A VM SCHEDULING ALGORITHM FOR REDUCING POWER CONSUMPTION OF A VIRTUAL MACHINE IN CLOUD ENVIRONMENTS

S. Hannah Nithya[1], Rose Rani John[2]

[1]PG Student, Department of Computer Science and Engineering, Karunya University, Tamilnadu, India
[2]Assistant professor, Department of Computer Science and Engineering, Karunya University, Tamilnadu India.

Abstract

This paper concentrates on methods which provide efficient processing time of a virtual machine, CPU utilization time of a virtual machine. As the user increases, the performance may be significantly reduced if the tasks are not scheduled in a proper order. In this paper the performance of two already existing algorithms DSP (Dependency Structural Prioritization) algorithm and credit scheduling algorithm are analyzed and compared. A single virtual machine's processing time and CPU utilization time are measured .Satisfactory results are achieved while comparing the two algorithms. This study concludes that the DSP algorithm can perform efficiently than the credit scheduling algorithm.

Keywords: *Virtual Machine, DSP algorithm, credit scheduling algorithm*

----------***----------

1. INTRODUCTION

Cloud computing is a network where large number of computers are connected together. In IT infrastructure cloud computing appears to be very important for the users to organize their applications in a distributed environment. This cloud computing enhances scalability, fault tolerance and availability. The main enabling technology for cloud computing is virtualization. Virtualization is a technology which consists of physical resources and divides the physical resources into virtual resources called Virtual Machine. Virtual machine implements program like a physical machine. Although this technology is proved to be effective there are some challenging issues such as the capability of the server is multiplexed hence measuring per VM power will be complicated. (Peng Xiao et al., 2012). In order to share the applications in the cloud computing atmosphere the performance that is QoS (Quality of service) should be considered. The technique available for energy conservation is to change the active server from power active mode to power saving mode (Zhigang et al., 2012). As the number of users increases the system performance may be reduced if the tasks are not properly scheduled. The operating cost increases unnecessarily. To overcome the difficulties two algorithms such as DSP_Height algorithm and DSP_Volume (Shifa et.al 2012) are analyzed for effective scheduling purpose. The CPU utilization time and the processing time are measured based on these algorithms. This paper is organized as follows. In Section 2 we describe the related work and the general comparison is addressed. In Section 3 we describe the results and discussion where results are analyzed. Finally the work is concluded with a brief discussion.

2. RELATED WORK

Many researchers are involved in developing many methods for power utilization. The power models used for measuring the power consumption is as follows. Bircher et al. (2010) proposed a representation of a full system power consumption named trickledown effect that estimates the time consumption spent on a halted or idle state. Dhiman et al. (2010) presents a model that estimates the power consumption of a virtual machine named VGreen which is applicable for a group of virtualized environment. Kansal et al. (2010) proposes Joulemeter which is a virtual machine metering mechanism. Ala E Husain et al. (2010) proposed VMeter for monitoring of system resources and measuring total power consumption. Hui Chen et al. (2012) addresses PTopW which monitors real time power consumption where a process is running on windows which is a process level power profiling tool. Betran et al. (2010) proposes decomposable power model which estimates the power consumption accurately. Krishnan et al.(2010) proposes the challenges for measuring power on a single virtual machine at its run time.

The scheduling is done to balance and share system resources effectively and achieve a target called quality of service. The need of scheduling arises when the number of users increases and if there is an improper assignment of tasks. Some examples for scheduling algorithm is FIFO (First In First Out), SJF (Shortest Job First Search), RR (Round Robin)

etc. Many researchers are involved in developing efficient algorithms for scheduling in order to reduce the power consumption. They are as follows, Lee et al. (2009) proposes two energy-conscious algorithms namely ECS and ECS makespan in order to reduce the energy consumption. Wang et al. (2013) proposes a scheduling for reducing power consumption of parallel tasks in a cluster with Dynamic Voltage Frequency Scaling (DVFS) technique. Power Aware Task Clustering (PATC) algorithm is used in order to reduce the power consumption. Zhang et al.(2013) presents six energy-efficient task scheduling algorithms with continuous speeds and six energy-efficient task scheduling algorithms with discrete speeds for reducing the power consumption.

2.1 Comparison of Scheduling Algorithms

A basic comparison is made between the characteristics of two algorithms.

Table -1: Basic comparison of algorithms

	Credit Scheduling algorithm	DSP Algorithm
Parameters used	PMC (performance measurement counters) value.	Weightage of jobs.
Approach/method	Using the PMC value and based on the credit the scheduling is done.	Based on the weightage and priority the scheduling is done.
Advantage	Does not trigger the Performance measuring counter values for idle nodes.	1.) Does not trigger value for idle nodes. 2.) Considers dependent of the job.
Disadvantage	Chances for deadlock condition to be occurred.	Not as efficient as greedy algorithms.

2.2 Credit Scheduling Algorithm

Scheduling algorithms are that support quality of service to the users. This credit scheduling algorithm affords a sharing of CPU time among all the virtual machines by a credit based mechanism (Peng Xiao et al., 2012). It utilizes the jobs which execute in less time and it is taken as a credit and schedules it. It uses the performance monitoring counters which measures the system state or activity as credits while scheduling the VM's. It schedules the VM effectively. The VM is first scheduled at time t1 and all the jobs in VM1 are scheduled in order. If a VM is idle it will not trigger any performance value of that particular VM.

2.3 DSP Algorithm

The DSP algorithm is defined as Dependency Structure Prioritization. The DSP algorithm is a process of scheduling. It is very necessary to execute the jobs in order to utilize the time efficiently. Scheduling the jobs contained in a virtual machine. It executes the job with a higher priority first. For example when a job J1 is required to be executed before a job J2, we say that J2 is dependent upon J1. Then J1 is executed first and the other jobs are executed after that job. (Shifa. H,Tim Miller 2012).There are two ways for taking the dependents they are the total number of dependents and the longest path of direct and indirect dependents of the jobs. And the first way is called as DSP_Volume algorithm and the second way is called that DSP_Height algorithm.

2.3.1 DSP_Volume Algorithm

The DSP_Volume algorithm gives the total number of direct and indirect dependents of a job. A higher weight is given to a job which as more dependents. It calculates a matrix which is based on Wars hall's algorithm (Tim Miller et al. 2012). The matrix value is based on I[a,c] : = I [a,c] V I [b,c] where I denotes the indirect dependents that are represented in the matrix. a,b,c are the variables which are taken as job1, job2, job3, job4 and so on depending upon the priority.

2.3.2 DSP_Height Algorithm

The DSP_ Height algorithm gives the total number of direct and indirect deepest dependents. A higher weight is given to a job which has a deepest dependent. It calculates a matrix which is based on Floyd_Warshall's algorithm. The matrix value is based on I [a,b] := max(I[a,b], I [a,c]+I[c,b]) where it takes the maximum value between I[a,b] and I [a,c]+I[c,b]. I denotes the indirect dependents in the matrix and a,b,c are the variables which is taken as job1, job2, job3 and so on.

A general comparison is made between the two DSP algorithms.

Table-2: General comparison of two algorithms

DSP_Volume algorithm	DSP_Height algorithm
▪ Consider more dependents.	▪ Consider deepest dependents.
▪ Takes the OR value between the dependents.	▪ Takes the maximum value between the dependents.
▪ Overall dependents are taken as weight.	▪ Length of the longest path is taken as weight.

3. SIMULATION RESULTS AND DISCUSSION

This session analyzes two different algorithms that are DSP algorithm and credit scheduling algorithms which are compared based on processing time and utilization time.

3.1 Credit Scheduling Algorithm Results

The processing time of a virtual machine is measured based on the scheduling algorithm known as credit scheduling algorithm. The processing time is given in milliseconds. Table 3 shows the processing time of Virtual machines got using credit scheduling algorithm.

Table 3 Processing time of VM's using credit scheduling algorithm

	VM1	VM2	VM3	VM4	VM5
Process-ing Time (ms)	247	371	452	875	212
	VM6	VM7	VM8	VM9	VM10
	815	673	454	124	823

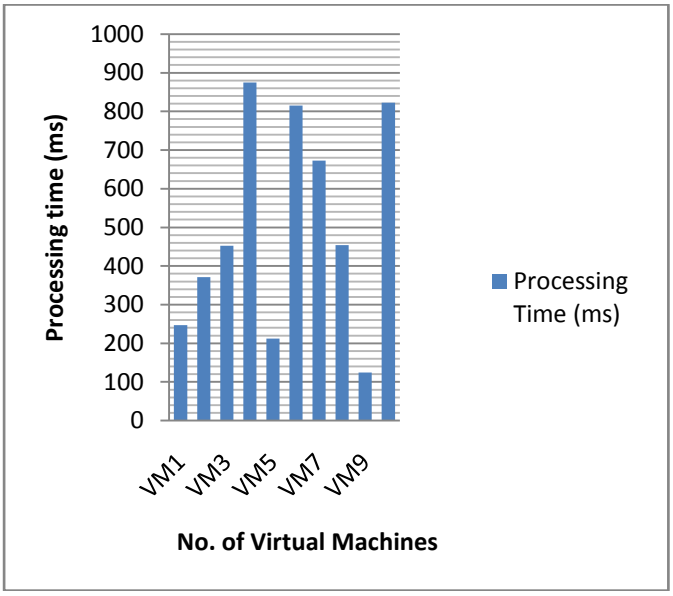

Chart -1: Processing time of VM's

The processing time of each virtual machine from VM1 to VM10 is calculated and denoted in the above diagram. The x-axis denotes the ten virtual machines VM1, VM2, VM3, VM4…VM10 where the y-axis denotes the number of milliseconds taken while processing the job

The CPU utilization time is measured based on credit scheduling algorithm. Table 4 shows the CPU utilization time of virtual machines got using credit scheduling algorithm. The CPU utilization time represented in terms of percentage.

Table 4: CPU utilization time of VM's using credit scheduling algorithm

	VM1	VM2	VM3	VM4	VM5
CPU Utilizat-ion Time (%)	20%	14%	11%	5%	23%
	VM6	VM7	VM8	VM9	VM10
	6%	7%	11%	40%	6%

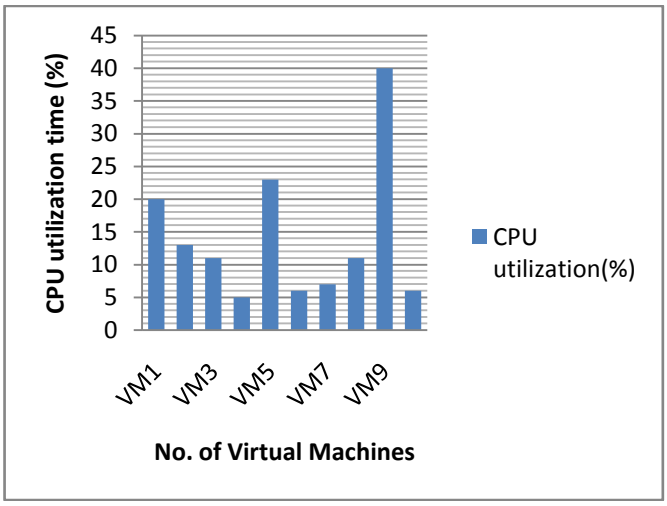

Chart -.2: CPU utilization time of VM's

The CPU utilization time of each and every individual VM is clearly calculated and shown in the above diagram. The y-axis denotes the percentage of the power and the x-axis denotes the ten virtual machines from VM1, VM2, VM3, VM4...VM10.

3.2 DSP Algorithm Results

The scheduling is done based on the weights and dependents of the jobs and the processing time is measured based on the DSP algorithm. Table 5 shows the processing time of each VM based on DSP_Height and DSP_Volume algorithm.

Table 5 Processing time of VM's using DSP_Volume and DSP_Height algorithm

	VM1	VM2	VM3	VM4	VM5
Processi-ng Time (ms)	231	212	213	214	122
	VM6	VM7	VM8	VM9	VM10
	231	235	213	215	221

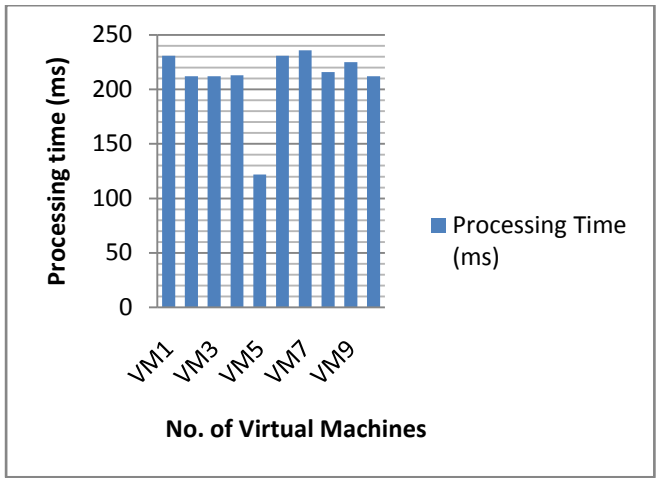

Chart -3: Processing time of VM's

The x-axis denotes the number of virtual machine and its processing time is denoted in y-axis in terms of milliseconds.

Table 6 CPU utilization time of VM's using DSP_Volume and DSP_Height algorithm

	VM1	VM2	VM3	VM4	VM5
CPU Utilization Time (%)	38%	34%	24%	36%	23%
	VM6	VM7	VM8	VM9	VM10
	27%	25%	30%	40%	26%

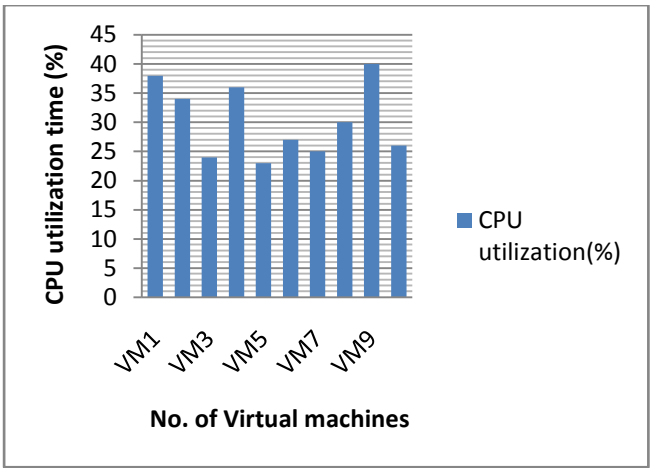

Chart -4: CPU utilization time of VM's

The CPU utilization time of each Virtual Machine is calculated and given by using DSP_Volume algorithm and DSP_Height algorithm. The x-axis denotes the number of virtual machines and the y-axis denotes the percentage of CPU usage. From the above results it shows that DSP algorithm performs better than the credit scheduling algorithm.

3. CONCLUSIONS

This paper mainly focused on scheduling algorithms which effectively calculate the CPU utilization time and processing time for many virtual machines that are used by many users. Two different algorithms namely DSP algorithm (Shifa et al 2012) and credit scheduling algorithm (Peng Xiao 2012) were analyzed and compared. From the obtained results it concludes that DSP algorithm performed efficiently and decreases the processing time effectively than the credit scheduling algorithm.

REFERENCES

[1]. Bircher WL., John LK.,(2012) "Complete system power estimation using processor perfor- mance events". IEEE Transactions on Computers;61(4):563–77.

[2]. Bertran R., Gonzlez M., Martorell X., Navarro N., Ayguad E.,(2010)"Decomposable and responsive power models for multicore processors using performance counters". In: Proceedings of annual international conference on supercomputing. Ibaraki, Japan: ACM Press;. p. 147–58.

[3]. Bohra A., Chaudhary V.,(2010)"Vmeter: power modelling for virtualized clouds". In: Proceedings of IEEE international parallel and distributed processing symposium. Atlanta, USA: IEEE Press; p. 1–8.

[4]. Chen H, Li Y, Shi W. Fine-grained power management using process-level profiling. Sustainable Computing: Informatics and Systems 2012;2(1):33–42.

[5]. Dhiman G., Marchetti G., Rosing T.,(2010) "vgreen: a system for energy-efficient manage- ment of virtual machines". ACM Transactions on Design Automation of Electronic Systems;16(1):1–27.

[6]. Figueiredo J., Maciel P., Callou G., et al.(2011) "Estimating reliability importance and total cost of acquisition for data center power infrastructures". In: Proceedings of IEEE international conference on systems, man, and cybernetics. Piscataway: IEEE Press;. p. 421–6.

[7]. Kansal A, Zhao F, Liu J, Kothari N, Bhattacharya AA. Virtual machine power metering and provisioning. In: Proceedings of ACM symposium on cloud computing. Indianapolis, IN, USA: ACM Press; 2010. p. 39–50.

[8]. Krishnan B., Amur H., Gavrilovska A., Schwan K.,(2010) "Vm power metering: feasibility and challenges". ACM SIGMETRICS Performance Evaluation Review;38(3): 56–60.

[9]. Shifa H ., Tim Miller (2012) "Using Dependency Structures for prioritization of functional test suites". IEEE transactions 2012

[10]. Peng Xiao., Zhigang Hu., Dongbo Liu et al. (2012) " Virtual machine power measuring technique with bounded error in cloud environments". Elsevier 36(2013) 818-828 .

[11]. Y.C. Lee, A.Y. Zomaya, "On effective slack reclamation in task scheduling for energy reduction", Journal of Information Processing Systems 5 (4) (2009) 175–186.

[12]. L.M. Zhang, K. Li, D.C. Lo, Y. Zhang, "Energy-efficient task scheduling algorithms on heterogeneous computers with

continuous and discrete speeds", Sustainable Computing: Informatics and Systems 3, 2013, 109-118.
[13]. L. Wang, S.U. Khan, D. Chen, J. Kolodziej, R. Ranjan, C. Xu, A. Zomaya, "Energy-aware parallel task scheduling in a cluster", Future Generation Computer Systems, Volume 29, Issue 7, 2013, Pages 1661-1670.