COMPUTING

The concept of cloud computing represents a shift in thought, in that end users need not know the details of a specific technology. The service is fully managed by the provider. Users can consume services at a rate that is set by their particular needs. This on demand service can be provided at Cloud service providers are making a substantial effort to secure their systems, in order to minimize the threat of insider attacks, and reinforce the confidence of customers. For example, they protect and restrict access to the hardware facilities, adopt stringent accountability and auditing procedures, and minimize the number of staff who has access to critical components of the infrastructure. any time Nevertheless,insiders that administer the software systems at the provider backend ultimately still possess the technical means to access customers' VMs. Thus, there is a clear need for a technical solution that guarantees the confidentiality and integrity of computation, in a way that is verifiable by the customers of the service. For example, Terra is able to prevent the owner of a physical host from inspecting and interfering with a computation. Terra also provides a remote attestation capability that enables a remote party to determine upfront whether the host can securely run the computation. This mechanism reliably detects whether or not the host is running a platform implementation that the remote party trusts. These platforms can effectively secure a VM running in a single host. However, many providers run data centers comprising several hundreds of machines, and a customer's VM can be dynamically scheduled to run on any one of them. This complexity and the opaqueness of the provider backend create vulnerabilities that traditional trusted platforms cannot address.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 2, pp: (369-372), Month: April-June 2014, Available at: www.researchpublish.com

Process virtual machine

A process virtual machine (also, language virtual machine) is designed to run a single program, which means that it supports a single process. Such virtual machines are usually closely suited to one or more programming languages and built with the purpose of providing program portability and flexibility (amongst other things). An essential characteristic of a virtual machine is that the software running inside is limited to the resources and abstractions provided by the virtual machine—it cannot break out of its virtual environment

3. PROCESS SCHEDULING ALGORITHMS

The algorithm 1[10]

The process scheduling in VMs is done as shown in below algorithm. this makes the scheduling in such a way that the resource utilized to its maximum capabilities as represented in figure 1.

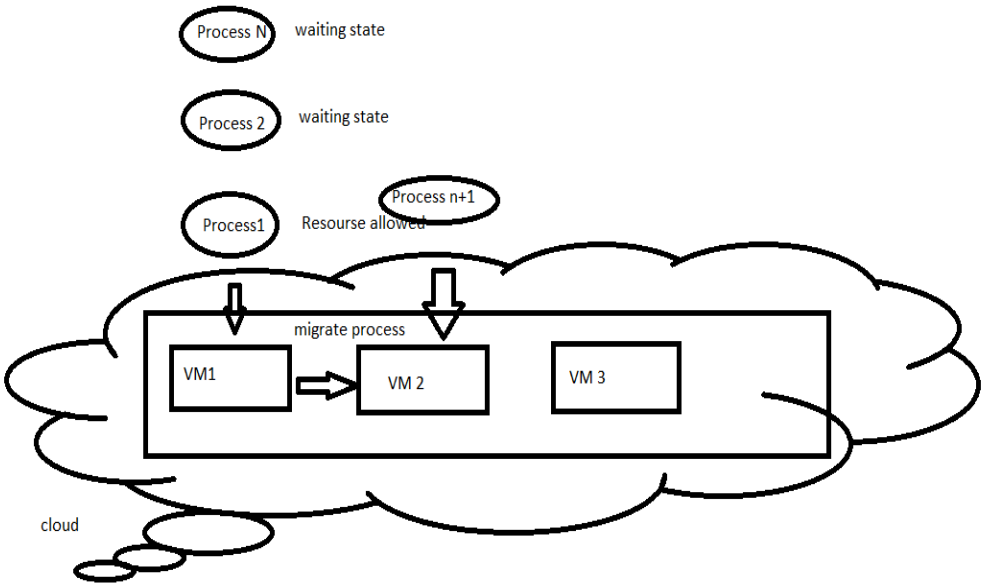

Figure 1 process sheduling in VM one by one

1. Input the M number of Clouds with L number of Virtual Machines associated with each cloud.
2. Define the available memory and load for each virtual machine.
3. Assign the priority to each cloud.
4. Input N number of user process request with some parametersSpecification like arrival time, process time, required memory etc.
5. Arrange the process requests in order of memory requirement
6. For i=1 to N
7. {
8. Identify the priority Cloud and Associated VM having Available Memory > Required Memory(i)
9. Perform the initial allocation of process to that particular VM and the Cloud
10. }
11. For i=1 to N
12. {
13. Identify the Free Time slot on priority cloud to perform theallocation. As the free slot identify, record the start time, process time, turnaround time and the deadline of the process.

14. }
15. For i=1 to N
16. {
17. Iffinishtime(process(i))>Deadline(Process(i))
18. {
19. Print "Migration Required"
20. Identify the next high priority cloud that having the free memory and the time slot and perform the migration of the process to that particular cloud and the virtual machine.
21. }
22. }

The above algorithm is capable to utilize the resource but this algorithm does not use the all virtual machine because while one other machines are working then other remain idle
For other VMs which are idle the new algorithm facilitates all the VMs to utilize for the processes.

The algorithm 2

The below diagram explains that if the processes are allowed to the VMs one by one the each VM will use and no waiting state will exist for any process

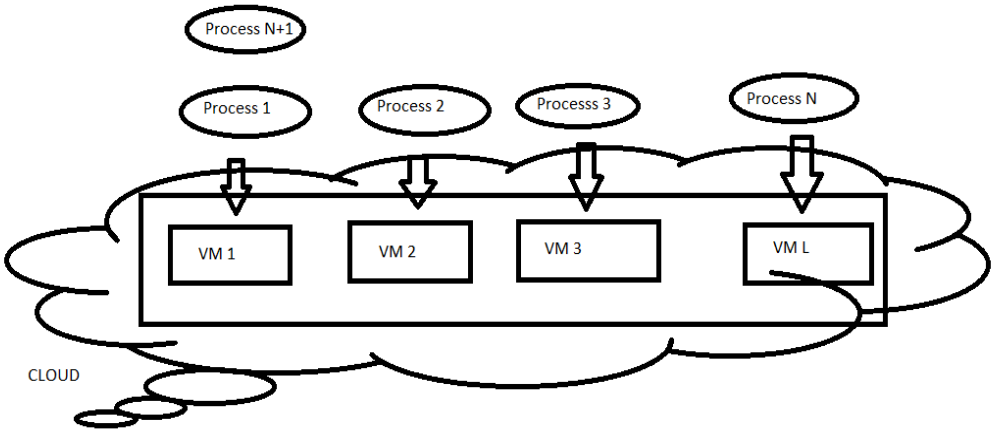

Figure 2 process sheduling for each VM simuntainiously

*a. *Input the M number of Clouds with L1,L2,L3……..,Ln(n tends to no. of last virtual machine))number of Virtual Machines
associated with each cloud.
*b. *Define the available memory and load for each virtual machine.
*c. *Assign the priority to each cloud.
*d. *Input n number of user process request with some parameters
specifications like arrival time, process time, required memory etc.
*e. *Arrange the process requests in order of memory requirement
*f. *For i=1 to n
*g. *{
*h. *Identify the priority Cloud and Associated VM having Available Memory(L1,L2,L3……Ln) > Required Memory(i)
*i. *Perform the initial allocation of process to that particular VM and
the Cloud
*j. *}
*k. *For i=1 to n
*l. *{

*m. *Identify the Free Time slot on priority cloud to perform the allocation. As the free slot identify, record the start time, process time, turnaround time and the deadline of the process.
*n. *}
*o. *For i=1 to n
*p. *(
q. start queue Q1.
*r. *{
*s. Process i1 allocate to VM L1.
*t. *Print "Migration Done"
*u. Process i2,i3……in allocate to VM L2,L3……,Ln respectivally.
w. Q1,i1,p1 ends till i(n+1) allots to L1 again.
*X. * start new Queue Q2[i(n+1)],Q3{i(2n+1)},……. Respectively.
*w. *}
*x. *}

4. CONCLUSION

The above algorithm distributes the process allocation in such a way that process does not concede each other and the waiting state time for process is very much less as well as all resources (VMs, memory) are using efficiently. That means the dead lock occurring chances are very much less.

REFERENCES

[1] ”Anupama Prasanth” Cloud Computing Services: A Survey”,International Journal of Computer Applications Vol. 46, No.3, May 2012, PP.0975 – 8887.

[2] Mical Miller, “Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online”. IJCA (2011) ,PP1113–1122.

[3] B. BazeerAhamed and S. Syed Sabir Mohamed,” Implementation of Trusted Computing Technologies in Cloud Computing”, International Journal of Research and Reviews in Information Sciences ,Vol. 1, No. 1, March 2011,PP .60-68.

[4] Edna Dias Canedo, Rafael Timóteo de Sousa Junior, and Robson de Oliveira Albuquerque,” Trust model for reliable file exchange in cloud computing”, International Journal of Computer Science & Information Technology. Vol.4, No.1, Feb 2012,PP.226-234.

[5] Sean Marston ZhiLia, Subhajyoti Bandyopadhyay Juheng Zhang, AnandGhalsasi “Cloud computing — the business perspective”, Decision Support Systems 51 (2011), PP.176–189.

[6] Flavio Lombardi a, RobertoDiPietro,” Secure virtualization for cloud computing”, Secure virtualization for cloud computing, Journal of Network.

[7]. LeenaKhanna“ Cloud Computing: Security Issues And Description Of EncryptionBased Algorithms To Overcome Them”*International Journal of Advanced Research in Computer Science and Software Engineering,*Volume 3 March - 2013, pp. 279-283

[8] Thomas Weigold, Thorsten Kramp and Peter Buhler*”*ePVM - An Embeddable Process Virtual Machine*”* Annual International Computer Software and Applications Conference (COMPSAC 2007), IEEE.

[9] Robert Balzer,”Process virtual Machine”,USC information Science Institute,PP.37-40 IEEE

[10]GeetaJangra, PradeepVashist, ArtiDhounchak, ”Effective scheduling in Cloud computing is a risk”,International Journal of Advance Research in Computer Science and Software Engineering,volume3,Issue8,August 2013,PP.448-4452.