Fuzzy Based Energy Efficient Resource Allocation in Cloud Computing

Ramesh T

Sathyabama University, Chennai, India

Suresh. R. M

Sri Lakshmi Ammal Engineering College, Chennai, India

ABSTRACT

An energy efficient cloud computing model is requisite for applications over cloud's server farms, and including a quantitative sense of relational energy consumption with resource distribution while allocating virtual machines for sensible work process. In any case, it is as still a daunting problem to coordinate computational work process and its ensuing executions in an energy-effective situation over cloud platform, as the expanding operational costs of cloud computing is noteworthy toward cloud server farms all over the world. This study introduces the computational system to demonstrate the effective energy-utilization in cloud server farms and give a way to productively distribute assets in scenarios of busy network. Here, we demonstrate the dynamic algorithm for dissemination of resources of virtual machines for consistent and high performance workflow process and its ensuing executions on icanCloud and OMNET++ simulator

KEYWORDS

Energy-aware method, resource allocation, scientific workflow, cloud computing

I. INTRODUCTION

As of late, cloud computing technology has risen as a viable and proficient method for resource provisioning. Because of the centralized administration of cloud framework, clients can access resources on-demand and billed in a "pay-per-use" model [1], [2]. Owing to the scalability of cloud frameworks, an expanding number of clients utilize applications such as scientific computing and business analytics [3]. An extensive number of fields such as bioinformatics, cosmology, astronomy, and high-energy material science are influenced by computing platform supported by cloud technology [4-5]. Such logical workflow processes can profit by huge scale cloud frameworks.

A solitary logical workflow process more often than not contains hundreds or a large number of assignments, subsequently requiring a lot of computing assets for execution. Luckily, those assets can be provisioned by the cloud frameworks [6-7]. Notwithstanding, the assignments contained in logical workflow processes have variations and uncontrolled interchanges. In this way, the cloud administration framework needs to dispense asset for logical workflow process for efficiently to ensure energy saving resource distribution. Within a cloud platform, the computing assets are given to virtual machines (VMs). The VMs are generally given in different specs, which are quantitatively measured by a few setup parameters including the quantity of CPU centres, the measure of memory, the circle limit, and so on [8].

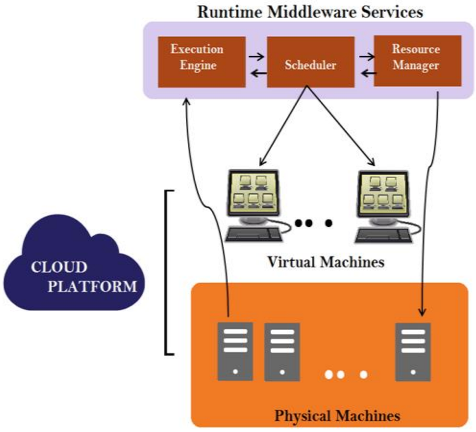

Fig 1: Illustration of Resource Allocation on Cloud Platform.

As the logical work process executions in cloud framework to work around tremendous energy utilization, it is critical to manage VMs in a proficient way to save more energy. This causes the energy utilization of a cloud computing platform to increase wide computing request on lease all throughout the world [9]. In a cloud framework several servers devoured huge amount of energy inform of electricity, processing power and cooling support [10]. Thereby, causing a lot of CO2 emanation in the atmosphere; which on quantitative scale average around 2% of worldwide CO2 emanation [3]. Therefore, it is of central significance to model such energy consumption on quantitative scale in order to come up with several measures for reduction in energy usage.

In this study, we exhibited the model for estimating energy consumption in cloud platforms . As other work process models can be changed into a successive model by present develop systems [11], just the consecutive work process is examined by simulating computing scenario of cloud.

II. METHODOLOGY: ENERGY CONSUMPTION MODEL OF DYNAMIC RESOURCE ALLOCATION

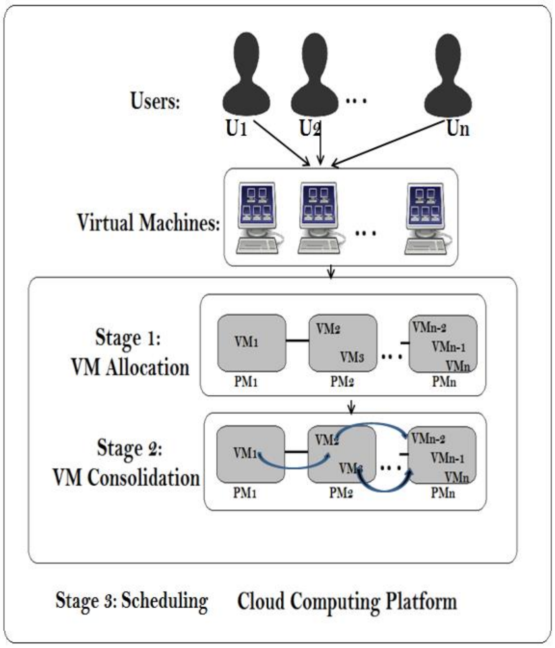

Fig 2: Illustration of the several stages of computing in the cloud computing platform.

An execution of a computational program in cloud platform is divided into three phases such as: VM Allocation phase, VM consolidation phase and interaction phase between physical machine (PM), virtual machine and users to schedule overall computing operations [12-15]. Thus, the total energy (E_T) for the overall execution of a program in a cloud platform at time t can be represented mathematically in form of the following equation as:

$$E_T = E_P + E_S + E_{VM} \ (1)$$

$$E_T = \left(c + j\sqrt{\log_2 n}\right)t_f + n.\theta.t_c + t_c(n)(2)$$

Where, E_P is the energy of the execution in time t over the physical machine which is accomplished to achieve the division of task in paralleling sequence, E_S is the energy for execution of scheduling operation of PM, VM and n number of overall connected users in time t over all the multi-processors. Also, c is the number of cycles, n is the number of processors, t_f is the average time to execute a flop by the processor, t_c is the time for the load balancing of the cloud's multiprocessor while communicating jobs b/w processors for the division and fetching of jobs, θ represents communication to communication ratio (from VM to PM) and t_c is the time taken for the computational operation over a PM processor.

Now, we need to reduce the time taken for the scheduling of the jobs and communication overheads for VM-PM processing. Therefore, the objectives are respectively divided into two parts such as:

1. Synchronously Parallel Bulk Sequencing of computational Jobs.
2. Grid-wise Reduction of Computation to Communication Ratio.

Since, threading and queuing of the jobs is differed by the architecture of the system, thus a lot of libraries had been already built upon it. However, to reduce the complexity of VM based queuing process we outlined our work upon the novel combination of page ranking method for the effective memory utilization in order to parallelize the number of queued processing jobs. This algorithm replaces the recursive rounds of I/O with one step. This is summed in the following algorithm:

Algorithm: ***Synchronously Parallel Bulk Sequencing Algorithm***

Input: N number of algorithm or parts of algorithm to be executed (V & E), n number of processors, l=cost of synchronization, g=bandwidth

Output: S_{cc} superstep which consists of communication step b/w VM-PM, computation steps of parallelizing and the synchronization step.

Step 1: Evaluate & Initialize a partition matrix:

$$\sum_{i=1}^{cc} \rho_{V,E} = \begin{cases} 1 \ if \ g \geq \dfrac{\sum_{i=0}^{c} w}{\sum_{j=1}^{c'} l} \\ 0 \ if \ g > \dfrac{\sum_{i=0}^{c} w}{\sum_{j=1}^{c'} l} \end{cases} (3)$$

where, V & E are the vertex and edge of the graph of the memory page, w is the local computation in process. Also, c is the number of iteration in computation and c' is the number of communication overheads.

Step 2: Compute the page rank of the vertex:

$$\varphi_V = \frac{n - E}{|V|} + \sum_{i=1}^{cc} \rho_{V,E}\, \varphi_E + l * g \ (4)$$

Step 3: Calculate the superstep by:

$$S_{cc} = \sum_{i=1}^{cc} \rho_{V,E} * \varphi_V \quad (5)$$

Step 4: End Process.

This reduces the variant of the memory page and maps the parallelization of computational jobs on VM in one go. Additionally, it emulates the optimized mapping of programming model over VM-PM framework. The other methods usually has one component per vertex, but the proposed algorithm uses the single balancing equation for parallelization depending upon the allowable bandwidth in synchronous with the computational workload that to one iteration based on the page rank equations [16-24]. This reduces the memory mapping and thus prioritizes the jobs based on page ranking [25-31]. Irrespective of the shuffling the algorithm eschews the hashing table for effective memory utilization with respect to the amount of the jobs required for balancing the work load.

Now, for grid-wise reduction of computation to communication ratio b/w VM & PM we model the process by using frames as a unit of given jobs and default allocation of minimum permissible processing time. A content element can be represented as r items for which there is a sequence of m number of frames given by $X \in \{{X_{x_0}}^{(1)}, {X_{x_0}}^{(2)}, \dots, {X_{x_0}}^{(n)}\}$. Here X is the set of possible values of a frame. A frame could be a short video segment, a short sequence of processors blocks of jobs, or a short processing segment. Content frames may overlap spatially, temporally, or both. Here, overlapping time windows is 2 sec long and starts every 185 ms; with overlap of 15/16 are used as frames for each instances. Let us suppose that C, B , and Y be matrix of filtered output, Y be the matrix of filters for stimulant variable and response variable for each X, such that C = XBY . Then C is a superframe of B. The length of a frame S is equivalent to the total number of frames in it and is denoted by |S|.

Now,

$$X_{x_0} = \begin{bmatrix} {X_{x_0}}^{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & {X_{x_0}}^{(r)} \end{bmatrix}, \quad {X_{x_0}}^{(j)} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, j = 1,2, \dots r \quad (6)$$

whereX_{x_0} is the covariance map from X which asserts to the association formed between the frames $S(t)$with that of stimulant and response variable. *X_i* is the position of input record. *C_l* is the cluster value which contains various values from 1 to *l*.

Now, at each step we calculate:

$$S(t) = \sum_{i=-\infty}^{+\infty} \sum_{k=1}^{N_{SC}} C_{li} S_l(t - iT_s) \quad (7)$$

$$S_l(t) = \prod(t) e^{j2\pi f_l t} \quad (8)$$

$$\prod(t) = \begin{cases} 1, 0 < t \le T_s \\ 0, t \le 0, t > T_s \end{cases} (9)$$

Where C_{li} is the i^{th} information set at the l^{th} subcarrier for the communication (when output of one iteration is propagated to the input of the other VMs), T_s is the symbol period ,S_l is the waveform for the l^{th} subcarrier, $N_{sc}$is the number of subcarriers(number of matching iterations), f_l is the frequency of the subcarrier, and π(t) is the pulse shaping function. Follow this process to complete the dataset in all records. Thus, the dynamics of the equation for a computational job is dictated by the communication rule as shown below:

At time t, when page p_t is requested by PM devices:

Then,

Fetch $x(p_t, r(p_t, t))$ ← $[forwardshare(i,j), backwarshare(i,j)]$. (It will increase in times t' > t.

where,

$$forwardshareVM(i,j) = \sum_{\substack{k\le i\\ l>j}} \left\langle \begin{matrix} forward\ weight\ of\ reference \\ from\ process\ task\ k\ to\ memeory\ instance\ l \end{matrix} \right\rangle, (10)$$

$$backwarshareVM(i,j) = \sum_{\substack{k>i\\ l\le j}} \left\langle \begin{matrix} backward\ weight\ of\ reference \\ from\ prcoess\ task\ k\ to\ memeory\ instance\ l \end{matrix} \right\rangle, (11)$$

$$z(i) = number\ of\ memeory\ insertion\ points\ among\ F_1, F_2, \ldots F_j, and$$

$$z(j) = index\ of\ last\ bottom\ inserting\ points\ among\ F_1, F_2, \ldots F_j,$$

Compute weighted relationship between shared mutations to form a sequence of functions to determine the feasible sequence of to form for a tree branch T in 2D relationship:

$$R_{i+1} = \left(\int_I |T_i(x-1, y-1) - T_i(x+1, y+1)|\, dx \right)^{\frac{1}{2}} (12)$$

Check if$R_{i+1} < R_{Max}$

based on weighed ordering in history indices of tree structure:

$$H_{t+1} = \{s_1, a_1, r_1, \ldots, s_{t+1}, a_{t+1}, r_{t+1}\}$$

$$H(i,p) \leftarrow \left(\sum_{t=t(i,j)+1}^{t(i,j+1)-1} R_j(t) \right) - z(i,j) - w(i), (13)$$

here, $x(i, p)$ be an indicator to the event that the solution is in state i during the p_{th} phase of memory instance and n_i be the number of phases of state$i$. Thus, forming a) forming a dynamic sequence. Thus the overall minimize energy can be summed into one as:

$$E(t) = \sum_{i=1}^{cc} H(i,p).S_{cc}\ (14)$$

III. RESULTS & DISCUSSION

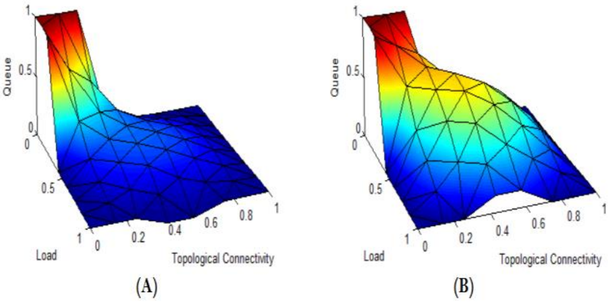

Fig 3: (A) Selection of frames queue, load and evolving topological connectivity of VM-PM before the simulation, (B) Selection of frames queue, load and evolving topological connectivity of VM-PM after the simulation.

As per the basic model, an idle server which utilizes around 60% of its peak load in order to keep resources running in the background, whereas the rest of the power is consumed by the multiprocessors and scaled linearly with the corresponding computing load [36, 37]. More accurate models [38] implies towards a more non-linear approach for model belonging to power consumption. In this study, we utilize a more in depth model of resource allocation for reduction of energy usage, which is analyzed by several performance results of the existing methods for different configuration of severs (PM-VM relationship) from a series of manufactures [39-41]. Figure 3 shows a proper VM-PM topological module and a queue based on their traffic load to adjust the congestion state reflected in fig 3 (A) by the increased queue occupancy. Here, fig 3(B) shows the evolved VM-PM topological connectivity to adjust the load and dynamically super-step calculation for the execution of jobs. Here, the energy consumption profile of the server's VM and PM is adjusted with the increased workflow as shown in figure 4 (A) & (B) respectively. The simulation parameter for the experiment is shown in table I below.

Table I: Simulation Parameter

	Parameter	Values
Topology	Core switches	4
	Aggregation switches Access switches	3
	Servers	9
	Access links	1Gb/s
	Aggregation links	5Gb/s
	Core links	5Gb/s
Data Centre	Average load of data centre	0.4
	Simulation time	30 minutes
	Distribution of job arrival	Exponential

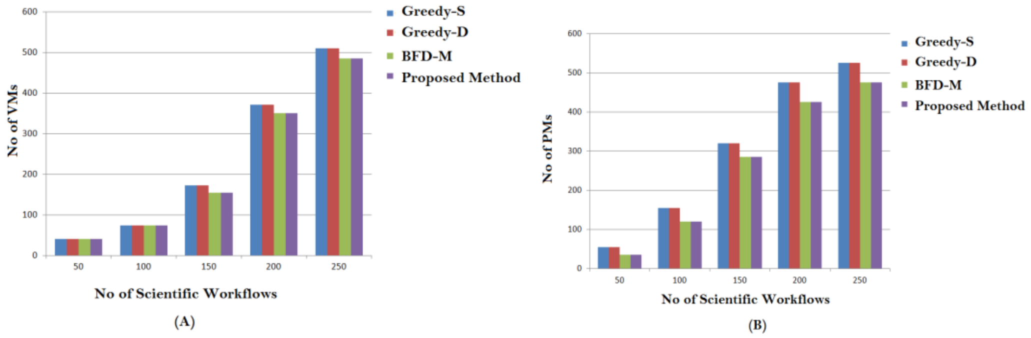

Fig 4: (A) Comparative performance evaluation of proposed method with other existing methods for VMs, (B) Comparative performance evaluation of proposed method with other existing methods for PMs.

Numerous proficient advancements were made for resource allocation in cloud platform, including Dynamic Voltage Scaling/Dynamic Voltage and Frequency Scaling (DVS/DVFS) innovation, Greedy-S, Greedy-D, BFD-M (Best Fit Decreasing). Also, Greedy-S, Greedy-D among these has been ended up being a successful and promising method for resource allocation for a much more extensive scope of cloud frameworks. Nonetheless, these innovations neglected to think about the virtualized environment and the inner communication b/w VM-PM. In the virtualized cloud environment, reconciliation of adaptive administration systems and approaches with the virtualization innovations, i.e., live relocations and hub mode exchanging arrangements is made possible. The energy utilization of every PMs is created by the multicore processors, reserve, memory and hard plate. Be that as it may, in the virtualized cloud computing platform, the energy utilization is assessed by the proposed model through two sections, i.e., Synchronously Parallel Bulk Sequencing of computational Jobs and Grid-wise Reduction of Computation to Communication Ratio. In a cloud environment, the energy utilization is estimated to demonstrate the virtual machine control utilization by weighted parameters and its topological relationship by physical hardware or PMs. Since, proposed energy utilization is displayed as an integrated social affair of the several modules at the play. This models of CPU, Memory, and communication and with other hardware assets requires to be assessed by a PM control, which principally alludes to CPU and Memory. Nonetheless, in this novel VM-PM control utilization displayed treatment of the resources as separated into two classifications. To decrease the energy utilization of processors in cloud server by utilizing the advances of VM combination and VM-PM live relocation is implemented through a heuristics planning calculation for element reallocation of VMs utilizing live relocation to accomplish the energy reserve funds and changing inert PMs to rest mode.

IV. CONCLUSION

An energy aware method for intelligently allocation of computational resources based on VM and PM is presented in this work. Due to its dynamic organization for logical workflow process and its executions the model dynamically adjust with the computational processing on multicores of cloud platform for VM-PM configurations for effective energy utilization of the application running over intermediary stages of cloud computing. This ought to be enhanced for meeting its specific needs for high performance scientific computing over cloud.

REFERENCES

[1] Z. Xiao, W. Song, and Q. Chen, “Dynamic resource allocation using virtual machines for cloud computing environment,” IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1107–1117, Jun. 2013.

[2] S. Ibrahim, B. He, and H. Jin, “Towards pay-as-you-consume cloud computing,” in Proc. 8th IEEE Int. Conf. Service Comput., Jul. 2011, pp. 370–377.

[3] X. Zhu, L. Yang, H. Chen, J. Wang, S. Yin, and X. Liu, “Real-time tasks oriented energy-aware scheduling in virtualized clouds,” IEEE Trans. Cloud Comput., vol. 2, no. 2, pp. 168–180, Jun. 2014.

[4] X. Liu, Y. Yang, Y. Jiang, and J. Chen, “Preventing temporal violations in scientific workflows: Where and how,” IEEE Trans. Softw. Eng., vol. 37, no. 6, pp. 805–825, Nov. 2011.

[5] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, “Characterizing and profiling scientific workflows,” Future Gener. Comput. Syst., vol. 29, no. 3, pp. 682–692, Mar. 2013.

[6] R. N. Calheiros and R. Buyya, “Meeting deadlines of scientific workflows in public clouds with tasks replication,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 7, pp. 1787–1796, Jul. 2014.

[7] X. Fei and S. Lu, “A dataflow-based scientific workflow composition framework,” IEEE Trans. Services Computing, vol. 5, no. 1, pp. 45–58, Mar. 2012.

[8] D. Warneke and O. Kao, “Exploiting dynamic resource allocation for efficient parallel data processing in the cloud,” IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 6, pp. 985–997, Jun. 2011.

[9] J. Baliga, R. W. Ayre, K. Hinton, and R. S. Tucker, “Green cloud computing: Balancing energy in processing, storage, and transport,” Proc. IEEE, vol. 99, no. 1, pp. 149–167, Dec. 2011.

[10] A. Jain, M. Mishra, and S. K. Peddoju, “Energy efficient computing- green cloud computing,” in Proc. 1st IEEE Int. Conf. Energy Efficient Technol. Sustainability, Apr. 2013, pp. 978–982.

[11] L. Qi, Y. Tang, W. Dou, and J. Chen, “Combining local optimization and enumeration for qos-aware web service composition,” in Proc. IEEE Int. Conf. Web Services, Jul. 2010, pp. 34–41.

[12] X. Li, J. Wu, S. Tang, and S. Lu, “Let’s stay together: Towards traffic aware virtual machine placement in data centers,” in Proc. 33rd IEEE Int. Conf. Computer Commun.), Apr. 2014, pp. 1842–1850.

[13] Y. Guo, A. L. Stolyar, and A. Walid, “Shadow-routing based dynamic algorithms for virtual machine placement in a network cloud,” in Proc. 32nd IEEE Int. Conf. Comput. Commun., Apr. 2013, pp. 620–628.

[14] G. Terzopoulos and H. Karatza, “Performance evaluation and energy consumption of a real-time heterogeneous grid system using DVS and DPM,” Simul.Model. Practice Theory, vol. 36, pp. 33–43, May 2013.

[15] Y. H. Du, P. C. Xiong, Y. S. Fan, and X. Li, “Dynamic checking and solution to temporal violations in concurrent workflow processes,” IEEE Trans. Man Cybern. A, Syst. Humans, vol. 41, no. 6, pp. 1166–1181, Nov. 2011.

[16] J. Zheng, T. S. E. Ng, K. Sripanidkulchai, and Z. Liu, “Pacer: A progress management system for live virtual machine migration in cloud computing,” IEEE Trans. Netw. Service Manag., vol. 10, no. 4, pp. 369–382, Dec. 2013.

[17] D. Meisner, B. T. Gold, and T. F. Wenisch, “PowerNap: Eliminating server idle power,” ACM SIGARCH Comput. Archit. News, vol. 37, no. 1, pp. 205–216, Mar. 2009.

[18] Y. C. Lee and A. Y. Zomaya, “Energy efficient utilization of resources in cloud computing systems,” J. Supercomput., vol. 60, no. 2, pp. 268–280, May 2012.

[19] C. Isci, J. Liu, B. Abali, J. O. Kephart, and J. Kouloheris, “Improving server utilization using fast virtual machine migration,” IBM J. Res. Develop., vol. 55, no. 6, pp. 1–12, Nov. 2011.

[20] S. He, L. Guo, and Y. Guo, “Real time elastic cloud management for limited resources,” in Proc. 3rd IEEE Int. Conf. Cloud Comput., Jul. 2011, pp. 622–629.

[21] Designing Your Cloud Infrastructure [Online]. Available: https:// technet.microsoft.com/en-us/library/hh831630.aspx, 2014.

[22] X. Meng, V. Pappas, and L. Zhang, “Improving the scalability of data center networks with traffic-aware virtual machine placement,” in Proc. 29th IEEE Int. Conf. Comput. Commun., Mar. 2010, pp. 1–9.

[23] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, “CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” Software: Practice Experience, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[24] N. Liu, Z. Dong, and R. Rojas-Cessa, “Task scheduling and server provisioning for energy-efficient cloud-computing data centers,” in Proc. 33rd IEEE Int. Conf. Distrib.Comput. Syst. Workshops, Jul. 2013, pp. 226–231.

[25] S. Zhang, B. Wang, B. Zhao, and J. Tao, “An energy-aware task scheduling algorithm for a heterogeneous data center,” in Proc. 12nd IEEE Int. Conf. Trust, Security Privacy Comput. Commun., Jul. 2013, pp. 1471–1477.

[26] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers,” Concurrency Comput.: Practice Experience, vol. 24, no. 13, pp. 1397–1420, Oct. 2012.

[27] N. J. Kansal and I. Chana, “Cloud load balancing techniques: A step towards green computing,” Int. J. Comput. Sci., vol. 9, no. 1, pp. 238–246, Jan. 2012.

[28] C. T. D. Lo and K. Qian, “Green computing methodology for next generation computing scientists,” in Proc. 34th IEEE Annu.Comput.Softw. Appl. Conf., Jul. 2010, pp. 19–23.

[29] A. Kansal, F. Zhao, J. Liu, and N. Kothari, “Virtual machine power metering and provisioning,” in Proc. 1st ACM Symp. Cloud Comput., Jun. 2010, pp. 39–50.

[30] C. Mobius, W. Dargie, and A. Schill, “Power consumption estimation models for processors, virtual machines, and servers,” IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 6, pp. 39–50, Jun. 2014.

[31] A. E. H. Bohra and V. Chaudhary, “VMeter: Power modelling for virtualized clouds,” in Proc. 24th IEEE Int. Conf. Parallel Distrib. Process., Workshops Phd Forum, Apr. 2010, pp. 1–8.

[32] B. Krishnan, H. Amur, A. Gavrilovska, and K. Schwan, “VM power metering: Feasibility and challenges,” ACM SIGMETRICS Perform. Eval. Rev., vol. 38, no. 3, pp. 56–60, Dec. 2010.

[33] M. Mezmaz, N. Melab, Y. Kessaci, Y. C. Lee, E. G. Talbi, A. Y. Zomaya, and D. Tuyttens, “A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems,” J. Parallel Distrib. Comput., vol. 71, no. 11, pp. 1497–1508, Nov. 2011.

[34] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. WetherallReducing, “Network energy consumption via sleeping and rate-adaptation,” in Proc. 5th USENIX Symp. Netw. Syst. Des. Implementation, Apr. 2008, pp. 323–336.

[35] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, “A self-tuning DVS processor using delayerror detection and correction,” IEEE J. Solid-State Circuits, vol. 41, no. 4, pp.792–804, Apr. 2006.

[36] N. B. Rizvandi, J. Taheri, A. Y. Zomaya, and Y. C. Lee, “Linear combinations of DVFS-enabled processor frequencies to modify the energy-aware scheduling algorithms,” in Proc. 10th IEEE/ACM Int. Conf. Cluster, Cloud Grid Comput., May 2010, pp. 288–397.

[37] R. Nathuji and S. Karsten, “Virtualpower: Coordinated power management in virtualized enterprise systems,” in Proc. 21st ACM Symp. Operating Syst. Principles, Oct. 2007, pp. 265–278.

[38] A. Beloglazov and R. Buyya, “Energy efficient resource management in virtualized cloud data centers,” in Proc.10th IEEE/ACM Int. Conf. Cluster, Cloud Grid Comput., May 2010, pp. 826–831.

[39] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,” Future Gener.Comput. Syst., vol. 28, no. 5, pp. 755–768, May 2012.

[40] M. A. Rodriguez and R. Buyya, “Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds,” IEEE Trans. Cloud Comput., vol. 2, no. 2, pp. 222–235, Apr. 2014.