International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 4– No.9, December 2012 – www.ijais.org

Multi-Core Scheduling in Cloud Computing

Abhishek Kaushik
Computer Science and Engineering Department
Thapar University
Patiala-147004

Inderveer Chana
Computer Science and Engineering Department
Thapar University
Patiala- 147004

ABSTRACT –
I.T. industry keeps on looking for high processing power, the reason behind the introduction of multi-core processors. Cloud computing is also to adopting multi-core processors. Prologue of multi-core processors in cloud computing comes with discrepancies, one of which is resource scheduling. The virtual machine that runs on cloud fights for the core in multi-core processors. This leads to problems like starvation and under-utilization. There are several techniques for multi-core resource scheduling but none of present techniques focuses on resource scheduling of multi-core processors especially in cloud environment. A new approach is proposed in this paper which overcomes the problems which occurs when present multi-core resource scheduling techniques compared in the cloud environment.

General Terms
Cloud computing, multi-core processors, cloud sim

Keywords
Cloud computing, multi-core processors, scheduling

1. INTRODUCTION
With the development of Computer and Internet, new challenges are arising day by day. One of those challenges is increasing in demand of connectivity and data handling.. To overcome these challenges, Cloud Computing emerges as a elastic infrastructure that can be managed according to increase or decrease in resource demand. As being an I.T technology, it needs to change according to the latest I.T trends to cope up with customer demands.

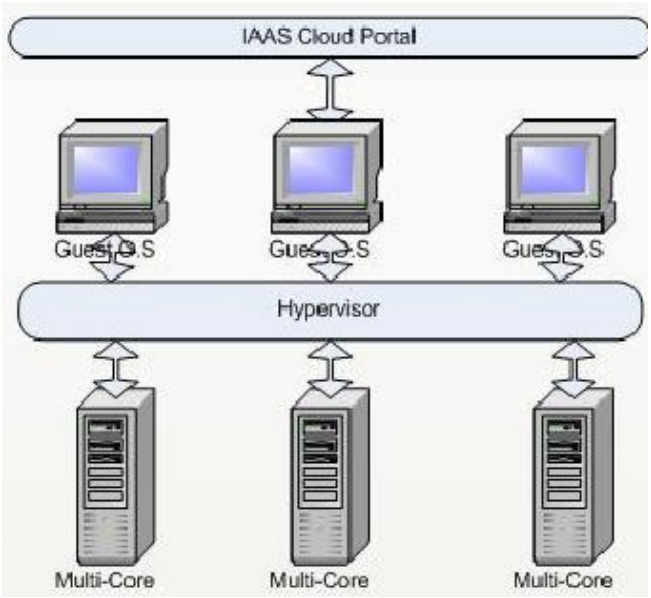

Fig 1: Multi-Core Processors in Cloud Computing

One of these trends is Multi-Core processors. Continues increasing demand of "high processing" [1] pushing single-core processors out of market. So, it becomes mandatory for cloud computing to adopt multi-core processors [2] but this adaptation generate whole new set of problem. The issue here is incompatibility of virtualization [3] and multi-core processors. Virtual machines running on cloud always fight for resources. In case of multi-core processors resources are number of cores. Due to this race of virtual machines for getting maximum number of cores, situations like starvation and underutilization occurs. As virtualization and multi-core both are inseparable part of cloud computing, it is necessary that they must work in harmony with each other.

Further organization of this paper is as, section second discusses related work. In section three a comparison of existing resource scheduling techniques is done. Section four present the proposed technique. Section five is conclusion and future scope followed by section six references.

2. RELATED WORK

There are several researchers are working on the resource scheduling in multi-core processors. Some of these are discussed here which are most relevant to the work we are presenting. Zhiyuan Shao,. Jian,Huang, Hai Jin, Kan presents an Underlying Layout Aware Scheduling (ULAS) [4] technique. They proposed mapping of virtual cpus on physical cpu can be done to according to the workload generated on respected vrtual machines. Jian Huang, Hai Jin, Kan Hu, Zhiyuan Shao [5] presents a resource scheduling technique for cases like threshing .They say that the history of resource utilization can be used to allocate the future resources for using a technique Forecasting and Time Delay Subtraction (FTDS). Y. Zhang, A. Bestavros, M. Guirguis, I. Matta, and R. West suggest [6] changing the Virtual Machine itself in such a way that they can share the resources evenly without fighting for resources. Yong Park, Dongkun Shin [7] explains a similar technique that utilizes the history of resource utilization. In their technique guest operating system utilize the history of resource utilization and feed that data to the virtual machine monitor for the proper resource allocation.

All these techniques discussed above surely presents the effective method for multi-core resource scheduling but none of these techniques have been implemented in the context of cloud computing. Whereas our goal in this paper is to provide resource allocation technique in case of multi-core processors with respect to the cloud computing.

3. PROBLEMS WITH EXISTING TECHNIQUES

The extent of present resource allocation techniques for multi-core processors is only up to a limited number of systems.

Implementation of these techniques in cloud can lead to many problems. Some of these techniques are compared in Table 1.1

with respect to cloud from our previous work [8]. The problems that came on surface are:-

- Modify Guest Operating System

Modifying guest os is not flexible enough for cloud computing. Cloud provides every type of operating system to its customers and not all of them are open source. So this technique cannot be applicable in case of heterogeneous operating systems environment because operating systems like windows exists.

- Modify Virtual Machines

Making modification in virtual machines, so that instead of fighting for resources they themselves "divide them evenly" is also an incompatible technique for cloud. At a single time there are thousands of virtual machine runs on a cloud and it is not easy to modify every single one of them and also the machines that are generated every second on customers demand. And even if it is done, it will be going to add a huge extra overhead to the cloud.

- History of resource utilization

In a live Cloud computing environments there are thousands of resources and there are thousands for application running on them. If cloud providers starting to store resource utilization data for every machine and every application it will become a problem itself due to the enormous amount of data.

Table 1.1 Comparison of existing scheduling techniques [8]

Technique	Test Bed	In Cloud Computing
Underlying Layout Aware Scheduling & Domain based Static priority scheduling	Red Hat Enterprise Linux Server 5.1 Xen 3.4 Unstable	Need to deal with flexibility problem before implementing this technique in Cloud Computing
Forecasting and Time Delay Subtraction Scheme	Red Hat Enterprise Linux Server 5.1 Xen 3.4-unstable	Due to huge number of virtual machines in Cloud Computing, storage of all Past resource utilization data is not entirely possible
Friendly Virtual Machine	User Model Linux Xen	FVM comes with an extra overhead to VM's and in cloud computing the no. of VM's is high enough to decrease overall performance
Task-Aware VM Scheduling	Linux 2.4.20 Xen VMM 4.0.1	Cloud customer can demand for any type of operating system but this technique limits to open source operating system and virtualization software

4. PROPOSED TECHNIQUE

Considering all the problems explained in section 3 of this paper we propose a "Length based resource scheduling". A pseudo code of the designed technique is given below which is used to allocate the applications to the virtaul machines. The algorithm takes vm mips and cloudlet length as an input and returns total execution time. First the algorithm sorts the list of VMs and CloudLet in decreasing order of mips and length respectively. After sorting it alloocates the vm with highest number of mips to cloudlet with gratest lentgh. The algorithm stops by giving the total execution time.

1. **Input:** vm mips, cloudlet length and
   Output: total execution time
2. Foreach vm in vmlist Do
3. Vmlist ← vm.getMips( )
4. Vmlist.sortDecreasingMips( )
5. Foreach cloudlet in cloudletlist Do
6. Cloudletlist ← coudlet.getLength ( )
7. Cloudletlist.sortDecreasinglength( )
8. for vm.getId from 1 to I do
9. for cloudlet.getId from 1 to J do
10. broker.bindCloudletToVm ← Vm.getId
    + Cloudlet.getCloudletId
11. end
12. return total execution time

In the proposed techniques the allocation of virtual machines to the jobs that are requested by user is done on the basis of the length of the job. The "length" of the job can be defined in case of computational applications as " number of variables" that need to process and it is obvious that if the numbers of variables are high the more intense the job is. Similarly in case of data handling the length of the job represent the "amount of data" that need to process. The amount of data can be in GB's or MB's more the amount is more costly the job. Proposed technique assigns virtual machines with high number of Machine Instruction per Second (MIPS) to the job with greatest length. The reason behind this segregation is that, if the applications running on virtual machines already have the maximum possible number of the MIPS available they are not going to fight for the resources. Ultimately the problems like under resource utilization and starvation are going to reduce significantly. Also this technique over comes the problem stated in section 3. It neither requires change in operating system nor in the virtual machines and history of resource is utilization also not required in this technique.

5. EXPERIMENTAL RESLUTS

For the implementation purpose we used cloudsim 2.0. CloudSim [9] is a holistic software framework for modeling Cloud computing environments and performance testing application services. We modified the cloudsim according to proposed technique and compared the results with the First Come First Serve (FCFS) [10] technique.
We performed the experiment using two virtual machines and two jobs for the simplicity purpose. Figure 2 shows a case of ideal scenario when job request of high length and Virtual machine with high number of MIPS created at same time. In Figure red bar represents FCFS and Blue represents proposed technique. Same pattern is followed in further diagrams

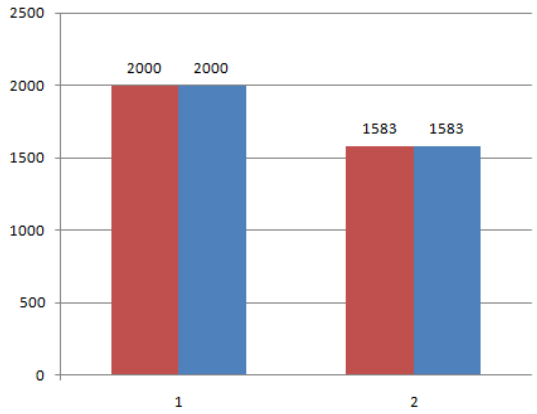

Fig 2: Resource request Ideal Scenario

In such case that job with more length is assigned to the virtual machine with high number of MIPS according to the FCFS.
The total time of completion for both techniques is same.
Occurrences of Ideal case scenario in real cloud environment are very less. Generally, the virtual machine creation and job request comes randomly.

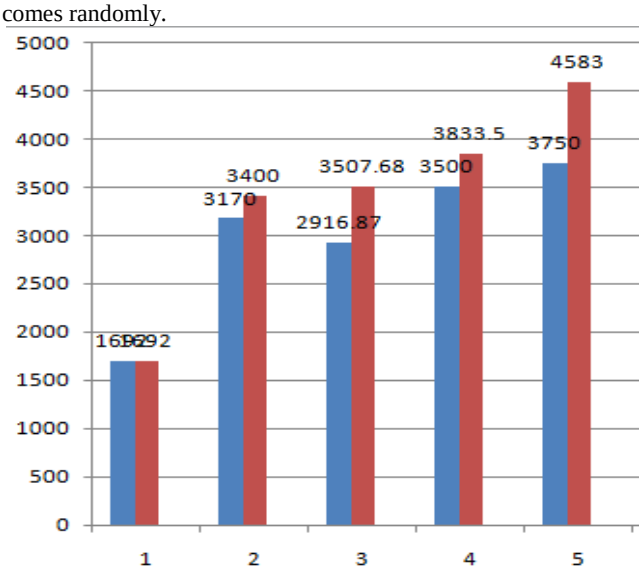

Fig 3: Resource request General Scenario

Such scenario for both FCFS and proposed technique are compared in Figure 3.
It clears from the bar chart in Figure 3 that the total time of execution of two jobs on two virtual machines is less than in proposed technique then FCFS in every case.

6. CONCLUSION AND FUTURE SCOPE

Concept of multi-core processors in cloud computing has not been intensively studied. We have addressed the multi-core resource scheduling problem in accordance to cloud and proposed a technique based on that. Proposed technique not only overcome the addressed challenges but also proved its superiority over first come first serve technique experimentally.
We planning on further investigate the multi-core usability in cloud computing and to embed the service level agreement [11] as a factor for the resource scheduling. Also the further improvement of proposed technique by including more efficient factor for resource scheduling is under process

7. REFERENCES

[1] Gepner, P. ,Kowalik, M.F. ," *Multi-Core Processors: New Way to Achieve High System Performance" ,* International Symposium on Parallel Computing in Electrical Engineering, 13-17 Sept. 2006, pp 9-13
[2] Multi-Core processors available at: http://spectrum.ieee.org/semiconductors/processors/multicore-cpus-processor-proliferation
[3] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R.,Pratt, I.,War eld, A, " *Xen and the art of virtualization.*" In: SOSP '03: Proceedings
[4] Zhiyuan Shao,. Jian,Huang, Hai Jin, Kan Hu , "*XenMVM: in proceedings of Exploring Potential Performance of Multi-Core System in Virtual Machine Environmen"t*, Ubiquitous Information Technologies and Applications (CUTE), 2010 Proceedings of the 5th International Conference on , December 2010
[5] Jian Huang, Hai Jin, Kan Hu, Zhiyuan Shao," *Adjusting Virtual Computing Resources in Threshing Cases* "in Proceedings of Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on Real Time Systems, Pisa, April 2010
[6] Y. Zhang, A. Bestavros, M. Guirguis, I. Matta, and R. West, "*Friendly Virtual Machines: Leveraging a Feedback-Control Model for Application Adaptation*", Proceedings of the 1st ACM/USENIX Conference on Virtual Execution Environments (VEE'05), Chicago, Illinois, pp.2-12, June 2005
[7] Yong Park, Dongkun Shin, "*Task-aware Virtual Machine Scheduling for Multi-core Real-time Systems*" School of Information and Communication Engineering, Sungkyunkwan University, Korea, June 2011.
[8] Kaushik, A. and Chana, I., "Multi-Core: A cloud Perspecticve," International Conference on Recent Trends in Computing, Mechatronics and Communication, at Om Institute of Technology and Management, Hisar, page no. 417 - 420, ISBN 978-81-923446-0-7, February, 2012.
[9] Calheiros, R., Ranjan, R., Beloglazov, A., De Rose, C. and Buyya, R.,"Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.
[10] Dong-Won Park, Swaminathan Natarajan, Arkady Kanevsky, Journal of Systems and Software, Volume: 33, Issue: 1, pp 57-63, 1996
[11] Patel, P., Ranabahu, A. and Sheth, A. 2009. Service Level Agreement in Cloud Computing. Wright State University (Sept. 5. 2009).