Metadata of the chapter that will be visualized in SpringerLink

Book Title	Internet and Distributed Computing Systems
Series Title	
Chapter Title	Fuzzy Logic Based Energy Aware VM Consolidation
Copyright Year	2015
Copyright HolderName	Springer International Publishing Switzerland

Corresponding Author	Family Name	**Monil**
	Particle	
	Given Name	**Mohammad Alaul Haque**
	Prefix	
	Suffix	
	Division	Electrical and Computer Engineering Department
	Organization	North South University
	Address	Dhaka, Bangladesh
	Email	monil01@gmail.com
Author	Family Name	**Rahman**
	Particle	
	Given Name	**Rashedur M**
	Prefix	
	Suffix	
	Division	Electrical and Computer Engineering Department
	Organization	North South University
	Address	Dhaka, Bangladesh
	Email	rashedur.rahman@northsouth.edu

Abstract	Global need of computing is growing day by day and as a result cloud based services are getting more prominent for its pay-as-you-go modality. However, cloud based datacenters consume considerable amount of energy which draws negative attention. To sustain the growth of cloud computing, energy consumption is now a major concern for cloud based datacenters. To overcome this problem, cloud computing algorithm should be efficient enough to keep energy consumption low and at the same time provide desired QoS. Virtual machine consolidation is one such technique to ensure energy-QoS balance. In this research, we explored Fuzzy logic and heuristic based virtual machine consolidation approach to achieve energy-QoS balance. Fuzzy VM selection method has been proposed to select VM from an overloaded host. Additionally, we incorporated migration control in Fuzzy VM selection method. We have used CloudSim toolkit to simulate our experiment and evaluate the performance of the proposed algorithm on real-world work load traces of PlanetLab VMs. Simulation results demonstrate that the proposed method provides best performance in all performance metrics while consuming least energy.

Fuzzy Logic Based Energy Aware VM Consolidation

Mohammad Alaul Haque Monil(✉) and Rashedur M. Rahman

Electrical and Computer Engineering Department, North South University, Dhaka, Bangladesh
monil01@gmail.com, rashedur.rahman@northsouth.edu

Abstract. Global need of computing is growing day by day and as a result cloud based services are getting more prominent for its pay-as-you-go modality. However, cloud based datacenters consume considerable amount of energy which draws negative attention. To sustain the growth of cloud computing, energy consumption is now a major concern for cloud based datacenters. To overcome this problem, cloud computing algorithm should be efficient enough to keep energy consumption low and at the same time provide desired QoS. Virtual machine consolidation is one such technique to ensure energy-QoS balance. In this research, we explored Fuzzy logic and heuristic based virtual machine consolidation approach to achieve energy-QoS balance. Fuzzy VM selection method has been proposed to select VM from an overloaded host. Additionally, we incorporated migration control in Fuzzy VM selection method. We have used CloudSim toolkit to simulate our experiment and evaluate the performance of the proposed algorithm on real-world work load traces of PlanetLab VMs. Simulation results demonstrate that the proposed method provides best performance in all performance metrics while consuming least energy.

1 Introduction

Cloud computing services are getting more popular for its scalability, reliability and pay-as-you-go model. Techno-giants have already started providing cloud services and IT companies are now moving from traditional CAPEX model (buy the dedicated hardware and depreciate it over a period of time) to the OPEX model (use a shared cloud infrastructure and pay as one uses it). To cope up with the ever increasing need of computing, cloud service providing companies are now using warehouse sized datacenters to meet user demand which incurs considerable amount of energy. At the beginning of this cloud computing era, cloud service providers were focused mainly on catering the computing demand that leads to expansion of cloud infrastructures; hence energy consumption. For these reasons, energy consumption by data centers worldwide has risen by 56 % from 2005 to 2010, and in 2010 is accounted to be between 1.1 % and 1.5 % of the total electricity use. Moreover, carbon dioxide emissions of the ICT industry are currently estimated to be 2 % of the global emissions which is equivalent to the emissions of the aviation industry [4].

VM Consolidation is one of the techniques which draws researchers' attention and is an active field of research in recent time. VM consolidation method makes the

G. Di Fatta et al. (Eds.): IDCS 2015, LNCS 9258, pp. 1–8, 2015.
DOI: 10.1007/978-3-319-23237-9_4

underutilized servers shut-down by increasing the utilization of active datacenters. As we know that inactive datacenter or datacenter in sleep mode causes minimal energy and in this way energy consumption can be reduced considerably. However, to achieve this outcome, we need to consolidate different VM in one server and migrate VMs from datacenter to datacenter which may lead to SLA violation. So, algorithms must be designed in such a way that not only reduces power consumption but also serves desired QoS (such as SLA).

2 Related Works

VM consolidation algorithm needs to be designed in such a way that there will be minimum energy consumption, minimum violation of SLA, efficient VM migration and minimum number active hosts in a given time. Considerable number of researches has been conducted for VM consolidation using various methods based on heuristics.

In [1, 2, 4], Beloglazov et al. proposed heuristic based approach to deduce thresholds thorough different statistical measures. VM Consolidation problem is divided into sub-problems and algorithms for each sub-problem had been designed. The sub-problems are: (i) Under load detection, (ii) Overload detection, (iii) VM selection and, (iv) VM placement. Heuristic based algorithms are designed for each sub-problems and designed in such a way that they can adapt and keep their threshold changing based on different scenario in different time so that they can still provide the functionality and consolidation decision in changed environment. This adaption process allows the system to be dynamic. These algorithms were implemented in CloudSim developed by Clouds lab in the University of Melbourne. References [5, 6] describe CloudSim which provides various functionalities of a cloud environment and facilitates cloud simulation. References [1, 2, 4] have also used CloudSim for simulation. The main components of CloudSim are datacenter, Virtual Machine (VM) and cloudlet. Cloudlet can be real data from real cloud. The simulator creates datacenter, Virtual Machine and cloudlet on the run based on the defined parameters. When the simulation starts, virtual machines are placed in the datacenter for processing. Sub problems (i–iv) are already developed in CloudSim. To develop further, one needs to create new class to develop new methods and test it. In [7, 8] we worked with basic VM selection algorithm and introduced migration control in the built in CloudSim VM selection methods. Farahnakian et al. [9] used the ant colony system to deduce a near-optimal VM placement solution based on the specified objective function. In [3] VM consolidation with migration control is introduced. Here VMs with steady usage are not migrated and not steady VMs are migrated to ensure better performance, the migrations are triggered and done by heuristic approaches. Main advantages of heuristics are that a static and acceptable performance could be achieved with very less amount of errors. Sheng et al. [11] designed a prediction method based on Bayes model to predict the mean load over a long-term time interval and also the mean load in consecutive future time intervals. Prevost et al. [10] introduced a framework combining load demand prediction and stochastic state transition models. They used neural network and autoregressive linear prediction algorithms to forecast loads in cloud data center applications.

3 Proposed Method

In this work we have designed Fuzzy VM Selection with migration control algorithm and VM placement and underload detection are adjusted. However, before going in detail, CloudSim overview of VM consolidation is introduced. The algorithm below portrays the basic VM consolidation approach designed in CloudSim.

Algorithm 1. VM consolidation in CloudSim

1. *Input number of hosts;*
2. *Interface with real cloud data;*
3. *VM created and assigned to hosts;*
4. *Cloudlet created and assigned to VMs;*
5. *for every specified time interval*
6. *Execute Underload detection;*
7. *Identify overloaded host through overload detection.*
8. *VM is selected for migration from overloaded host.*
9. *VM is placed in available datacenters.*
10. *Preserve history and calculate QoS*
11. *end*
12. *Simulation ended and provide Energy consumption and other QoS value*

Algorithm 1 provides a basic flow of VM consolidation in CloudSim. At first the hosts are created, then the real cloud data is taken as input. Based on the real data, VMs and cloudlets are created. Then VMs are assigned to hosts and cloudlets are assigned to VMs. Based on dynamic consolidation technique, status is checked for every scheduled interval. For every scheduled interval, underload detection algorithm is executed and less utilized hosts are put into sleeping mode by transferring all VM to other active VMs. Then overload detection is executed, and overloaded hosts are identified. At later steps, VM is selected from the overloaded hosts to migrate. Then those VMs are placed in available hosts or if needed hosts are switched on from sleeping mode. After each iteration, a log is kept to calculate energy consumption and QoS. At the end of the simulation energy consumption and QoS is shown. In the next section our proposed methods are discussed.

3.1 Fuzzy VM Selection with Migration Control

Fuzzy technique is an attractive approach to handle uncertain, imprecise, or un-modeled data in solving control and intelligent decision-making problems. Different VM selection methods offer different advantages. Therefore, if we want to generate a method which will have the benefits of all selection methods, then we can combine them together and based on the merit of the metric a fuzzy output value will be generated and our objective will be fulfilled. A set of rules of inference can be devised to generate result. So, fuzzy logic is an ideal tool for this work. It will consider all the options and depending on those options a fuzzy value will be generated based on the predetermined rules of inference. To develop the fuzzy VM selection method, we have selected three distinguished methods as metric and each of them offers some advantages over others and different researches have already proven them. The following subsections will be focusing on the metrics we will be using as inputs to our fuzzy systems, member ship functions generated, inference rules and algorithms for computation.

(1) Minimum migration Time: Minimum Migration time policy selects the VM which can be migrated within minimum time limit [2, 4]. The migration time is limited by the memory the VM is using and the bandwidth and migration control is applied. At any moment *t*, the MMT with Migration Control policy finds VM *x* that will be selected for migration by the following formula:

$$x \in V_h | \forall y \in V_h, \frac{RAM(x)}{NET_h} \leq \frac{RAM(y)}{NET_h} \tag{1}$$

This policy gives us the lowest SLA from all the VM selection models. So this will be considered as one input of the fuzzy system.

(2) Correlation: This method works based on the idea that the higher the correlation between the resource usage by applications running on an oversubscribed server, the higher the probability of server being overloaded [4]. Basically this instructs that higher correlation of CPU usage of one VM with other VM should be migrated. Migration control is applied with maximum correlation method to identify the migratable VM.

Let there are n numbers of VMs and $X_1, X_2 \ldots X_n$ is the CPU usage of n VMs which are under consideration for migration. Let Y be the VM for which we want to determine the maximum correlation with i^{th} VM. The augmented matrix for the rest is denoted by X and the $(n\text{-}1)x1$ vectored of Y is expressed by y.

$$X = \begin{bmatrix} 1 & x_{1,1} & \ldots & \ldots & x_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1,1} & \ldots & \ldots & x_{n-1,n-1} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ \vdots \\ y_n \end{bmatrix} \tag{2}$$

A vector of predicted value is denoted by ŷ.

$$\hat{y} = Xbb = (X^T X)^{-1} X^T y \tag{3}$$

Having found the predicted value the correlation coefficient is:

$$R^2_{Y,X_1,\ldots X_{n-1}} = \frac{\sum_{i=1}^{n} \left(y_i - m_y\right)^2 \left(\hat{y}_i - m_{\hat{y}}\right)^2}{\sum_{i=1}^{n} \left(y_i - m_y\right)^2 \sum_{i=1}^{n} \left(\hat{y}_i - m_{\hat{y}}\right)^2} \tag{4}$$

This is how correlation can be calculated.

(3) Migration control metric for steady resource consuming VM: It has been proven that migration control provides better result in energy aware VM consolidation. Besides, this approach saves the unwanted traffic load [3]. Migration control can be done in various ways. We can stop migrating the high CPU using VMs or we can restrict steady resource consuming VM from migration. In this work we will take steady resource consumption as a non migration factor because when a VM consumes

almost constant resource, it means it will be the least possible VM to make this host overloaded. We have used standard deviation as a calculation of migration control.

Let, there are two VMs x and y in host h and V_h be the set of VMs. $CPU_u(x_t)$ is the CPU utilization of time t, which is current time. $CPU_u(x_{t-1})$, $CPU_u(x_{t-2})$ ..... $CPU_u(x_{t-n})$ are the CPU utilization of up to previous n number time frames when overload detection algorithm was activated. So migration control parameter can be given by 5. Here $\mathrm{CPU}_{\mathrm{average}}$ means average CPU utilization for last n cycles.

$$stdev = \sqrt{\frac{1}{n}\sum_{1}^{n}\left(\mathrm{CPU_i} - \mathrm{CPU_{average}}\right)^2} \tag{5}$$

(4) Fuzzy Membership function: A FIS (Fuzzy Inference System) is developed to provide fuzzy VM selection decision using three metrics we discussed above as input. Member ship function needs to be defined to develop the FIS. We are using total 4 linguistic variables including VM selection as output. We have used trapezoidal membership function. Range of these membership functions is chosen from the simulation from Real cloud data of PlanetLab. In order to do so, we have run the simulation and collected data of all these variables and proportioned to decide the range. Membership functions of the linguistic variables are given below:

- RAM: *T(RAM) = {Low, Medium, High}*
- Correlation: *T(Correlation) = {Low, Medium, High}*
- Standard Deviation: *T(Stdev) = {Low, Medium, High}*
- VM selection: T(Vmselection) = {Low, Medium, High, Very High}

(5) Fuzzy Inference Rule: Fuzzy inference rules are generated from the given linguistic variables. We have given equal weight on the variables to influence the VM selection value. If RAM is low it gets high priority as it makes the migration faster. If correlation is high then it gets high priority in migration as the higher the correlation is, the higher the probability of host being overloaded. Finally, if standard deviation is high then it will get priority in migration compared to its steady state counterparts.

(6) Fuzzy VM selection with migration control: Combination of Fuzzy VM selection method and migration control is given in 6 & 7. These equations indicate that a VM will be nominated for migration if it produces lower CPU usage value than the migration control threshold and posses highest fuzzy value among the lower migration control value group. If all VMs of an overloaded host produce more CPU usage value than the migration control threshold, then the highest fuzzy output producing VM will be migrated. Migration control is calculated by CPU usage of last n cycle. The equations are as follows:

$$x \in V_h | \forall y \in V_h, \quad Fuzzy\ Output(x) \geq Fuzzy\ Output(y)$$

Only if;

$$\frac{[CPU_u(x_t) + CPU_u(x_{t-1}) + CPU_u(x_{t-2}) + \ldots CPU_u(x_{t-n})]}{(n+1)} < CPU_{thresold} \tag{6}$$

If for every VM vm,

$$\frac{[CPU_u(vm_t) + CPU_u(vmx_{t-1}) + \ldots + CPU_u(vm_{t-n})]}{(n+1)} \geq CPU_{thresold}$$

Then;

$$x \in V_h | \forall y \in V_h, \quad Fuzzy\ Output(x) \geq Fuzzy\ Output(y) \tag{7}$$

4 Experimental Result

We have implemented our algorithms in CloudSim 3.0.3 to evaluate the performance of our proposed method. We have considered 800 heterogeneous physical nodes, half of which are HP ProLiant G4 and the rest are HP ProLiant G5 servers. Energy consumption is calculated based on the HP ProLiant G4 and HP ProLiant G5. CPU usage and power consumption are calculated based on the data provided in Table 1. These servers are assigned with 1860 MIPS (Million Instructions Per Second) and 2660 MIPS for each core of G4 and G5 servers. Network bandwidth is considered as 1 GB/s. The VMs which were created were single core. VM were assigned 4 types. High-CPU Medium Instance (2500 MIPS, 0.85 GB); Extra Large Instance (2000 MIPS, 3.75 GB); Small Instance (1000 MIPS, 1.7 GB); and Micro Instance (500 MIPS, 613 MB) [4]. We have used real world work load data provided as part of CoMon project, a monitoring infrastructure for PlanetLab. This data are collected from more than thousand VMS of from server located more than 500 different locations. These real world traces contain VM utilization files every 5 min. Data of 10 days of 2011 has been used in this experiment. Based on two metrics the performance of the proposed method is compared.

(1) Energy Consumption (kWh): Energy consumption is computed taking into account all hosts throughout the simulation time and by mapping CPU and energy consumption from the Table 1.
(2) SLAV: Service level agreement violation, SLAV, comes from the product of overload time fraction (OTF) and performance degradation due to migration (PDM), i.e. SLAV = OTF*PDM.

Table 1. Power Consumption for different level of utilization

Machine Type	Power consumption based on CPU utilization					
	0 %	20 %	40 %	60 %	80 %	100 %
HP G4 (Watt)	86	92.6	99	106	112	117
HP G5 (Watt)	93.7	101	110	121	129	135

In CloudSim, there are five built in Overload detection algorithms (IQR, LR, LRR, MAD and THR) and three built in VM selection (MC, MMT, RS) methods. So in combination there are 15 methods (IQR_MC, IQR_MMT, IQR_RS, LR_MC, LR_MMT, LR_RS, LRR_MC, LRR_MMT, LRR_RS, MAD_MC, MAD_MMT, MAD_RS, THR_MC, THR_MMT, THR_RS) which will be compared to our proposed MSMD_FS method based on aforementioned performance metrics. Based on the result for 10 days Fig. 1 Box graphs have been prepared to compare the results. Metric wise result is given below:

(1) Energy Consumption: Basic target of this research is to design a VM consolidation algorithm so that the energy consumption is reduced. By comparing the proposed and existing methods in the first graph of Fig. 1, it is found that energy consumption is significantly reduced in Proposed (MMSD_FS) method. Minimum energy consumption by proposed method is 102 Kwh where the minimum of all other methods is 112 Kwh, 8.5 % reduction. If we consider average value, MMSD_FS consumed 136.5 Kwh and all other method consumed 169 Kwh on average resulting 19 % energy saving. So by this we can infer that the basic objective of this research is achieved by saving energy consumption.

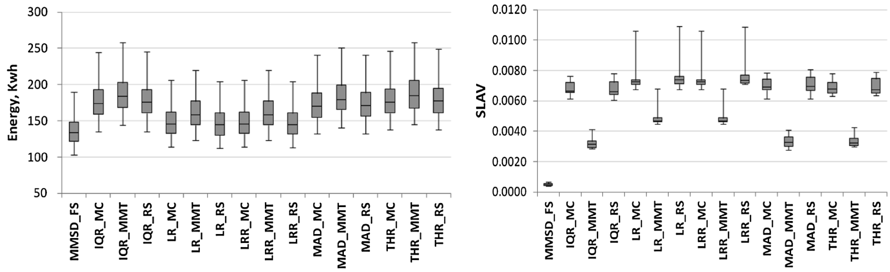

Fig. 1. Energy and SLAV is shown in these box graphs. MMSD_FS is the fuzzy VM selection method, which is the first one in the both chart and rest are the built in algorithms of CloudSim.

(2) SLA Violation (SLAV): SLA violation is one of the key indicators of QoS. SLA violation is calculated by keeping two scenarios into consideration, (1) if any VM got overloaded, and (2) the SLA violation incurred while migration. With a method having low SLA violation indicates ensuring users the desired QoS. From Fig. 1, it is clearly visible that SLA violation is considerably decreased. Minimum SLAV by proposed method is 0.0004 % whereas the minimum of all other method is 0.00279 %, resulting 84 % reduction. If we consider average value, MMSD_FS incurred 0.0005 % SLAV and all other methods incurred 0.00617 % on an average, resulting 91 % reduction in SLA violation.

From the performance metrics it can be inferred that the proposed method outperforms all other methods.

5 Conclusion

In this paper we have devised algorithm for fuzzy VM selection method and introduced migration control in the selection method. Fuzzy VM selection methods take intelligent decision to select a VM to be migrated from one host to other. After simulation and making comparison against existing methods, it has been found that the proposed method outperformed other previous methods in both perspectives, i.e., more energy saving and less SLA violation. Therefore, it can be inferred that the objective, energy-SLA trade off has been achieved in this work in an efficient manner. As a future work we have plan to improve the default VM placement and underload detection algorithm built in CloudSim to achieve more energy saving and less SLA violation.

References

1. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. Future Gener. Comput. Syst. (FGCS) **28**(5), 755–768 (2011)
2. Beloglazov, A., Buyya, R.: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurrency Comput. Pract. Experience (CCPE) **24**(13), 1397–1420 (2012)
3. Ferreto, T.C., Netto, M.A.S., Calheiros, R.N., De Rose, C.A.F.: Server consolidation with migration control for virtualized data centers. Future Gener. Comput. Syst. **27**(8), 1027–1034 (2011)
4. Beloglazov, A.: PhD Thesis: Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing (2013). http://beloglazov.info/thesis.pdf
5. Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F.D., Buyya, R.: CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Exp. **41**(1), 23–50 (2011)
6. CloudSim link. http://code.google.com/p/CloudSim/
7. Monil, M.A.H., Qasim, R., Rahman, R.M.: Incorporating migration control in VM selection strategies to enhance performance. IJWA **6**, 135–151 (2014)
8. Monil, M.A.H., Qasim, R., Rahman, R.M.: Energy-aware VM consolidation approach using combination of heuristics and migration control. In: ICDIM 2014, pp. 74–79 (2014)
9. Farahnakian, F., Ashraf, A., Liljeberg, P., Pahikkala, T., Plosila, J., Porres, I., Tenhunen, H.: Energy-aware dynamic VM consolidation in cloud data centers using ant colony system. In: 2014 IEEE 7th International Conference on Cloud Computing (CLOUD), pp. 104–111 (2014)
10. Prevost, J., Nagothu, K., Kelley, B., Jamshidi, M.: Prediction of cloud data center networks loads using stochastic and neural models. In: Proceedings of the IEEE System of Systems Engineering (SoSE) Conference, pp. 276–281, 27-30 2011
11. Di, S., Kondo, D., Cirne, W.: Host load prediction in a Google compute cloud with a Bayesian model. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Salt Lake City, UT, 10–16 November 2012

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	/ through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	/ through letter or through characters	new character / or new characters /
Change to italics	under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	/ through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	
Insert comma	(As above)	,
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	/ through character or where required	
Reduce space between characters or words	between characters or words affected	