Journal of Theoretical and Applied Information Technology
20th July 2013. Vol. 53 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

VM CONSOLIDATION TECHNIQUES IN CLOUD DATACENTER

[1]T.R.V.ANANDHARAJAN, [2]DEEPAK BHARGAVAN, [3]M.A. BHAGYAVENI
[1,2,3]Anna University, Department of ECE, College of Engineering, Guindy, Chennai, INDIA.
E-mail: [1]anandharajan.t.r.v@gmail.com, [2]deepak.bhargavan@gmail.com, [3]bhagya@annauniv.edu

ABSTRACT

The crunch in resources and ability to serve the interest of customers with maximum efficiency and minimum resources has led to a wide range of research in cloud resource management. Various cloud service providers currently in the market are testing several algorithms to manage their resources and has not been able to strike a balance between service quality and energy management. Through our paper we intend to strike a deal between energy and quality by maintaining both within preferred limits. We proposed an Efficient Energy Aware (EEA) analysis by introducing RHO (Ram based Host Overloading detection) algorithm. It has shown significant improvements over existing power aware host overloading detection algorithms like IQR (inter quartile range). This algorithm gave 37.26% improvement in terms of energy and 70% improvement in PDM (Performance degradation due to migration) over IQR.

Keywords: *Distributed System, Cloud Computing, Datacenter, Green IT*

1. INTRODUCTION

Amazon came up with a concept of network of workstations which was then a research topic at Berkley (1994-1999). This concept of cluster of computers protected the system against single point failure. Cloud computing gave way to minimum infrastructure establishments that could serve with high efficiency. Every cloud service provider has a large number of hosts or servers which are geographically stationed at various locations according to its customer bases and economic viability. The Proposed algorithms are aimed at managing these servers in an energy efficient manner without much degradation in QOS (quality of service).

As stated by Jianzhe Tai et al [1] ARA (adaptive resource allocation) algorithms were deployed for data center management. But these algorithms were non power aware as it did not take into account the current energy utilization of hosts. It led to poor performance in terms of energy management. Barnaby Malet et al [2] has tried to resolve issues of cloud resource management over multiple cloud but has not taken into account the utilization limits for data centers or its periodic updating.

In contrast to the discussed studies we propose an algorithm that is capable of handling a network of data centers with minimum failure and energy. We focus on energy consumption of data centers and predict its future performance by adaptive statistical analysis. We manage the load on data centers by efficient monitoring and periodic updating of its performance. Proposed algorithm has deployed efficient statistical analyses free from biases by outliers or extreme quantities.

2. RELATED WORK

The earlier works have considered some important parameters but have not considered some important aspects too. The tabular column listed below shows those aspects and we have taken sincere efforts to address those issues.

Table 1: Existing work in Datacenter

Author	*Specific Component of Consideration*	*Energy Consumption analysis in Datacenter*
Imada et. al [3]	Virtual Machine	Yes
Li et. al [4]	CPU	No
Yiuo Mei et, al [5]	CPU	Yes
Marzolla et. al [6]	Virtual Machine	Yes
Moghaddam et. al [7]	Virtual Machine	Yes
Yongqiang Gao et. al [8]	CPU	Yes
H. Viswanathan et. al [9]	Virtual Machine	No
Wang et. al [10]	RAM	No
Weiming Shi et. al [11]	CPU	No

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Eugen Felleret. al [12]	CPU	Yes
Liting Hu et. al [13]	CPU	No
Eugen Feller [14]	CPU	No

In this paper we concentrate on the Datacenter work. Datacenter work involving virtualizaed machines are considered and our proposed algorithms have been implemented. We have made a significant contribution than our work in our previous paper [15]

3. DATA CENTER UTILIZATION CALCULATION

We have our data centers broker and cloudlets created followed by identification of utilization values of all hosts. If any of these utilization values exceed our statistically determined value of utilization we declare that host to be overloaded. The overloaded hosts assign an array namely H_{oj}.

3.1 efficient Energy Aware Ram Consolidated Vm Selection Policy (Eearvs)

Figures We in this paper propose an Efficient Energy aware algorithm by introducing RAM consolidation for VM selection policies. EEARVS has proved to be highly effective in combination with RAM consolidated host overloading detection algorithms as they are capable of reducing energy requirement significantly. We have our data centers broker and cloudlets created followed by identification of utilization values of all VMs with in a host under consideration. VM with minimum RAM utilization will be considered for migration. Migratable VMs are saved in a linear array named migratable VM list V_m. EEARVS when applied in combination with IQR(Inter Quartile Range), MAD(Median Of Absolute Deviation), and LR(Local Regression) gave competitive results than established VM selection algorithms like MC(Maximum Correlation), MMT(Minimum Migration Time) and MU(Minimum Utilization).
We have implemented minimum utilization technique to establish the migratable VMs list. The migratable VMs list is indicated by the variable V_m.
If we have 'k' number of overloaded hosts, the VM migration policy is applied to VMs within these 'k' hosts. Let each of these hosts have 'n' number of VMs. Then we can approximate the utilization values of these VMs for a single host-1 as a matrix V_1 shown below:

$$V_{ij} = \begin{bmatrix} V_{11} & \cdots & V_{1n} \\ \vdots & \ddots & \vdots \\ V_{n1} & \cdots & V_{nn} \end{bmatrix} \quad (1)$$

Where the column matrix represented by $[V_{11}, V_{21} \ldots\ldots\ldots V_{n1}]$ refers to utilization values of 'n' number of VMs in host-1 at a particular time instant say 'T_1'. From these values $[V_{11}, V_{21} \ldots\ldots\ldots V_{n1}]$ We select the one with minimum utilization for migration. This can be proved with the help of theorem of minimum achievable utilization for fault tolerant processing of periodic tasks. The theorem states that for any task set utilization factor is given by 'U' which can be represented by an equation :

$$U = \frac{C_1}{T_1} + \frac{C_2}{T_2} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + \frac{C_n}{T_n} \quad (2)$$

Where the factors C_i and T_i represent the execution time of the task and period of the task respectively. The theorem declares a task set 'S' consisting of 'n' tasks. The task set 'S' can be defined by S= $\{C_i, T_i, R_i\} \mid$ i=1,2,……n, R_i represent the release time. The same utilization concept has been implemented into the cloud concept by calculating utilization of a VM by adding the RAM requirement of all cloudlets running in a VM divided by the total RAM capacity of the VM.

$$V_{11} = \frac{\sum_{i=1}^{n} C_i}{R_v} \quad (3)$$

Here C_i represents the RAM utilized by each cloudlet if there are a total of 'n' Cloudlets running in this VM under consideration. 'R_v' represents the total RAM allocated to the VM under consideration. V_{11} represents the utilization of VM-1 at time instant T_1. There by we obtain the utilization values of all the VMs inside the host under consideration, and select VMs with minimum utilization values for migration by mapping them into a migratable VMs list V_m.Pseudo code for EEARVS:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm RVS:
Input :VM list Output: Migratable VM list
For each host all the VMs are loaded in to an array
For each host in overutilized hostlist do
Is hostoverutilized then
Migratable VMs list.add(minumimum utilization VM)
Migration map.list(update V_m)
If host in underloaded hostlist do
Migratable VMs list.add(VM list)
Migration map.list(update V_m)
Return migration map V_m

3.2 efficient Energy Aware Ram Regression Technique (Eearrt)

While working towards developing an algorithm that will efficiently manage the hosts in cloud environment with minimum energy and minimum SLAV we developed a coding for host overutilization detection by implementing RAM consolidation with Local Regression Technique and evaluated its performance in combination with VM selection algorithms MMT, MC, MU. The cloudlet scheduler has been redesigned to retrieve the RAM information regarding each application. The RAM requirement of each VM is sampled and summed up for each host from VM scheduler. Current requested ram and host allocated ram info are collected for calculating utilization history.

$$y_0 = a_0 + a_1 H_{1n} + a_2 H_{1(n-1)} + a_3 H_{1(n-2)} + \ldots\ldots\ldots\ldots\ldots$$
$$y_1 = a_0 + a_1 H_{2n} + a_2 H_{2(n-1)} + a_3 H_{2(n-2)} + \ldots\ldots\ldots\ldots\ldots \quad (4)$$

From the above equations representing linear egression equations we approximate it in to a form given below involving up to nth value or latest value of utilization so that we can linearise the above equation.

$$y_0 - a_0 - a_1 H_{1n} = 0$$
$$y_1 - a_0 - a_1 H_{2n} = 0 \quad (5)$$

The utilization value of the hosts are represented by an n*n matrix given below:

$$H_{ij} = \begin{bmatrix} H_{11} & \cdots & H_{1n} \\ \vdots & \ddots & \vdots \\ H_{n1} & \cdots & H_{nn} \end{bmatrix} \quad (6)$$

Each row of the utilization history represents the utilization of each host in order of latest to oldest with respect to time.

$$H_1 = [H_{11}, H_{12}, \ldots\ldots\ldots\ldots H_{1n}] \quad (7)$$

These rows representing utilization of each host is reversed in order to set the first elment as the value containing latest value of utilization corresponding to the host under consideration. $H_{1rev} = [H_{1n}, H_{1(n-1)}, \ldots\ldots\ldots\ldots H_{11}]$. From the host utilization array by applying least square technique we obtain the point corresponding to utilization of the host at any particular instant represented by the equation given below:

$$\mathrm{P(x, y)} = \sum_{\mathrm{i=1}}^{\mathrm{n}} W(x) * (y - a_0 - a_1 H_{in}) \quad (8)$$

To P(x,y) we apply linearization technique to obtain the slope and intercept. On obtaining the slope ‘s’ and intercept ‘i’ we can predict the value of utilization of host at any particular time instant say ‘k’ as shown below:

$$Z(u_k) = s u_k + i \quad (9)$$

From the above equation we can predict the value of utilization at ti me instant ‘k+1’ as shown below:

$$Z(u_{k+1}) = s u_{k+1} + i \quad (10)$$

Hence we could predict the future utilization values of hosts and suitably use it towards migrating VMs from hosts having probability of getting overloaded in near future.

3.3 Efficient Energy Aware Ram Consolidated Host Overloading Detection (Eearho)

We in this paper propose EEARHO based on RAM consolidation for an efficient energy analysis in a data center. The value of utilization of each host is calculated using the concept of total number of VMs running within a datacenter and the current amount of RAM requested by all of them. This quantity when divided by the total RAM allocated to the host under consideration is done we obtain the utilization of that particular host. The utilization of host-1 at time instant‘t’ is given by H_{1t}. Here R_i represents the RAM capacity of each VM within Host-1. ‘$\mathrm{R_h}$’ gives the total RAM allocated to host-1.

$$H_{1t} = \frac{\sum_{i=1}^{n} R_i}{\mathrm{R_h}} \quad (11)$$

Hence we obtain the matrix of utilization history of all hosts for ‘n’ particular time instants. This utilization matrix is used for calculating the MAD (Median of absolute deviation) for each host. The process is explained below.

Journal of Theoretical and Applied Information Technology
20th July 2013. Vol. 53 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$H_{ij} = \begin{bmatrix} H_{11} & \cdots & H_{1n} \\ \vdots & \ddots & \vdots \\ H_{n1} & \cdots & H_{nn} \end{bmatrix} \quad (12)$$

Here the matrix H_{ij} represents the utilization values of all the hosts at 'n' time instants. We have the values of these host utilization arranged in oldest to the latest fashion with respect to time. At any particular time instant say T_1 we consider utilization values of all hosts from the first column represented by the variable 'H_1'

$$H_1 = [H_{11}, H_{21}, \ldots\ldots\ldots\ldots H_{n1}] \quad (13)$$

We calculate the median of array 'H_1' and name it as 'M_1'.We calculate the deviation array and name it as delta 'D_1', D_1 can be represented as shown below:

$$D_1 = [M_1 - H_{11}; M_1 - H_{21}; \ldots\ldots\ldots\ldots M_1 - H_{n1}] \quad (14)$$

We calculate the median of the deviation array D_1 which in turn is represented by 'X_1'. Similarly we calculate the X_i value for all the 'n' time instants represented as

$$X_i = \{X_1; X_2; \ldots\ldots\ldots\ldots X_n\} \quad (15)$$

At a particular instant T_1 we compare the values of utilization for all hosts i.e. $H_1 = [H_{11}, H_{21}, \ldots\ldots\ldots\ldots H_{n1}]$ with X_1 and all those hosts with utilization value greater than X_1 is declared as overutilised. Hence we set an array of overutilised hosts named H_0 comprising all those hosts with its utilization value greater than X_1, i.e., $H_{i1} > X_1$ .This array of overutilsed hosts is deducted from the list of total hosts to obtain the array of underutilized hosts. The array of underutilized hosts is being considered while allocating new VMs to the data centers. This list of underutilized hosts will also be considered for switching off of data centers thereby reducing power consumption by allocating the current VMs in these data centers to other hosts without affecting the tasks performed.

The simulation results of running our EEARHO algorithm along with its Pseudo code are given below:

```
Algorithm RHO:
Input :Host list              Output: H_0; H_u
all the hosts are loaded in to an array
For each host in hostlist do
   hostlistadd(host.utilisation)
Hostlist.update
   If host.utilization > threshold do
         Overutilisedhost.list(update H_0 )
         If host not an element of overutilisedhost.list do
                 Underutilized hostlist.add(hosts)
                 underutilisedhost.list(update H_u )
Return migration map H_0; H_u
```

4 RESULT:

From various studies using cloudsim simulation software involving PlanetLab workload we found that the performance of EEARHO has been significantly better than EEARRT and EEARVS. The EEARHO technique stands unique in reducing the energy consumption of hosts to a significantly lower limits.Inside EEARHO coding we have implemented the Ram consolidation (R_h) for utilization calculation. When simulated for the MAD statistical technique we obtained results that showed significant improvements in terms of energy. The improvement in energy can be explained by the effectiveness of implementing a greater number of instructions per second which in turn has been attributed by the introduction of H_i. The performance improvement can be attributed to the efficiency with which we could predict the future R_h requirement of the host. By applying median of absolute deviation and calculating X_i for any time instant helped us in grouping hosts in to H_0 list thereby increasing the energy efficiency by improved Ram allocation.

Table 2: Energy Vs AVGSLAV

	RHOMMT	IQRMMT	LR MMT	RRTMM T
ENERGY	85.32	117.08	116.71	130.96
AVGSLAV	17.02	10.72	10.42	10.25

From table 2 we can see that by implementing RHO algorithm for host overloading detection we could

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

significantly reduce the overall energy consumed by the data centers. But there is a tradeoff between average SLA violation and energy as it is evident from graph. When we were successful in reducing the energy we had an increase in SLAV which in turn proves the fact that energy efficiency and SLAV are inversely related. From figure 1 we can see that the RHO has 37.22% better performance in terms of energy efficiency when compared to IQR algorithm for host overloading detection.

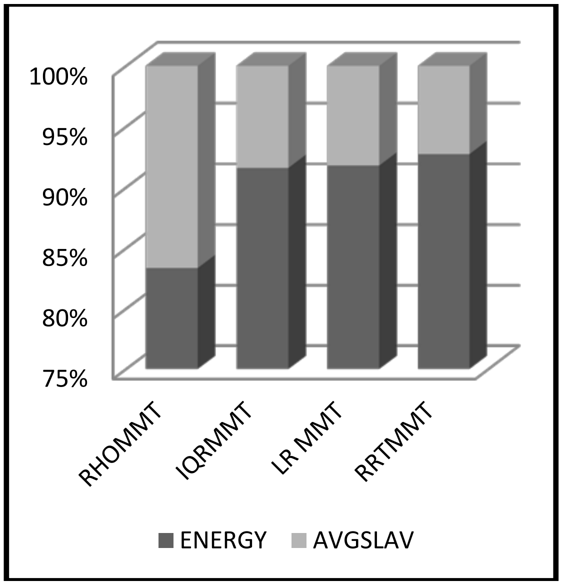

Figure 1:Energy Vs AVGSLAV

PDM Vs SLAV(MMT):

From table 3 we can see that RHO has significantly better performance in terms of PDM when compared to other host overloading detection algorithms. Here again we can see that PDM and SLAV has a trade off ,as we try to improve PDM we have compromises in SLAV. From figure 2 we can see that RHO has 90% improvement in PDM when compared to IQR.

Table 3: PDM Vs SLAV

	RHOMMT	IQR MMT	LR MMT	RRTMMT
PDM	0.01	0.1	0.06	0.11
SLAV	0.0064	0.0051	0.0025	0.0056

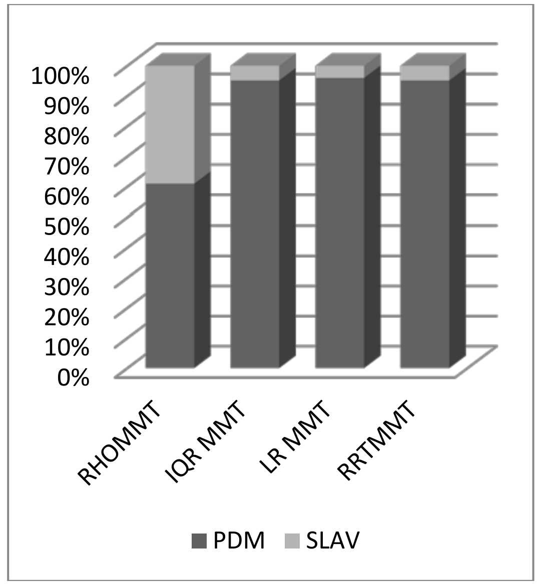

Figure 2: PDM Vs SLAV

ENERGY Vs AVGSLAV (MC)

In table 4 for VM selection algorithm MC (Maximum Correlation) we have compared the performance of host overloading detection algorithms RHO, IQR, LR, RRT and MAD. From figure 3 we can see that RHO has been able to give 37.26 % improvement over IQR in terms of energy efficiency. Here again we can see a tradeoff between AVGSLAV and Energy. As we try to improve upon energy factor we have a trade off in AVGSLAV.

Table 4: Energy Vs AVGSLAV

	RHOMC	IQRMC	LR MC	RRTMC
ENERGY	85.21	116.96	116.75	131.16
AVGSLAV	16.57	10.41	10.05	10.46

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

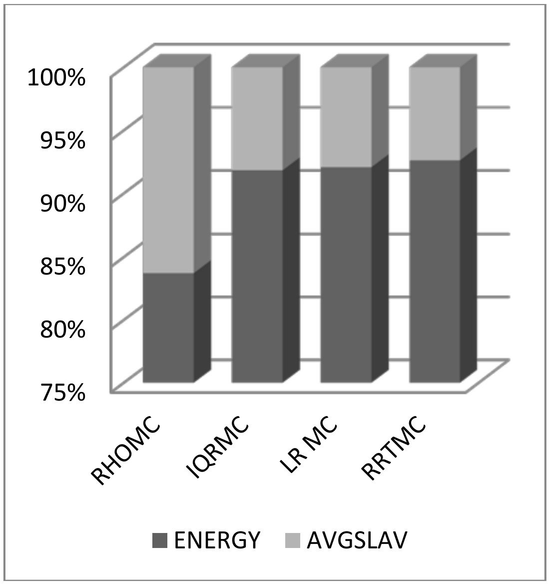

Figure 3:Energy Vs AVGSLAV

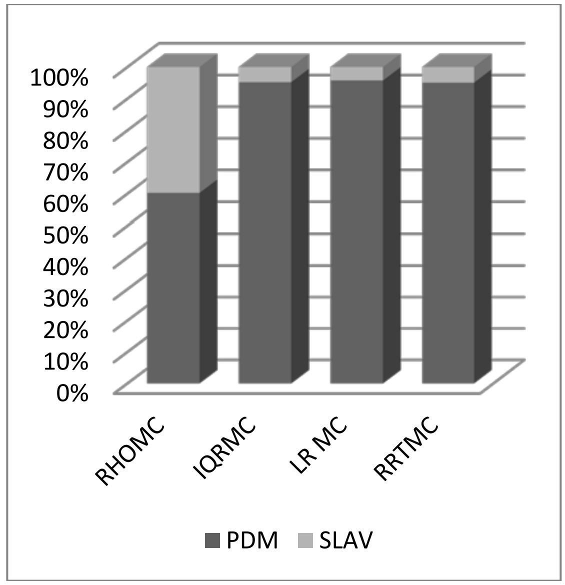

Figure 4: PDM Vs SLAV

PDM Vs SLAV (MC):

In table 5 we compare performance of MC along with host overloading detection algorithms like RHO, LR, MAD, RRT and IQR. We can see that there is a tradeoff between PDM (power degradation due to migration) and SLAV. As we try to improve up on PDM we have a faceoff in SLAV. In figure 4 we can see that there is 110% improvement in terms of PDM by implementing RHO over IQR.

Table 5: PDM Vs SLAV

	RHOMC	IQRMC	LR MC	RRTMC
PDM	0.01	0.12	0.07	0.12
SLAV	0.0066	0.006	0.0031	0.00624

5 SUMMARY and FUTURE scope:

To maximize their ROI cloud service providers targets striking a deal between energy and SLAV. With minimum resources they aim at providing quality service to its customers and this can be made possible only with efficient resource allocation algorithm. We have implemented new resource allocation algorithms by working in host overloading detection. Our analyses have shown that significant energy improvements could be achieved when compared to the existing power aware resource allocation algorithms. This energy conservation has been brought about by compromising with SLAV by maintaining the latter with in safe limits. Further works can be aimed at producing better energy SLAV tradeoff so that the cloud scenario could be made more efficient and robust.

REFRENCES:

[1] J. Tai, J. Zhang, J. Li, W. Meleis, and N. Mi, “ArA: Adaptive resource allocation for cloud computing environments under bursty workloads,” in *Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International*, 2011, pp. 1–8.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2] B. Malet and P. Pietzuch, "Resource allocation across multiple cloud data centres," in *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*, 2010, p. 5.

[3] T. Imada, M. Sato, and H. Kimura, "Power and QoS performance characteristics of virtualized servers," in *Grid Computing, 2009 10th IEEE/ACM International Conference on*, 2009, pp. 232–240.

[4] Y. Li and Y. Huang, "TMemCanal: A VM-oblivious Dynamic Memory Optimization Scheme for Virtual Machines in Cloud Computing," in *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, 2010, pp. 179–186.

[5] Y. Mei, L. Liu, X. Pu, S. Sivathanu, and X. Dong, "Performance analysis of network i/o workloads in virtualized data centers," 2011.

[6] M. Marzolla, O. Babaoglu, and F. Panzieri, "Server consolidation in clouds through gossiping," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, 2011, pp. 1–6.

[7] F.F. Moghaddam, M. Cheriet, and K.K. Nguyen, "Low carbon virtual private clouds," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, 2011, pp. 259–266.

[8] Y. Gao, H. Guan, Z. Qi, B. Wang, and L. Liu, "Quality of Service Aware Power Management for Virtualized Data Centers," *Journal of Systems Architecture*, 2013.

[9] H. Viswanathan, E. Lee, I. Rodero, D. Pompili, M. Parashar, and M. Gamell, "Energy-aware application-centric vm allocation for hpc workloads," in *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, 2011, pp. 890–897.

[10] L. Wang, H. Wang, L. Cai, R. Chu, P. Zhang, and L. Liu, "A Hierarchical Memory Service Mechanism in Server Consolidation Environment," in *Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on*, 2011, pp. 40–47.

[11] W. Shi and B. Hong, "Towards profitable virtual machine placement in the data center," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, 2011, pp. 138–145.

[12] E. Feller, C. Rohr, D. Margery, and C. Morin, "Energy Management in IaaS Clouds: A Holistic Approach," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, 2012, pp. 204–212.

[13] L. Hu, K.D. Ryu, D. Da Silva, and K. Schwan, "v-Bundle: Flexible Group Resource Offerings in Clouds," in *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*, 2012, pp. 406–415.

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

[15] T.R.V. Anandharajan and M.A. Bhagyaveni, "Co-operative Scheduled Energy Aware Load-Balancing technique for an Efficient Computational Cloud," *International Journal of Computer Science Issues*, vol. 8, pp. 571–576, 2011.