ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

A Metaphorical Research on Diversified Approaches of DVS Enabled Energy - Aware Cloud Data Centers under the Cloudsim Platform

C.Saranya[1], Dr. R. Manicka Chezian[2]

Research Scholar, Department of Computer Science, NGM College, Pollachi, India.[1]

Associate Professor, Department of Computer Science, NGM College, Pollachi, India.[2]

ABSTRACT: Energy has turn into one of the majority critical issues. Energy-efficient designs encompass engage in recreation significance roles for hardware as fine as software implementations. Energy-efficiency can be achieved by adopting the dynamic voltage scaling (DVS) scheme. DVS has been a solution scheme in exploiting to reduce energy dissipation by lowering the supply voltage and operating occurrence. In this dissertation we anticipated diversified approaches on DVS enabled Energy aware Cloud Data centers. DVS has implemented on every approaches with different mechanisms and it provides energy saving and different advantaged features of the system. The scrutiny of research would give the instruction concerning what has been done previously in the same area, what is the in attendance tendency and what are the other related areas, Comparison of algorithms imposed with DVS features under the Cloudsim. The analysis has resultant that the DVS has reduced the power and improve the system utilization.

KEYWORDS: Cloud Computing, Cloudsim, Data Centers, DVS, Energy (Power)-Aware, Power Consumption.

I. INTRODUCTION

Cloud computing are a vast collection of merely obtainable virtualized property such as Hardware, Software progress platform and services. The Cloud Computing skeleton is collected of five important characteristics, three SPI models, and four deployment models." There are four deployment types of clouds, according to the NIST (U.S. National Institute of Standards and Technology).They is:

- **Public clouds –** easily reached by any user/consumer
- **Private clouds -** clouds easy to get to only to a single business or group
- **Community clouds -** division of group of customers with needs are in common
- **Hybrid clouds -** combinations of the private, public, community types**.**

Service models are **SaaS** (Software as a Service), **PaaS** (Platform as a Service) and **IaaS** (Infrastructure as a Service). Structural designs for cloud use special components, in constituent based upon the services they are using in the cloud. These IDC containing thousands of operation computational nodes as complete as virtualization technology using many virtual machines (VMs).[1][2]

Energy-aware DVS are solutions that can be used at special levels, allowing the minimization of the power burning up of every host in a data center. Finally, DVS (Dynamic Voltage and Frequency Scaling) [3] can dynamically change the voltage and frequency of a CPU of a host according to its load. DVS can be activated in five different modes: Performances, Power save, User- Space, Conservative, and On Demand. This manuscript focuses on power-aware administration DVS approach under the Cloudsim platform. In section 2 Cloudsim with the features of the DVS used in Cloudsim. In section 3.anaysis of DVS based algorithms and its explanation. In section 4 analysis of DVS with Tabular Content and what are all the authors anticipated DVS Schemes with merits and demerits. Section 5, Analysis of methods with DVS and without DVS. In section 6, conclude the paper with a summary and directions for the future work.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

II. STATE OF THE ART

2.1 Simulator - Cloudsim

Cloudsim [4] is a toolkit for model and mock-up of Infrastructure as a Service (IaaS) cloud computing environments. It allows users to portray the individuality of data centers, including number and characteristics of hosts, to be had storage space, network topology, and patterns of data centre's usage. Not any of the existing open resource simulators contains a form for DVS.

2.2 DVS in Cloudsim

The central part features for DVS simulation were additional to a new wrap up, called DVS. In the simulator, a frequency modifies directly impacts the ability of CPU's, measured in MIPS. The build-up of DVS in a straight line affects the presentation of the CPU capability (and hosts), which are subject to usual changes throughout simulations. The inclusion of DVS in Cloud Sim required changes in the running of events performed by the simulation core. In fact, the Power Save mode induces a delay in the implementation of tasks as the CPU operates at its lower frequency. Underneath this situation, the development of chronological events should be as the crow flies linked with the end of every one event.

Table 1: Comparison of Various Cloudsim

Cloudsim	Platform	Programming Lang	Networking	Simulator Type	Availability
Cloud Analyst	Cloudsim	Java	Limited	Event Based	Open Source
Green Cloud	NS2	C++/OTCL	Full	Packet Level	Open Source
Network Cloudsim	Cloudsim	Java	Full	Packet Level	Open Source
MDCSIM	CSIM	C++/Java	Limited	Event Based	Commercial

Table 2: Comparison of Various Cloudsim

DVS MODE		CLOUD SIM		ERROR
DURATION	ENERGY	DURATION	ENERGY	
1442	15	1442	15.45	0.45
1400	14.5	1400	15.34	0.84
1460	16	1460	15.75	0.25
1470	17	1470	18	1
1480	18	1480	18.5	0.5

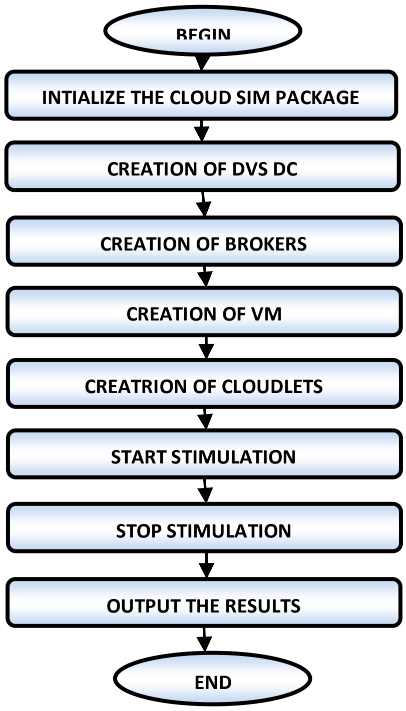

Figure 1: Cloudsim Life Cycle

DOI: 10.15680/ijircce.2015. 0307041

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

III. DIVERSIFIED APPROACHES ON DVS ENABLED ENERGY – AWARE DATA CENTERS

3.1 Classification of DVS Algorithms

Noteworthy amount of investigate has been done in the pasture of DVS. A bulky number of algorithms have been premeditated which can be broadly off the record into three categories. This classification was made by Yuan et al in [5] as follows:

(i) Real-time DVS (RT-DVS): RT-DVS algorithms differ in their techniques to utilize the static slack available due to the low CPU utilization of the application or dynamic slack obtainable due to the actual implementation time being much lesser than the worst case execution time of the real-time application.
(ii) *General Purpose DVS (GP-DVS)* : GP DVS algorithms are based on two techniques - prediction and speed setting. Prediction involves predicting the future workload while speed setting involves deciding the speed at which to run. These are interval based schedulers where the prediction and speed setting decisions are made for every interval so as to minimize the idle time in that interval.

(iii) *Statistical DVS:* Both GP-DVS and RT-DVS algorithms are not well-matched for multimedia applications as their run time stipulate varies. Statistical DVS algorithms have been intended to deal with run time stipulate variations. The core idea of Statistical DVS is to change the CPU speed based on the demand distributions of the applications. These algorithms engage either online or off line profiling of the applications to get the probability giving out of the cycle demand of the applications and make scheduling and frequency scaling decisions based on that.

3.2 Dynamic Voltage Frequency Scaling (DVS)

Dynamic voltage Frequency scaling (DVS), previously incorporated into many topical processors is perhaps the greater part appealing method for reducing energy consumption. DVS reduces energy consumption of processors based on the fact that such power utilization in CMOS circuits has a direct relationship with (1) working frequency and (2) the square of the supplied voltage. Thus, DVS saves energy by switching between processor's voltages/frequencies to execute errands during *slack* times. Although DVS was in the beginning planned for task scheduling on single processors and not long been extended and used in parallel and distributed computing systems as well [6, 7].

To organize DVS, it have got to be properly incorporated with a job scheduler by using one of the following two approaches:(1) Throughout the scheduling process or (2) slack reclamation after scheduling. In the first approach, tasks graph are planned on DVS enabled processors by minimizing in concert energy and make distance at the same time [8-9]. In the second approach, an independent scheduler is primary used to distribute tasks among processors without considering energy consumption. The procedures are followed by an independent DVS method to minimize energy consumption of tasks by filling the generated tasks' slack times.

3.3 Real-Time DVS (RT-DVS)

RT-DVS that adapt the OS's Real-time scheduler and task management examine to afford noteworthy energy reserves while maintain real-time time limit guarantees. We demonstrate through simulations and a working prototype completion that these RT-DVS algorithms intimately approach the hypothetical lower bound on energy consumption, and can with no trouble decrease energy consumption 20% to 40% in an embedded real-time scheme.

3.3.1 Cycle-Conserving RT-DVS (CC-DVS)

Even though real-time responsibilities are particular with worst-case calculation necessities, they in the major use to a great extent less than the worst case on most invocations. To take best benefit of this, a DVS mechanism could shrink the in service frequency and voltage when responsibilities use less than their worst-case time portion, and add to frequency. When a job is released for its next incantation, we can't know how much computation it will in fact require, so we must make the conservative supposition that it will need its specified worst-case processor time. we can devise DVS algorithms that avoid wasting cycles (hence "cycle conserving") by plummeting the operating frequency. This is rather similar to slack time stealing [10].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

3.3.2 Look-Ahead RT-DVS (LA-DVS)

The most aggressive RT-DVS algorithm that we introduce in this manuscript attempts to achieve even better energy savings using a look-ahead technique to decide future computation need and defer task execution. look-ahead scheme tries to postpone as much work as possible, and sets the operating frequency to meet the minimum work that have to be done now to make sure all prospect deadlines are met. Of course, this may well perhaps necessitate that we will be required to run at high frequencies later in order to whole all of the deferred work in time.

3.4 Reference Dynamic Voltage-Frequency Scaling (RDVFS)

RDVS is a simplified version of the algorithm introduced by Kimura et al in [7] for power-scalable high performance clusters at the bottom of DVS. It reduces the power utilization of processors by selecting the negligible obtainable processor incidence competent of concluding a job in a known time casing. Which is the primary occurrence larger than finest frequency considered the best discrete frequency candidate to carry out the job within the known time casing and covering its related drooping time.

RDVS algorithm: *slack reclamation by one frequency*
Input: *the scheduled tasks on a set of P processors*
1. ***for*** *task* $A^{(k)}$ *scheduled on processor* P_j
2. Compute the optimum continuous frequency $(f^{(k)}_{Opt-cont})$
3. Pick the closest higher frequency to $(f^{(k)}_{Opt-cont})$ *in the CPU frequency set,*
4.$t^{(k)}_{RDVFS} = \frac{(f^{(k)}_{Opt-cont})}{f^{(k)}_{RDVFS}} T^{(k)}$
5.$E_{RDFS} = f^{(k)}_{RDVFS} \; t^{(k)}_{RDVFS} + P_{Idle}$
6. end for
7. return $(f^{(k)}_{RDVFS}, f^{(k)}_{RDVFS})$ for all tasks

Figure 2: RDVS Algorithm

3.5 Maximum- Minimum Frequency for Dynamic Voltage-Frequency Scaling (MMF-DVS)

Maximum-Minimum Frequency for Dynamic Voltage- Frequency Scaling (MMF-DVS) technique on hand in [11] is comparable to RDVS as together of these approaches use DVS to decrease energy consumption of scheduled needy tasks in clusters. Unlike RDVS algorithm which applies only one frequency to execute a job, MMF-DVS uses a linear combination of maximum and minimum processor frequencies to attain the optimal energy consumption regarding to slack time of the task.

3.6 Multiple Frequency Selection for Dynamic Voltage-Frequency Scaling (MFS-DVS)

The key thought is to carry out each task with a linear combination of additional than one frequency such that this combination consequences in using the lowest energy by covering the whole slack time of the task. We have knowledgeable our algorithm with in cooperation random and real-world application job graphs and compared with the consequences in preceding research in [7] and [11]. The RDVS algorithm decreases task implementation power by choosing the most excellent processor's speed with respect to the task's unoccupied time [7]. The solution idea of MFS-DVS is to carry out tasks by means of a linear grouping of obtainable frequencies so that their slack times are fully filled / enclosed MFS-DVS can be defined. Contrast with R-DVS and MMF-DVS it produces the most excellent results.

MFS-DVS algorithm: *linear combination of frequencies*
Input: *the scheduled tasks on a set of P processors*
For *kth-task A(k) scheduled on processor*
Apply RDVS algorithm on this task
if $\boldsymbol{E^{(k)}_{RD}}$ *for this task* ***then***
- This task is eligible for MFS-DVS
- Solve optimization problem in by linear programming
else
RDVS is the optimal result
4. end if
5. end for
6. return *(the voltages and frequencies of optimal execution of* the task)

Figure 3: MFS-DVS Algorithm

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

3.7 Simplified-Multiple Frequency Selection DVS (SMFS-DVS)

In most DVS algorithms, it is assumed that processor energy consumption is a convex function of frequency (or voltage) as: The convex function relationship between power and voltage was used by Ishihara in [12] where CPU power is just a square function of voltage – not frequency. This problem has been solved in the MVFS-DVS algorithm in this chapter by considering a general form between power and voltage-frequency in CPUs. The MVFS-DVS algorithm claims that independent of the method of modelling the relationship between power and voltage-frequency.

SMFS-DVS algorithm: Post-processing algorithm to optimally energy consumption of scheduled tasks
Schedule tasks given by a DAG using an scheduling algorithm
for k=1:*number of tasks in DAG*

- Select the kth task
- Calculate $f_{ideal}^{(k)}$
- Select the immediate frequencies in the processor frequency set before and after $f_{ideal}^{(k)}$. These frequencies are $f_{RD}^{(k)}$ and $f_{RD-1}^{(k)}$.
- Calculate associated times from and energy of task
- Select $(f_{RD}^{(k)}, f_{RD-1}^{(k)})$ associated to the lowest energy for this task

End for
return (individual frequencies pair for execution of each task)

Figure 4: SMFS-DVS Algorithm

3.8 Multiple Voltage-Frequency Selection for Dynamic Voltage-Frequency Scaling (MVFS-DVS)

Multiple Voltage-Frequency Selection DVS (MVFS-DVS), the key in scheme is to implement errands using a linear mixture of obtainable frequencies so that all slack times be completely utilized. The MVFS-DVS [13]algorithm is obtainable in three steps. Firstly, energy expenditure of each task is formulated as an optimization problem with constraints. Secondly, formal proofs are provided to demonstrate that the most favourable set of at most two voltage-frequencies will forever lead to smallest amount energy expenditure.

MVFS - DVS algorithm
Post-Processing algorithm to optimize energy consumption of scheduled tasks
Schedule tasks given by a DAG using a scheduling algorithm
For k=1:*number of tasks in* DAG
-select the kth -task
-calculate $f_{ideal}^{(k)}$
-Divide processor frequency set into two groups (U,L):

$$\frac{[f1, \ldots, fr]}{U} > f_{ideal}^{(k)} > \frac{[fr+1, \ldots, fn}{L}$$

-Calculate time and energy for all
$f_i \in U$ and $f_i \in L$
-select (f_i, f_j) associated to the lowest energy for this task
End for
Return (individual voltage-frequency pair execution of each task

Figure 5: MVFS-DVS Algorithm

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

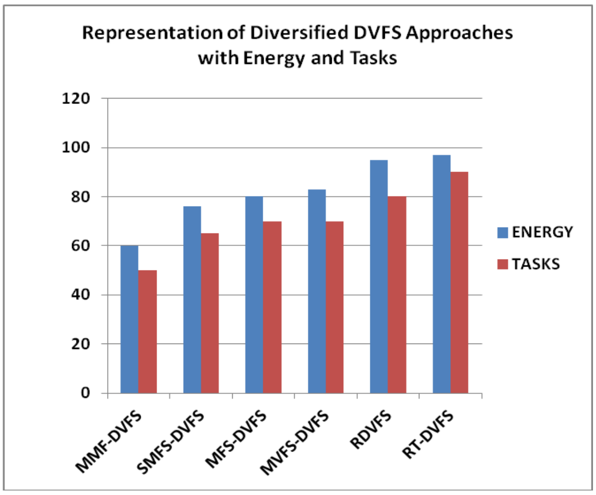

Figure 6: Representation of Diversified DVS Approaches with Energy and Tasks

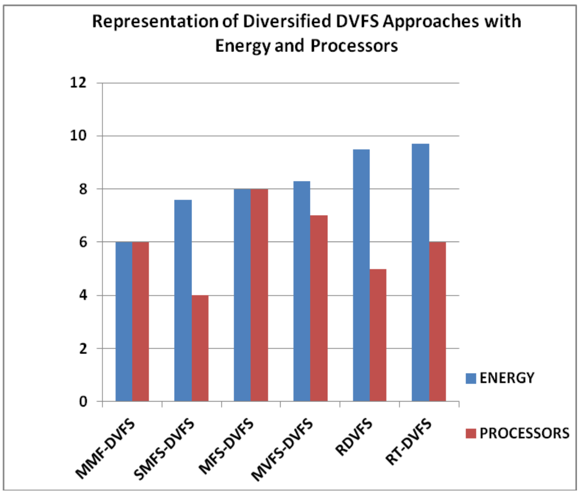

Figure 7: Representation of Diversified DVS Approaches with Energy and Processors

I. REPRESENTATION AND COMPARISON OF DVS METHODS WITH FEATURES, MERITS AND DEMERITS

Khargharia et al. [14] - Optimize power and performance (performance/watt) of data centers at every level of the hierarchy at the same time as maintaining scalability. They adopt a mathematically rigorous optimization move toward to provide the request with the required amount of memory at runtime.

Kim et al. [15] - Investigate power-aware provisioning of VMs for real-time services and show the performance results through simulation. Similar studies are conducted in and Pillai and Shin present a class of novel real-time algorithms to modify the OS task management policies, thereby providing significant energy savings while meeting real-time deadlines. They also build a working prototype and employ simulations to obtain quantitative results.

Shin et al. [16] propose SimDVS, a unified simulation environment for evaluating DVS and in attendance the assessment results.

Zhang et al. [17] - Current optimal procrastinating DVS for hard real-time systems by using stochastic workload information. Simulation results show up to 30% energy savings for single-task workloads and 74% for multi task workloads over a baseline case by means of a constant worst-case execution voltage.

Lee et al. [18] - At hand a novel on-line DVS algorithm. It combines the underlying OS task running mechanism with a real-time scheduler and employs OS-level statistics to decide DVS policies. Simulation results are obtained and used to validate the effectiveness of their proposed algorithm.

Seo et al. [19] - Proposition a novel DVS strategy that dynamically balances the task loads of multiple cores to optimize power consumption throughout execution and adjusts the numeral of active cores to diminish leakage power consumption below low load conditions. However, the on top of measurement and simulation-based methods has their limitations.

1) They necessitate widespread and thorough experiments and measurements.
2) They cannot cover the entire parameter space and only a part of points in such space can be simulated or tested.
3) They are applicable to implemented systems or system prototypes only.

Hence, it is impossible for them to perform analysis at earlier life-cycle phases.

Garg et al. [20] - Regard as a number of energy efficiency factors (such as energy cost, carbon emission rate, workload, and CPU power efficiency) as inputs of a dispersed DVS-enabled data center energy optimization model. Through logical assessment, they demonstrate that their wished-for scheduling policies are able to achieve on standard up to 25% of energy savings in assessment to profit based scheduling policies that attain the similar presentation.

Mezmaza et al. [21] - Recommend a parallel bi-objective hybrid genetic optimization framework so as to takes into account both build span and energy consumption for DVS-enabled cloud.

Alenawy and Aydin [22] - Present a method to intend a stochastic model by means of discrete time Markov chains and consider the minimization of control consumption with given presentation constraints. However, they be unsuccessful to take into account workstation breakdown and recovery.

Table 3: Comparison Table of DVS-Enabled methods, Functionalities, Merits and Demerits

Authors	Methods	DVS	Explanation	Energy	Algorithms	Pros	Cons
B. Khargharia Et Al.,	Hierarchical control and Performance Management	Yes	Optimize for control & routine	Yes, 65% saving	Min-Min Heuristic Algorithm	To deal with control & recital	Runtime complication & operating cost.
K. H. Kim, A. Beloglazov, and R. Buyya .	DVS-Enabled HRT-VM Provisioning	Yes	Real-time overhaul as a real-time VM request & To provision VM in Cloud DVS.	Yes	Min-price HRT-VM provisioning & Min-price profitable SRT-VM provisioning	Reduce power consumption and increase their profit	Impact of the cooling systems
S. U. Khan.	Multi-objective Self-adaptive Weighted Sum Technique	Yes	Power- aware reserve portion compared to greedy in addition to linear relaxation heuristics.	Yes	AWS, greedy, in addition to LR heuristics.	AWS, greedy, along with LR heuristics.	Exploring troubles of forceful natural world.
P. Pillai and K. G. Shin.	Real-Time DVS (RT-DVS)	Yes	Real time energetic voltage scaling coupled with OS task supervision mechanisms.	Yes,40 % energy saving	RT-DVS algorithm, cycle-conserving EDF algorithms	Indicate that 20% to 40% power savings can be achieved	DVS with probabilistic or numerical time limit & Guarantees
D. K. Shin, W. S. Kim, J. K. Jeon, J. H. Kim, and S. L. Min.	Sim DVS, a unified simulation environment	Yes	Sim DVS of the Intra DVS algorithm and Inter DVS algorithms.	Yes	DVS Algorithms for Hard Real-Time Systems, Heuristics for Hybrid DVS algorithms	Overhead such as the context switches & The comparative energy efficiency.	Not specified
Y. Zhang, Z. J. Lu, J. Lach, K. Skadron, and M. R. Stan	Optimal procrastinating DVS (OP-DVS) method	Yes	Optimal procrastinating voltage scheduling (OP-DVS) for hard real-time systems using stochastic workload information.	Yes, 30% - 74% saving	Single-Task OP-DVS Algorithm, Local OP-DVS Algorithm, Global OP-DVS Algorithm	Optimal in terms of energy minimization with no deadline misses	Need Improvement of proposed methods
C. H. Lee, C. N. Nat, and K. G. Shin.	Dynamic Priority Scheme for Pinwheel Task Model	Yes	Dynamic Priority Scheme for Pinwheel Task Model	Yes, 10-80% saving	A power-aware scheduling algorithm	Integrating static and dynamic scheduling more effective in power saving	Higher priority until the ends of deadlines
E. S. Seo, J. K. Jeong, S. Y. Park, and J. W. Lee	Energy - efficient approach to scheduling periodic real-time tasks	Yes	straightforward VS selects the utilization of the Heaviest - weighed down core in a VFD	Yes, 22% saving	Hyper period-Based Multi core Voltage Scaling Scheduling algorithm	The consumption of the heaviest-loaded core in a VFD..	On-line schedule trade by means of the dynamic slack reclamation.
S. K. Garg, C. S. Yeob, A. Anandasivamc , and R. Buyyaa	High Performance Computing (HPC)	Yes	Near-optimal scheduling policies so as to exploit heterogeneity crossways several data centers	Yes, average up to 25% of energy savings	Meta Scheduling Algorithms	HPC users need the capability to increase fast and scalable access	Server on/off, workloads management
M. Mezmaza et al.,	Parallel Bi-objective Hybrid Met heuristic method	Yes	parallel applications are by means of the sole goal of minimizing completion time	Yes, 47.5%	Energy-conscious ,heuristic, parallel bi-objective hybrid genetic algorithm	To diminish the completion time	Poor system performance.
T. A. Alenawy	Energy-Constrained	Yes	RT development aims to	Yes	Greedy,	“greedy” to option the	supposed

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

and H. Aydin.	Weakly-Hard Real-Time development Problem		make the most of the scheme presentation		Energy-density algorithms	workload a priori, energy price in addition to deadlines	rapidity collection and on-line slack reclamation techniques

IV. COMPARISON OF DVS AND NON - DVS METHODS

Our research study presents efficiently maximize the resource utilization and results in significant reduction in power consumption for executing the jobs. Experimental results show DVS reduces more energy consumption than Non DVS scheme. The primary objective of this research work is to consolidate the Dynamic Voltage Frequency Scaling (DVS) technique in an efficient way to maximize the resource utilization and minimize energy consumption of the datacenters which will result in the less number of carbon footprints and thus will help more in achieving Green Cloud. Subsequent tasks must be complete to attain the prime objective:

- To optimize the routine of a variety of techniques has been projected.
- To plan and realize algorithm for Dynamic Voltage Frequency scaling technique by means of the Cloud Simulator.

Cloud Simulator is developed as a lean-to of a packet level network simulator Ns2 [15]. Unlike a small number of presented cloud computing simulators such as Cloud Sim [16] or MDC Sim [17]. Datacenters that host Cloud applications munch through enormous amounts of energy [14], and for this reason result in high operational costs and carbon footprints to the environment. Energy–aware tools [18] are the solution that allows the minimization of the power consumption in data center. The three common energy aware tools are ON/OFF method, migration mechanisms and Dynamic Voltage Frequency Scaling (DVS) Different method that helps in reducing the energy utilization has been projected. Dynamic Voltage Frequency Scaling (DVS) is a method to decrease the power and energy consumption of processors. Decreasing only the operating frequency be able to decrease the power consumption other than energy consumption remains the similar as the computation needs more instant to finish. Abating the supply voltage V can assist in dropping a significant amount of energy because there is quadratic relation between power and V as shown in following Equation:

$$P= C*V2*F \quad (1)$$

Decreasing the supply voltage and operating frequency decreases the power and energy utilization further. When the frequency f is abridged by half, this decreases the processor's power expenditure and still allows task to complete by deadline, and the energy consumption remains the same.

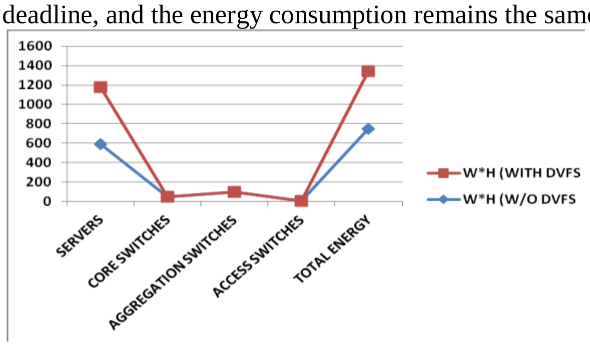

Figure 8: Graphical Representation of Energy with DVS and W/0 DVS

Energy Consumed	*W/O DVS*	*With DVS*
Servers	*590.2*	*590.2*
Core Switches	*51*	*0*
Aggregation Switches	*102*	*0*
Access Switches	*9*	*0*
Total Energy	*752.2*	*590.2*

Table 4: Energy with DVS Compared to Without DVS

5.1 System Architecture

The system architecture consists of job submission, Scheduling Algorithm, VM Manager, Servers and VMs, and the DVS Controller. A user submits the job to VM Manager. VM manager creates the required VMs from relative servers and deploys the job on these VMs assigned by Scheduling Algorithm. Process of Scheduling Algorithm is explained later. DVS controller sets the applicable supplies of voltages and frequencies to servers according to their needs.

5.2 Evaluation without DVS Algorithm

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

which shows that the total wattage consumed without DVS mode is 752.20 watt-hour and the total wattage consumed with DVS mode is 590.20 watt-hour. It shows the variation in energy consumption in DVS mode and Non DVS mode as the number of cloud user varies. And we can see in fig. 5 how DVS technique helps additional in abating the energy consumption. These method achieves up to 28% energy reduction. In fig. 6 we have exposed association of energy consumption amongst two methods at 50 % CPU utilization in data center. MM Policy [19] shows 1.2 KWh energy expenditure and energy consumption with DVS technique is 0.45 KWh.

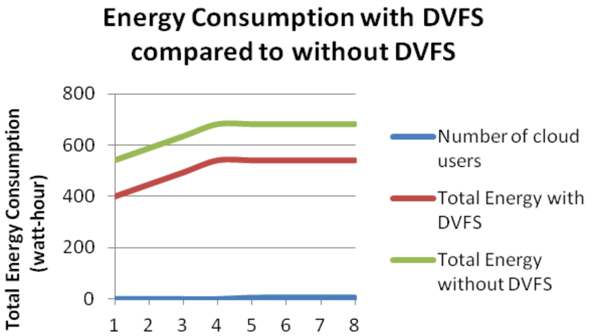

Figure 9: Energy Consumption with No of Cloud Users, DVS Compared to Without DVS

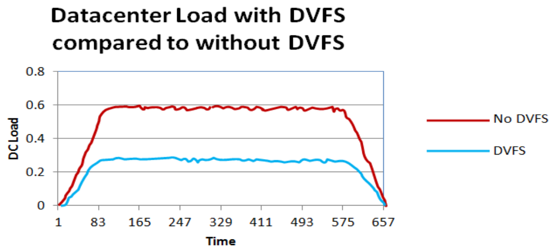

Figure 10: Data center Load with DVS Compared to Without DVS

In arrange to obtain energy efficient in data center, we can also exertion on frameworks that provide an intelligent consolidation methodology using different techniques such as turning on/off machines, machine learning techniques, and power-aware consolidation algorithms to contract with uncertain information while maximizing performance.

V. Conclusion and Future Work

In this paper, we presented a comprehensive deliberation of DVS Approaches of how to implement in various scheme gives a maximum *performance/watt* improvement of Data Centers .We are also analysing the comparative performance of metaphoric approaches on DVS. We have investigated power- aware provisioning the results have shown that data centers can reduce power consumption and increase their profit using DVS schemes. We have shown that the number of algorithms and the energy efficiency of idle cycles do not greatly affect the relative savings of the DVS mechanisms, while the voltage and frequency settings available on the underlying hardware and the task set CPU utilizations profoundly affect the performance of algorithms. Dynamic voltage scaling (DVS) circuits have been widely adopted in many computing systems to provide tradeoffs between performance and power consumption. This paper summarizes the state-of-art research results for energy-efficient algorithms of real-time tasks in DVS systems. In future we design a framework for energy efficiency and performance analysis of DVS-enabled cloud. This framework uses virtual machine request arrival rate, failure rate, repair rate, and service rate of data center servers as model inputs. Based on a queuing network- based analysis, this paper gives analytic solutions of three metrics. The proposed framework can be used to help the design and optimization of energy-aware high performance cloud systems. Dynamic voltage scaling is again an effective way to minimize the temperature by slowing down the speed. How to minimize the energy consumption under temperature constraints and timing constraints will become important in the near future.

Referency

1. C.Saranya, DR.R. Manicka Chezian,"A Study of Energy-Aware DVS Approaches on Virtualization IDC in Cloud Computing", International Journal of Innovations & Advancement in Computer Science (IJIACS),ISSN:2347-8616,Vol 4,Spl Issue, PP. 586-595,Mar 2015.
2. C.Saranya, DR.R. Manicka Chezian, "A Survey on Virtualization Data centers for Green Cloud Computing", International Journal for Trends in Engineering & Technology (IJTET), ISSN:2349-9303,Vol 3,Issue 2, PP. 17-22,feb 2015.
3. T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. D. Micheli, "Dynamic voltage scaling and power management for portable systems," presented at the Proceedings of the 38th annual Design Automation Conference, Las Vegas, Nevada, United States, 2001.
4. Quiroz A, Kim H, Parashar M, Gnanasambandam N, Sharma N. Towards autonomic workload provisioning for enterprise grids and clouds. *Proceedings of the 10th IEEE/ACM International Conference on Grid Computing (Grid 2009)*, Banf, AB, Canada, 13–15 October 2009. IEEE Computer Society: Silver Spring, MD, 2009; 50–57.
5. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

6. and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 2009; **25**(6):599–616.
7. G. Chen, K. Malkowski, M. Kandemir, and P. Raghavan, "Reducing Power with Performance Contraints for Parallel Sparse Applications," presented at The First Workshop on High-Performance, Power-Aware Computing, Denver, Colorado, 2005.
8. Q. Zhu, F. M. David, Y. Zhou, C. F. Devaraj, P. Cao, and Z. Li. Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management. In Proceedings of the 10th International Symposium on High-Performance Computer Architecture, February 2004.
9. R. Xiaojun, *et al.*, "An Energy-Efficient Scheduling Algorithm Using Dynamic Voltage Scaling for Parallel Applications on Clusters," in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, 2007, pp. 735-740.
10. Sekhar Darbha , Dharma P. Agrawal, Optimal Scheduling Algorithm for Distributed-Memory Machines, IEEE Transactions on Parallel and Distributed Systems, v.9 n.1, p.87-95, January 1998
11. T.-S. Tia, J. W.-S. Liu, and M. Shankar "Algorithms and Optimality of Scheduling Aperiodic Requests in Fixed-Priority Preemptive Systems", Technical Report, University of Illinois, 1994.
12. M. Armbrust, et al., "Above the clouds: A Berkeley view of cloud computing", Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.
13. Yann-Rue Lin , Cheng-Tsung Hwang , Allen C.-H. Wu, Scheduling techniques for variable voltage low power designs, ACM Transactions on Design Automation of Electronic Systems (TODAES), v.2 n.2, p.81-97, April 1997.
14. Lars S. Nielsen , Cees Niessen, Low-power operation using self-timed circuits and adaptive scaling of the supply voltage, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, v.2 n.4, p.391-397, Dec. 1994.
15. J. Baliga, R.W.A. Ayre, K. Hinton, R.S. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage and Transport", Proceedings of the IEEE, Vol. 99, Issue 1, pp. 149-167, Jan 2011.
16. Ts`epoMofolo, R Suchithra, "Heuristic Based Resource Allocation Using Virtual Machine Migration: A Cloud Computing Perspective"inInternational Refereed Journal of Engineering and Science (IRJES) ISSN (Online) 2319-183X, (Print) 2319-1821 Volume 2, Issue 5(May 2013), PP.40-45
17. Legrand, L. Marchal, and H. Casanova. Scheduling distributed applications: the SimGrid simulation framework. Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003.
18. Lim S-H, Sharma B, Nam G, Kim EK, Das CR, "MDCSim: A multi-tier data center simulation, platform", In: IEEE International Conference on Cluster Computing and Workshops (CLUSTER), (2009).
19. Iman I Yusuf, Heinz W Schmidt, and Ian D Peake. Architecture-based fault tolerance support for grid applications. In Proceedings of the joint ACM SIGSOFT conference–QoSA and ACM SIGSOFT symposium–ISARCS on Quality of software architectures–QoSA and architecting critical systems–ISARCS, pages 177– 182. ACM, 2011.
20. Wenbing Zhao, PM Melliar-Smith, and Louise E Moser. Fault tolerance middleware for cloud computing. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, pages 67–74. IEEE, 2010.

BIOGRAPHY

C. Saranya Jaikumar (26/07/1987) received her B.Com (Computer Applications) from Vellalar College, Erode, India. She completed her Master of Computer Applications from Erode Sengunthar Engineering College, Erode, India. Presently, she is a Research Scholar at Department of Computer Science, NGM College, Pollachi, India. She presented the Research Papers on International Conferences and Journals. Her area of interest includes Cloud computing, Mobile computing and big data and Hadoop.

Dr. R. Manicka chezian received his M.Sc Applied Science from PSG College of Technology, Coimbatore, India in 1987. He completed his M.S. degree in Software Systems from Birla Institute of Technology and Science, Pilani, Rajasthan, India and Ph.D degree in Computer Science from School of Computer Science and Engineering, Bharathiar University, Coimbatore. He has 25 years of Teaching experience and 17 years of Research Experience. He served as a Faculty of Maths and Computer Applications at P.S.G College of Technology, Coimbatore from 1987 to 1989. Presently, He is working as an Associate Professor of Computer Science in NGM College (Autonomous), Pollachi, India. He has published more than 120 papers in various International Journals and Conferences. He is a recipient of many awards like Desha Mithra Award and Best paper Award. He is a member of various Professional Bodies like Computer Society of India and Indian Science Congress Association. His research focuses on Network Databases, Data Mining, Data Compression, Mobile Computing and Real Time Systems, Network Security, Bio-Informatics and Distributed Computing.