ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

EIRP Based Energy Modelling For Green Cloud Services

T.R.V. Anandharajan, M.A. Bhagyaveni, Suresh, Sujith, Vimal, Vinoth

Department of ECE, CEG, Anna University, Chennai , India.

Department of ECE, CEG, Anna University, Chennai , India.

Department of ECE, CEG, Anna University, Chennai , India.

Department of ECE, CEG, Anna University, Chennai , India.

Department of ECE, CEG, Anna University, Chennai , India.

Department of ECE, CEG, Anna University, Chennai , India.

ABSTRACT— Cloud Computing and the virtualization techniques have helped in developing better SaaS (Software as a Service) platforms. Simulated Software architecture has helped to schedule the VMs (Virtual Machines) in a Cloud provider. The obligation of providing high quality of service to customers leads to the necessity in dealing with the energy and performance trade-off, as aggressive consolidation may lead to performance degradation. This in-turn produces an excess carbon footprint due to the IT sector. The proposed EIRP (Energy/Instruction Rate Performance) based Energy efficient modelling and the subsequent algorithms proved 27% more efficient than the legacy systems. We employed real time Planet Lab workload for our simulations. The proposed algorithms significantly reduced energy consumption, while ensuring a high level of adherence to the SLA (Service Level Agreements).

KEYWORDS— Cloud Computing, Green IT, Distributed Systems

I. Introduction

Cloud Computing has the potential to have a massive impact, positive or negative, on the future carbon footprint of the IT sector. In April 2007, Gartner [16] estimated that the Information and Communication Technologies (ICT) industry generates about 2% of total global CO2 emissions which are equal to the aviation industry. Better consolidation of servers in data centre is the need of the hour. 50% of the servers all over the world are under used which can be localized virtually by the Planet lab work load. There are various levels in this model of the topmost being the user. Who accesses the available hosts by means of VM's.

On the one hand, Cloud Computing data centers are now consuming 0.5% of all the generated electricity in the world, a figure that will continue to grow as Cloud Computing becomes widespread particularly as these systems are "always-on always-available". However, the large data centers required by clouds have the potential to provide the most efficient environments for computation. Computing on this concentration and scale will drive cloud providers to build efficient systems in order reduce the total cost of ownership (TOC) as well as improve their green credentials. There exist significant opportunities for energy conservation via techniques utilizing switching servers off or to low power modes. There are other crucial problems that arise from high power and energy consumption by computing resources. Power is required to feed the cooling system operation. The virtualization technology allows Cloud providers to create multiple Virtual Machine (VMs) instances on a single physical server, thus improving the utilization of resources and increasing the Return on Investment (ROI). The reduction in energy consumption can be achieved by switching idle nodes to low-power modes (i.e. sleep, hibernation), thus eliminating the idle power consumption.

By using live migration the VMs can be dynamically consolidated to the minimal number of physical nodes according to their current resource requirements. Evaluating the performance of Cloud provisioning policies, application workload models, and resources performance models in a repeatable manner under varying system and user configurations and requirements are difficult to achieve. To overcome this challenge, we propose a toolkit that enables modeling and simulation of Cloud computing systems and application provisioning environments. The toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It supports modeling and simulation of Cloud computing environments.

II. RELATED WORK

A systematic literature review was conducted on how our proposed work should address the issues on the Energy-Performance trade-off. Hence we selected papers in the specific areas published between the years 2009 – 2013. Our previous papers we discussed on the various algorithms and methods on Energy efficient strategies [1, 20]. From Table 1 it is evident that the VM consolidation is the best option when the challenge in a datacenter is in an energy perspective. Hence we in this paper concentrate on the energy perspective and how efficient models can be deployed is the goal.

The literature helped in gaining knowledge of the research challenges available. The challenges lie on the component to be considered. The component when considered has to address the Energy-Performance trade-off issue where the Energy depletion is high while the performance of the machine have not been harnessed to an efficient level. The VM consolidation and a better Energy modeling will prove to be a best solution for the aforementioned issues.

Table 1. Literature Review

Author	Virtualization Implementation?	Energy Consumption analysis in Datacenter	Component considered
Kusaka et al. [14]	No	Average	CPU
Mohsen Sharifi et. al [18]	Yes	Good	Virtual Machine
Hua et. al [11]	Yes	Nil	Virtual Machine
Wu et. al [22]	Yes	Average	Virtual Machine
Hsu et. al [10]	Yes	Good	CPU
Kim et. al [12]	Yes	Nil	CPU
Moreno et. al [15]	Yes	Good	CPU
Eugen Feller et. al [8]	Yes	Average	Virtual Machine
Kim et. al [13]	Yes	Average	Virtual Machine
Feller et. al [9]	Yes	Nil	CPU
Weiming Shi et. al [19]	Yes	Nil	CPU
Beloglazov, Anton, and Rajkumar Buyya [3]	Yes	Good	RAM and VM

III. ENERGY PERFORMANCE TRADE-OFF

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. Existing Methods

The The existing methods from the literature point of view have to be addressed specifically on the VM consolidation because the work play a substantial role in this domain. The algorithms and methods for the evaluation are taken from [3]

Median of Absolute Deviation:

The idea is to produce estimators that are not exorbitantly influenced by small departures from model surmises. The Median of Absolute Deviation (MAD) is a measure of statistical dispersion. It is a more robust estimator of scale than the sample variance or standard deviation, as it carries on better with disseminations without a mean or change, for example the Cauchy distribution. The MAD is a robust statistic, being more flexible to outliers in a data set than the standard deviation. In standard deviation, the distances from the mean are squared leading to large deviations being on normal weighted all the more vigorous. This implies that outliers might altogether impact the quality of standard deviation. In the MAD, the extent of the separations of a minor number of outliers is superfluous. For a uni-variate data set, the MAD is outlined as the average of irrefutably the deviations from the average of the information set, that is, the MAD is the average of unquestionably the qualities of deviations (residuals) from the information's average. In the proposed oversubscription identification calculation, the CPU usage limit is demarcated. Parameterical evaluation has been carried out, the parameters permits the conformity of the security of the system: an easier quality of effects in a higher tolerance to variety in the CPU usage, while potentially expanding the level of SLA violations brought about by the consolidation.

Interquartile Range:

This section proposes a method for setting an adaptive CPU utilization threshold based on another robust statistic. In descriptive statistics, the Interquartile Range (IQR) [2], also called the midspread or middle fifty, is a measure of statistical dispersion. It is equal to the difference between the third and first quartiles. Unlike the (total) range, the interquartile range is a robust statistic, having a breakdown point of 25%, and thus, is often preferred to the total range. For a symmetric distribution (i.e., such that the median equals the average of the first and third quartiles), half of the IQR equals the MAD.

Local Regression

The next heuristic is based on the Loess method (from the German loss – short for local regression LR) proposed by Cleveland [4]. The main idea of the local regression method is fitting simple models to localized subsets of data to build up a curve that approximates the original data. The observations (x_i, y_i) are assigned neighbourhood weights using the tricube weight function the weight function gives the most weight to the data points nearest the point of estimation and the least weight to the data points that are furthest away. The use of the weights is based on the idea that points near each other in the explanatory variable space are more likely to be related to each other in a simple way than points that are further apart. Following this logic, points that are likely to follow the local model best influence the local model parameter estimates the most. Points that are less likely to actually conform to the local model have less influence on the local model parameter estimates. The traditional weight function used for LOESS is the tri-cube weight function, $(x) = (1 - |x|^3)^3 \; I \; [|x| < 1]$. However, any other weight function that satisfies the properties listed in Cleveland [5] could also be used. The weight for a specific point in any localized subset of data is obtained by evaluating the weight function at the distance between that point and the point of estimation, after scaling the distance so that the maximum absolute distance over all of the points in the subset of data is exactly one.

The distance from x to x_i, and let $D_{(i)}(x)$ be these distances in order from most modest to biggest. At that point, the neighbourhood weight for the perception (x_i, y_i) is described by the weight $w_i(x)$. For x_i such that $D_i(x) < D_q(x)$, where q is the amount of perceptions in the subset of information limited around x. The measure of the subset is described by a parameter of the strategy called the transfer speed. For instance, if the level of the polynomial fitted by the strategy is 1, the parametric group of capacities is y = a + bx. The line is fitted to the information using the weighted least squares strategy with weight $w_i(x)$ at (x_i, y_i). The qualities of a and b are considered by minimizing the capacity. In the proposed calculation, this approach is connected to fit a pattern polynomial to the final k observation of the CPU utilization, where $k = [q/2]$. A polynomial is fit for a single point, the last observation of the CPU utilization (i.e., the right limit x_k of the data set). The problem of the boundary region is well-known as leading to a high bias. According to Cleveland, fitted polynomials of degree 1 typically distort peaks in the interior of the configuration of observations, whereas polynomials of degree 2 remove the distortion but result in higher biases at boundaries. Hence, for host overload identification, polynomials of degree 1 are decided to lessen the bias at the boundary. Let x_k be the final perception, and x_1 be the kth perception from the right boundary. In the proposed Local Regression (LR) algorithm, using the described method derived from Loess, a new trend line is found for each new observation [2]. This trend line is used to estimate the next observation. If the inequalities are satisfied, the algorithm detects a host overload, requiring some VMs to be offloaded from the host and any of the VMs allocated to the host.

Robust Local Regression:

The Robust Local Regression (LRR) version of Loess is vulnerable to outliers that can be caused by heavy-tailed distributions. To make Loess robust, Cleveland [4] proposed the addition of the robust estimation method bi-square to the least-squares method for fitting a parametric family. This modification transforms Loess into an iterative method. The initial fit is carried out with weights defined using the tri-cube weight function. Using the estimated trend line, it is applied to estimate the next observation. If the referred inequalities are satisfied, the host is detected to be overloaded. This host algorithm is denoted Local Regression Robust (LRR).

VM Selection:

Once a host overload is detected, the next step is to select VMs to offload from the host to avoid performance degradation. This section presents three policies for VM selection.

The Minimum Migration Time Policy:

The Minimum Migration Time (MMT) policy migrates a VM v that requires the minimum time to complete a migration relatively to the other VMs allocated to the host. The migration time is estimated as the amount of RAM utilized by the VM divided by the spare network bandwidth available for the host j. Let Vj be a set of VMs currently allocated to the host j. The MMT policy finds a VM v that satisfies the conditions formalized in.

The Random Selection Policy:

The Selection Choice (RS) policy randomly selects a VM to be migrated from the host according to a uniformly distributed discrete random variable, whose values index a set of VMs V_j allocated to the host j.

The Maximum Correlation Policy:

The Maximum Correlation (MC) policy is based on the idea proposed by Verma et al [21]. The idea is that the higher the correlation between the resource usage by applications running on an oversubscribed server, the higher the probability of the server oversubscription. According to this idea, those VMs are selected to be migrated that have the highest correlation of the CPU utilization with the other VMs. To estimate the correlation between the CPU utilization of VMs, the multiple correlation co-efficient is applied. It is used in multiple regression analysis to assess the quality of the prediction of the dependent variable. The multiple correlation coefficient corresponds to the squared correlation between the predicted and the actual values of the dependent variable. It can also be interpreted as the proportion of the variance of the dependent variable explained by the independent variables. A set of random variables represent the CPU utilization of VMs allocated to a host. The objective is to evaluate the strength of the correlation between remaining random variables. An augmented matrix containing the observed values of the independent random variables are taken into account and this gives us vector of observations for the dependent variable. The matrix is called augmented because the first column is composed only of one. A vector of predicted values of the dependent random variable is obtained. Having found a vector of predicted values, it is now possible to compute the multiple correlation co-efficient, which is equal to the squared coefficient of correlation between the observed values of the dependent variable and the predicted values.

B. Energy Curve Model

From our previous paper we have extended the Energy Curve Model [23]. Since VM consolidation is done to minimize the number of active physical hosts, the individual intermigration time interval has to be maximized since in a time of n frames the mean number of hosts that are alive is inversely proportional to the efficiency of VM consolidation. Consolidation of VM through SLA and VM migration is done by the Energy curve. The efficiency of VM consolidation is conceptualized by maximizing the time intervals between Virtual Machine Migration from overloaded host servers. Although VMs experience variable workloads, the maximum CPU capacity that can be allocated to a VM should be less than the overall maximum CPU capacity.

Hence, we limit the problem formulation to a single VM migration as in Figure 1, i.e., the time span of a problem instance is from the end of a previous VM migration where T_{VM} is the time when a VM migration starts; Th_{CPU} is the CPU utilization threshold defining the host oversubscription; $T_{ol}(T_{VM}, Th_{CPU})$ is the time during which the host has been overloaded, which is a function of T_{VM} and Th_{CPU}; T_{tot} is the total time, during which the host has been active, on and to the end of the next VM migration.

At some point in time T_{slav}, an SLA violation occurs and continues until time T_{slavi} SLAV interval. An SLAV in our terms based on the QoS metrics where both throughput and response time delivery is taken into account by the organizing Cloud.

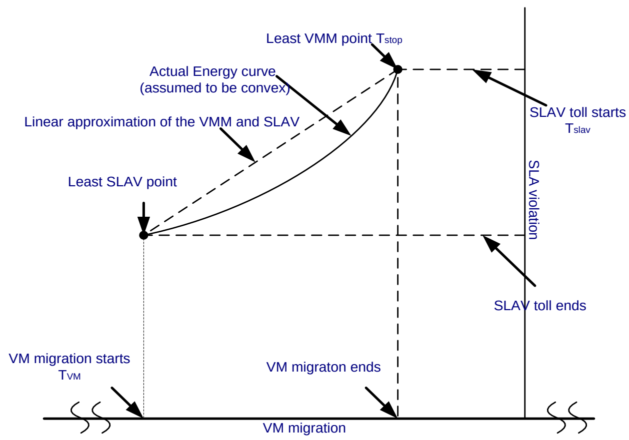

Figure 1: The Objective function in terms of SLAV and VM migration

$SLATAH = \frac{1}{m}\sum_{i=1}^{m} \frac{T_{100\%}}{Alive_{tot}} * PDM = \frac{1}{n}\sum_{i=1}^{n} \frac{\Theta_{pd\,(n)}}{CPU\,(n)_{tot}} = SLAV$ The hosts which are alive and consume 100% utilization and the performance degradation due to migration the product of SLATAH and PDM gives our SLAV [6].

$$SLATAH = \frac{1}{m}\sum_{i=1}^{m} \frac{T_{100\%}}{Alive_{tot}} \qquad (1)$$

$$PDM = \frac{1}{n}\sum_{i=1}^{n} \frac{\Theta_{pd\,(n)}}{CPU\,(n)_{tot}} \qquad (2)$$

$$SLAV = SLATAH * PDM \qquad (3)$$

If m is the number of hosts, $T_{100\%}$ is the total time during which the host i has experiences the utilization of 100% leading to an SLA violation. $Alive_{tot}$ is the total of the host i being in the VM feeder state, n is the number of VMs; $\Theta_{pd\,(n)}$ the estimate of the performance degradation of the VM n caused by migrations; $CPU(n)_{tot}$ is the total CPU capacity requested by the VM n during its lifetime [3]. In other words, due to the over-subscription and variability of the workload experienced by VMs, at the time T_{slav} the overall demand for the CPU performance exceeds the available CPU capacity and does not decrease until T_{slavi}. It is assumed that according to the problem definition, a single VM can be migrated out of the host. This migration leads to a decrease of the demand for the CPU performance and makes it lower than the CPU capacity [7]. We define T_{stop} to be the stopping time, which is equal to the latest of either the end of the VM migration or the beginning of the SLA violation. A VM migration takes time t. The problem is to determine the target time, T_{λ} when a VM migration should be initiated to minimize the total cost consisting of the energy cost and the cost caused by an SLA violation if it takes place(During a migration an extra host is used to accommodate the VM being migrated, and therefore, the total energy consumed during a VM migration is twice the cost incurred for one host.) Let T_r be the remaining time since the beginning of the SLA violation, i.e. $T_r = T_{stop} - T_{slav}$. As time increases for resource scarcity, the energy cost increases and this in turn increases the power consumed by the battery where the energy cost includes the cost for the normal operation and deadline time [17].

As the availability of host becomes lesser, the time of completion increases for SLA violation and this takes VM migration a longer time. Reduction in the completion time of SLAV improves customer QoS. Lesser resource also leaves a larger carbon footprint since consumption of Energy increases. Inefficient resources lead to more energy cost which is the sum of power consumed by the overloaded host and the new host, to be alive until VM migration continues. To maintain appropriate QoS further reduction in time is not allowed. To attain minimum energy performance CQRMPP algorithm has been for the available resource. The time for the VM provisioning involves the time after SLA violation and the start of VM migration. Hence energy (or) power consumed by the above mentioned time is also taken into account. To analyze the energy curve we define the cost model as shown in Figure 1. The cost associating with times are the SLAV and VM migration. Finding an efficient resource from the available resource is our goal for scheduling. QoS and reduction in performance degradation due to migration in a strategy where VMs are dynamic and volatile the scheduling has to be perfect.

C. Minimum Processing Power Policy

The Minimum Processing Power (MPP) policy migrates a VM from $\mathcal{v} = \{\mathcal{v}_1, \dots, \mathcal{v}_n\}$ that requires the minimum processing power to complete a migration relatively to the other VMs allocated to the available host based on our proposed MEP algorithm to the available host. The minimum processing power is estimated as the amount of power utilized by the VM divided by the MIPS available for the host j as given in equation 3. Let V_k be a set of VMs currently allocated to the host. The energy curve is usually non-linear. Total energy cost can be leveraged by harnessing the least costs of SLAV and VM migration. We introduce six steps for achieving this

- Select new host identification for VM migration to reside
- Least Energy curve which can be achieved by

$$Energy\ slope = \frac{point\ \ SLAV\ toll\ end - point\ \ SLAV\ toll\ start}{end\ of\ VM\ migration\ \ time - start\ of\ VM\ migration\ \ time} \quad (4)$$

- Association of VM is based on the ascending order of the Energy slope.
- The Sorted VM with minimum energy slope will be assigned to minimum performance hosts.
- Minimum performance hosts are identified using equation 2 & 3.
- The MPP is processed in parallel for all the VM in the hosts and critical parallel parts are identified and hence allocation of critical VM to critical host is realized in parallel in our simulation.

Figure 1 shows the identified slope of parallel critical VM's. The energy curve helps to estimate the efficient use of the available resource, SLAV and VM migration time is calculated. The minimum energy curve helps QoS which can be achieved by reduction in VM migration time and this deviates the SLAV time. Consolidation of VM due to reduced VM migration increases SLA violation and Energy consumed gets higher for the available resource. The energy consumed by the processor in a data center is measured by the energy per instruction which is given by the ratio between the power and the performance. We define the EIRP (Energy per Instruction Rate Performance) for a host having VMs as

$$Energy_{Host} = \frac{Power_{\ h}}{MIPS_h} \quad (5)$$

The Maximum Requested Millions of Instructions per Second (MIPS) policy migrates a VM *v* that requests the maximum MIPS in an overloaded host relatively to the other VMs allocated to the host. It is based on the policy that if a host is getting overloaded it is because of a new allotted cloudlet (workload) or increase in requested utilization of a workload. Hence after a Host is found to be overloaded the VM making it overloaded will be found by analyzing the VM requesting the maximum of MIPS of the processing element in a host. Let V_j be a set of VMs currently allocated to the host *j*. The Maximum MIPS policy finds a VM *v* that satisfies conditions formalized in

$$v \in V_j \mid \forall a \in V_j, MIPS_u(v) \geq MIPS_u(a) \quad (6)$$

where $MIPS_u(a)$ is the Millions of Instructions per Second currently requested by the VM *a*.

Algorithm 1: VM placement Optimization

```
Input: hostList Output: migrationMap
foreach host in hostList do
    if isHostOverloaded (host) then
        vmsToMigrate.add(getVmsToMigrateFromOverloadedHost(host)
    migrationMap.add(getNewVmPlacement(vmsToMigrate))
    vmsToMigrate.clear()
return migrationMap
```

The general algorithm of VM placement optimization is shown in Algorithm 1. The complexity of the algorithm is n*m where n is the number of VMs and m is the number of hosts to be optimized. First, the algorithm looks through the list of hosts and by applying the overloading detection algorithm checks whether a host is overloaded. If the host is overloaded, the algorithm applies the VM selection policy to select VMs that need to be migrated from the host. Once the list of VMs to be migrated from the overloaded hosts is built, the VM placement algorithm is invoked to find a new placement for the VMs to be migrated.

The host overloading is detected using Local Regression smoothing of curves proposed by Anton Beloglazov and Rajkumar Buyya. At each point in the data set a low-degree polynomial is fitted to a subset of the data, with explanatory variable values near the point whose response is being estimated. The polynomial is fitted using weighted least squares, giving more weight to points near the point whose response is being estimated and less weight to points further away. The value of the regression function for the point is then obtained by evaluating the local polynomial using the explanatory variable values for that data point. The LOESS fit is complete after regression function values have been computed for each of the n data points.

The weight function gives the most weight to the data points nearest the point of estimation and the least weight to the data points that are furthest away. The use of the weights is based on the idea that points near each other in the explanatory variable space are more likely to be related to each other in a simple way than points that are further apart. Following this logic, points that are likely to follow the local model best influence the local model parameter estimates the most. Points that are less likely to actually conform to the local model have less influence on the local model parameter estimates.

The traditional weight function used for LOESS is the tri-cube weight function,

$$T(u) = \begin{cases} (1-|u|^3)^3, & if\ |u| < 1 \\ 0, & otherwise \end{cases} \quad (7)$$

In our algorithm (LR), using the described method derived from Loess, for each new observation we find a new trend line $\hat{g}(x) = \hat{a} + \hat{b}x$. This trend line is used to estimate the next observation $\hat{g}(x_{k+1})$. The algorithm decides that the host is considered overloaded and some VMs should be migrated from it if the inequalities are satisfied.

$$s.\hat{g}(x_{k+1}) \geq 1, \qquad x_{k+1} - x_k \leq t_m$$

where $s \in R+$ is the safety parameter; and t_m is the maximum time required for a migration of any of the VMs allocated to the host.

IV. Results

In this experiment, we compare the performance of energy-conscious resource management techniques against a benchmark trivial technique, which did not consider energy-optimization during provisioning of VMs to hosts. In the benchmark technique, the processors were allowed to throttle at maximum frequency (i.e. consume maximum electrical power) whereas in this case, they operated at the highest possible processing capacity (100%).The energy-conscious technique was an extension of the trivial policy; it applied live migration of VMs every 300 seconds for adapting to the allocation. The basic idea here was to consolidate VMs on a minimal number of nodes and turn off idle ones in order to minimize power consumption. For mapping VMs to hosts, a greedy algorithm was applied that sorted VMs in decreasing order of their CPU utilization and allocated them to hosts in a first-fit manner. VMs were migrated to another host, if that optimized energy consumption. To avoid SLA violations, the VMs were packed on the hosts in such a way that the host utilization was kept below a pre-defined utilization threshold. This threshold value was varied over a distribution during the simulation for investigating its effect on the behavior of the system. The simulation was repeated 10 times; the mean values of the results that we obtained are presented below.

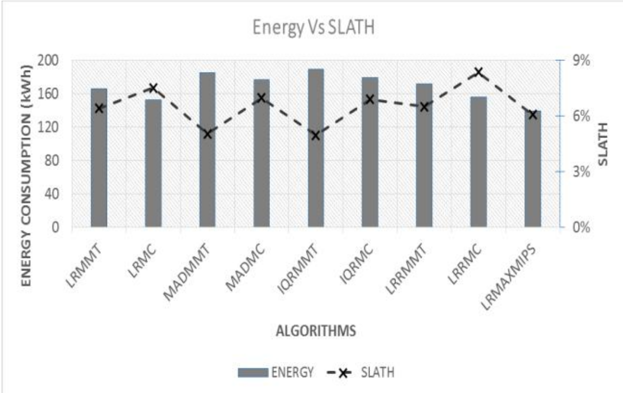

Figure 2. Effect of Energy and SLATAH in 24 hour simulation

The simulation is done with workload data recorded on 03/03/2011. The Energy consumed in each existing algorithm and proposed algorithm is tabulated and plotted. The Energy consumed by the existing algorithms LRMMT, LRMC, MADMMT, MADMC, IQRMMT, IQRMC, LRRMMT and LRRMC is 165.99 kWh, 152.11 kWh, 184.88 kWh, 176.24 kWh, 188.86 kWh, 178.8 kWh, 171.06 kWh and 155.68 kWh respectively for 800 hosts and 1052 VMs for a day. The simulated results with proposed power aware provisioning of VM selection is noted as 138.58 kWh as shown in Figure 2. The obtained result shows efficient reduction in the power consumed is reduced by 9% from the least power consumed in the existing algorithm. On analyzing the Energy consumption of the Host more percentage of energy consumption in the consolidated requested performance is more when there more number of VMs. But it is noted that the energy consumed is less than the total provisioned utilized performance. This is because of the better dynamic VM consolidation.

The SLA is a parameter which should be monitored for cost effective data center. We have to come up with an algorithm that balances both Energy consumption and SLA violation metrics. Since the VM that overloads the host is picked up for VM migration the host is no longer overloaded. But in the existing algorithms the VM which minimizes migration time is migrated that doesn't assure overcoming of the host overload. Thus as expected the SLA violation is minimum at all case in the proposed algorithm. The Figure 3 shows that the SLATH is greatly minimized in the proposed algorithm and both essential parameters are balanced to arrive at an effective ESV metric.

It is been observed that the overall SLA for the data center is also been reduced provided that the Energy consumed is also maintained as low as possible. This is noted from the Figure 2. The proposed algorithm LRMAXMIPS has an overall SLA violation of 0.170%. The Median Absolute Deviation and Interquartile range host overloading algorithm proves to have lesser overall SLA but its Energy Consumption reveals it be less efficient algorithm than the proposed since LRMAXMIPS balances between both metrics.

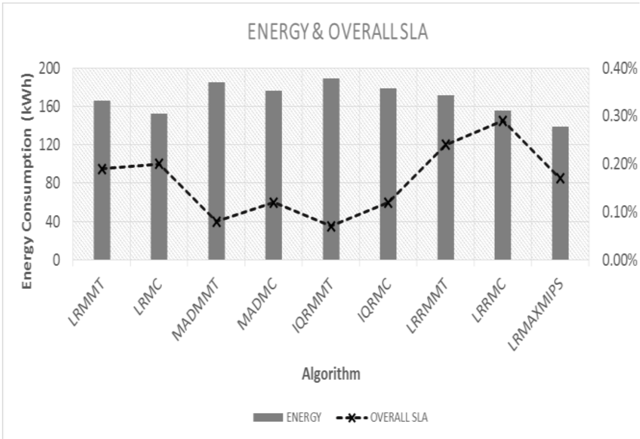

Figure 3. Effect of Energy and Overall SLA violation in 24 hour simulation

In existing algorithm of LRMMT the VM which takes minimum time to migrate will be migrated regardless of the utilization it imposes on the Host. Hence even after migrating a VM from an overloaded host there may be chances of existing a VM that again overloads the Host. Hence the VM consolidation must be done such that a single VM migration might migrate the right VM that is overloading the host. In the proposed algorithm the VM which is maximizing the overloaded Host utilization is figured out and migrated to the best fitting host. Thus the proposed algorithm maintains the performance degradation due to migration as expected PDM and comparable to the other efficient algorithm. The PDM variation with different algorithm is visualized from the Figure 4.

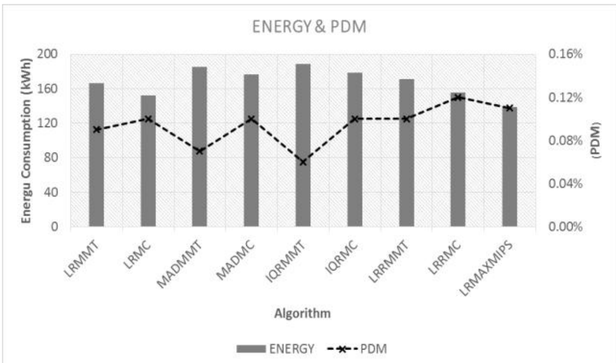

Figure 4. Effect of Energy and PDM in 24 hour simulation

The results showed that energy-conscious techniques can significantly reduce the total power consumption of data center hosts (up to 50%) as against the benchmark technique. However, these are only the indicator results; the actual performance of energy-conscious techniques directly depends on the type of application service being hosted in the cloud. There is much scope in this area for developing application-specific energy optimization techniques. With the growth of the utilization threshold, the energy consumption decreases because VMs can be consolidated more aggressively. This also leads to a decrease in the number of VM migrations and an increase in the number of SLA violations. Better trade-off has to be made between Energy and SLA and more efficient algorithms should be provided for the power aware data center.

V. CONCLUSION

On analyzing the results of simulating the data center with the proposed VM selection policy algorithm it has been noted that the energy consumed by the datacenter is reduced up to 27% from the legacy methods and algorithms. Hence by efficiently consolidating the VM with live migration among the active host in the data centers energy consumed can be reduced aggressively. At the same time Service Level Agreement maintained with the consumers should be kept low. The proposed algorithm proved to be an efficient algorithm that balances with both metrics Energy and SLA.

ACKNOWLEDGMENT

This work was supported by the University Grants Commission (UGC), New Delhi, India. RGNF Scheme-No. F.14-2 (SC) /2009 (SA-III) -April 2009, grant dated Nov 2010.

REFERENCES

[1] Anandharajan, T.R.V. and Bhagyaveni, M.A. 2011. Co-operative Scheduled Energy Aware Load-Balancing technique for an *Efficient Computational Cloud. International Journal of Computer Science Issues. (2011).*

[2] Beloglazov, A. et al. 2011. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in Computers. Academic Press. (2011).*

[3] Beloglazov, A. and Buyya, R. 2011. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurrency and Computation: Practice and Experience. Wiley Online Library. (2011).

[4] Cleveland, W.S. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association. Taylor & Francis Group. (1979).*

[5] Cleveland, W.S. and Loader, C. 1996. Smoothing by local regression: Principles and methods. *Statistical theory and computational aspects of smoothing. Springer. 10–49. (1996).*

[6] Deboosere, L. et al. 2012. Efficient resource management for virtual desktop cloud computing. *The Journal of Supercomputing. (2012).*

[7] Dobber, M. et al. 2009. Dynamic Load Balancing and Job Replication in a Global-Scale Grid Environment: A Comparison. *IEEE Transactions on Parallel and Distributed Systems. (2009).*

[8] Feller, E. et al. 2012. Energy Management in IaaS Clouds: A Holistic Approach. *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. 204–212. (2012).

[9] Feller, E. and Morin, C. 2012. Autonomous and Energy-Aware Management of Large-Scale Cloud Infrastructures. *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*. 2542–2545. (2012).

[10] Hsu, C.-H. et al. 2011. Energy-aware task consolidation technique for cloud computing. *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. 115–121. (2011).

[11] Hua, Z.Y. et al. 2010. Discussion of Intelligent Cloud Computing System. *Web Information Systems and Mining (WISM), 2010 International Conference on*. 319–322. (2010).

[12] Kim, J.H. et al. 2011. Performability analysis of iaas cloud. *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*. 36–43. (2011).

[13] Kim, K.H. et al. 2010. Power-aware Provisioning of Virtual Machines for Real-time Cloud Services. (2010).
[14] Kusaka, T. et al. 2011. Queuing theoretic approach to server allocation problem in time-delay cloud computing systems. *Teletraffic Congress (ITC), 2011 23rd International. 310–311. (2011).*
[15] Moreno, I.S. and Xu, J. 2011. Customer-aware resource overallocation to improve energy efficiency in realtime Cloud Computing data centers. *Service-Oriented Computing and Applications (SOCA), 2011 IEEE International Conference on*. 1–8. (2011).
[16] Pettey, C. 2007. Gartner estimates ICT industry accounts for 2 percent of global CO2 emissions. (2007).
[17] Protocols, H.R. et al. Power-Aware On-Demand Routing Protocols for Mobile Ad Hoc Networks. *Source*. (2012).
[18] Sharifi, M. et al. 2012. Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques. *The Journal of Supercomputing*. Springer.
[19] Shi, W. and Hong, B. 2011. Towards profitable virtual machine placement in the data center. *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*. 138–145.
[20] T.R.V. Anandharajan, Deepak Bhargavan, M.A. Bhagyaveni 2013. VM CONSOLIDATION TECHNIQUES IN CLOUD DATACENTER. *Journal of Theoretical and Applied Information Technology*. (2013).
[21] Verma, A. et al. 2009. Server workload analysis for power minimization using consolidation. *Proceedings of the 2009 conference on USENIX Annual technical conference*. 28–28.
[22] Wu, D. et al. 2012. On P2P mechanisms for VM image distribution in cloud data centers: Modeling, analysis and improvement. *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*. 50–57.
[23] TRV Anandharajan and MA Bhagyaveni. "Trust Based VM Consolidation in Cloud Data Centers." In *Recent Trends in Computer Networks and Distributed Systems Security*, pp. 103-114. Springer Berlin Heidelberg, 2014.