International Journal of Cloud-Computing and Super-Computing
Vol.1, No.1 (2014), pp.15-26
http://dx.doi.org/10.14257/ijcs.2014.1.1.02

Efficient Virtual Machine Placement with Energy Saving in Cloud Data Center

Madhukar Shelar[1], Shirish Sane[2], Vilas Kharat[3] and Rushikesh Jadhav[4]

[1] *K. T. H. M. College, Nashik and Ph.D. Research Scholar, Dept. of Comp. Sc., S.P. Pune University, India*
[2] *K. K. Wagh Institute of Engineering Education and Research, Nashik, India*
[3] *Department of Computer Science, S.P. Pune University, Pune, India*
[4] *Research – Team Leader, ESDS Software Solutions Pvt. Ltd., Nashik, India*
mnshelar@rediffmail.com[1], sssane@kkwagh.edu.in[2], laddoo1@yahoo.com[3], rushikesh.jadhav@esds.co.in[4]

Abstract

Cloud data centers provide computing infrastructure as a service to their customers on pay per use basis. In virtualized data centers CPU, RAM, storage and bandwidth are allotted to a Virtual Machine (VM) from pool of shared resources. An autonomic consolidation of VMs on appropriate Physical Machine (PM) by achieving performance and saving cost is the key challenge for virtualized data centers. This paper presents a self-organizing and multi-objective approach for autonomic consolidation of VMs. The proposed approach does the initial placement of VMs in appropriate PM of cloud data centre which addresses different issues altogether such as maximum resource requirement during setup of VMs, future demand of free resources at peak load, improving the performance and energy saving by keeping idle PMs at offline state. The performance of the proposed algorithm is evaluated by simulating a data center with randomly generated resource capacities of PMs and resource requirement of VMs. Experiment results of proposed technique are also compared with standard algorithms of VM consolidation such as first-first, next-fit and random-selection on two dimensions of resources-CPU and RAM.

Keywords: *Cloud Computing, Physical Machine, Virtual Machine, Virtualization*

1. Introduction

In recent years, cloud computing is the widely used trend in distributed computing for delivering computing resources as a service over the Internet. There are three basic service models of cloud computing - Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). In IaaS model, users acquire computing resources such as processing power, memory, storage etc from cloud providers and use these resources to deploy and run their applications. Amazon web services-EC2 [2], Rackspace cloud server [21] are popular IaaS providers. Powerful data centers are essential to provide computing infrastructure as a service to the customers. To ensure high quality of services, the performance and cost of data center is a critical factor [17]. Cost of data center may be controlled by utilizing computing resources via sharing and it also improves the performance of applications. One key technology to increase resource utilization is server virtualization. Server virtualization makes it possible to execute several virtual machines (VM) concurrently on top of single physical machine (PM). Each VM hosts guest operating system, middleware and applications and provides a partition of

ISSN: 2287-8491 IJCS

underlying resource capacity (CPU, RAM, network-bandwidth and storage) of PM [27]. Though VMs share physical resources, each VM runs individually with a proprietary resource; this makes it possible to guarantee the quality of provided service [17].

The important issue in server virtualization is a *Virtual Machine Placement* (VMP). It is the process of mapping VMs on to appropriate PM by resolving constraints of customers and cloud service provider. If a data center has limited number of virtual and physical machines then an operator can manage the VM to PM placement activity manually. But as the number of VMs and PMs are increased, it may be uncontrollable for an operator because of large number of possible mappings to be explored manually [14] and thus automation becomes necessary. VMP process can be performed at two different stages – Initial placement and runtime placement. Former stage is performed at the time of deployment of applications into cloud. VMs are created for applications and place them at appropriate PM based on the resource requirements of underlying applications and Service Level Agreement (SLA) between customer and cloud service provider. If PM is unable to provide sufficient resources for VM which is already in running stage then it can be migrated to another PM to satisfy its resource demand. Such VM placement is referred to as runtime or dynamic placement. Our proposed approach is concerned with the initial placement stage of virtual machines. The problem of placing a VM on appropriate PM become more complex due to the various constraints such as cost [6, 14, 27], performance [14-15], availability [11, 15, 29], load balancing [7, 28], traffic pattern among VMs [20], scalability [28-29], energy [17, 19, 29] etc.

In [22], we have proposed a multi-objective approach for initial VM placement in cloud data center. In this approach we categorize all PMs in data center into four distinguished classes and suggested an autonomic and efficient technique for VM placement among physical servers. In continuation of this work, the performance of proposed approach is assessed through different metrics such as placement failure rate, SLA violations, number of active servers and energy consumption by data center. The standard existing algorithms for VM placement such as first-fit, next-fit and random selection are considered as base for the assessment. To evaluate the performance of our proposed algorithm, we have developed a simulation model which proves our algorithm is to be more effective than the existing standard algorithms. We use randomly generated dataset about available physical resources in data center and virtual machines for loading client applications.

The remainder of the paper is organized as follows. Section 2 reviews the related work. In Section 3, we introduce the problem formulation and discussed different related issues. An algorithm of multi-objective solution for the virtual machine consolidation is proposed in Section 4. Experiment results are shown in Section 5 and Section 6 concludes the work.

2. Related Work

Virtual Machine placement problem is extensively studied by many researchers and have adopted several different approaches. This problem is often formulated as a variant of the vector bin-packing problem.

Cost savings for better utilization of computing resources is the key factor considered by many researchers. Hyser *et al.* [14] presents a high level overview of VM placement and proposed a system architecture design of an autonomic VM placement to achieve cost savings for better utilization of computing resources. However, it lacks a resource management policy. Chaisiri *et al.* [6] proposed optimal virtual machine placement algorithm which can minimize the cost spending in reservation and on-demand plans for resource provisioning, whose goal is to minimize the number of used nodes. In [27] researchers designed an autonomic resource manager which predefines set of VM classes based on resource capacity. Each VM class comes with specific CPU and memory capacity. A VM must be chosen for the application among set of predefined VM classes

as per the current workload. The main drawback of this approach is that there may be over provisioned of resources allocated to application in specific VM.

The performance of data center can be improved by distributing the load of applications equally among all physical hosts. Hyser *et al.* [14] focused on building a framework with a load balancing policy, in which data center load is spread among all available physical hosts and resource usage is balanced as much as possible across all resource types. The initial VM placement problem for placing HPC applications is presented by Gupta *et al.* [12]. This research proposes a topology and hardware awareness techniques for optimizing the placement of VMs for HPC applications. However, as the resource requirement of application increases, run time placement with live migration should be taken into consideration; such an issue is not addressed in this research.

Availability of cloud applications is equivalently crucial as cost saving and performance. While scaling up and down resources for cloud applications, Wang et al. [29] has compared the influence of vertical and horizontal resizing techniques on resource management. Vertical scaling technique has the advantage of performance whereas horizontal scaling enhances the overall availability of applications. Availability of cloud applications can be maintained by keeping multiple copies of VMs on different PMs and distributing incoming requests among these copies [11]. It reduces the resource requirement of each copy and helps to utilize the server more efficiently. However, this research does not focus on the issue of maintaining consistency and cooperation between multiple copies of VMs. Jayasinghe *et al.* [15] proposed an algorithm that improves the availability of services in addition to the performance by deploying VMs across different isolation levels of the data center.

The VM placement can be optimized by considering the traffic pattern among them. Virtual machines with large mutual bandwidth usage are assigned to the same physical host or to host machines in close proximity [20]. However the traffic load among virtual machines is practically dynamic in nature and this research does not focus on the issue of determination of traffic pattern among virtual machines.

Energy saving is also one of the main issues in consolidation of virtual machines on physical servers in virtualized data centers. Cutting down energy consumption in data center is obviously reducing number of running physical servers [6, 17, 29]. Each VM require different dimensions of resources such as CPU, memory, storage and bandwidth. VM is hosted on a particular PM only when it satisfies its resource demand for all dimensions. Therefore some PM may have unutilized resources, referred to as resource fragments. X. Li *et al.* [17] proposed an algorithm EAGLE, which reduces number of PMs and sizes of resource fragments. However, applications are dynamic in nature, so their future demand of resources is not taken into account. An energy efficient approaches proposed in [16, 19] based on live VM migrations to reduce number of running PMs in data center. However, live VM migration requires transferring the working state and memory from one Physical Machine (PM) to another PM and thus consumes a large amount of I/O and network traffic [18], which causes significant impact on performance of applications [23, 30].

Existing research suggests mapping of VMs to a PM as per its current demand of the resources till resources are available on the PM. Whenever a running VM demands for additional resources and no more resources are free on the allotted PM, the VM is migrated to some other PM. The VM Migration can be either static or dynamic [30]. In static migration, VM is shutdown and only configuration file is sent from source to target server. Static migration takes negligible time for migration and also allows migration between different hypervisors more easily. In dynamic or live migration, VM working state and memory is sent from source to target server. Dynamic migration generates more network traffic and migration between different hypervisors cannot be easily possible, but there is no intervention of VM operation during resource reallocation process. Our proposed approach considers the future resource demand of applications and postpones

VM migration as long as possible which results in higher resource utilization, application performance and energy saving.

3. Problem Formulation

Consider the data center which owns number of heterogeneous servers and provides the service of hosting user applications on rent basis. Suppose that the cluster of servers is fully virtualized and when clients put forward their application tenants, VM is created as per the resource requirement and one of the PMs in data center with sufficient resources will be selected to place newly created VM. Each PM consists of different dimensions of resources such as CPU cores, memory, storage and network-bandwidth. In our problem, we consider only two dimensions of resources- CPU cores and Memory, to characterize a VM placement problem. We do not consider the disk size dimension, because Network Attached Storage (NAS) is used in data centers as a main storage across cluster of servers [10]. If m be number of VMs are running on same PM, then total CPU and memory utilization is estimated as the sum of CPU and memory usage of all those m number of VMs. Let n be the total number of PMs in data center and m be the number VMs to be placed at current time slot. Assume that, Tot_{pc} and Tot_{pm} are the total number of CPU cores and memory capacity of PM p respectively. Total CPU cores and memory allocated in PM p is denoted by $Alloc_{pc}$ and $Alloc_{pm}$ respectively. Req_{vc} and Req_{vm} represent the CPU cores and memory requirement by VM v. The available CPU and memory resources on PM p can be calculated as in below equations 1 and 2.

$$Avail_{pc} = Tot_{pc} - Alloc_{pc} \quad (1)$$

$$Avail_{pm} = Tot_{pm} - Alloc_{pm} \quad (2)$$

During runtime, if VM demands for more resources and associated PM be with a short of resources then existing research work states to migrate it on another PM having sufficient resources. To avoid such frequent live VM migrations, we must define some resource-cap say 70%, which is an upper bound on resource usage of PMs of data center. Dynamic resource requirement of VMs can be satisfied with available 30% free resources on PM with the help of Vertical Scaling technique implemented by eNlight cloud [9].

3.1. Acceptance State

Let Req_{vc} and Req_{vm} are CPU cores and memory requirement by VM v respectively. VM v can be hosted on PM p iff sufficient resources are available in p and after hosting v, total resource allocation of PM p should not go beyond its resource-cap. That means, equations 3 and 4 must be true before placement of VM v on PM p. Such a state is referred to *Acceptance State* for VM v on PM p.

$$(Alloc_{pc} + Req_{vc}) <= Tot_{pc} * Rcap_{c} \quad (3)$$

$$(Alloc_{pm} + Req_{vm}) <= Tot_{pm} * Rcap_{m} \quad (4)$$

Where, $Rcap_c$ and $Rcap_m$ are percentage of resource-caps on CPU and Memory usage for PM respectively, which are assumed to be 70%.

3.2. PM Startup Time

PMs that are idle (not allocated any resources) are kept at offline state for energy saving. The startup time of offline physical machines may delay for deployment of new virtual machines. So, some t number of idle PMs must always be kept at online state as buffer PMs to save the startup time, where t is some predefined threshold value.

Let Dtv be the time required for deployment of VM v.

$$Total\ deployment\ time = \sum_{v=1}^{m} Dtv \quad (5)$$

Where, m is the number of VMs to be deployed at current time slot. The objective is to minimize the total deployment time.

3.3. Placement Failure Rate

During deployment of new VM, appropriate PM is selected as per the VM placement algorithms; however, selected PM may not satisfy the resource demand of VM to be placed, which is referred as VM placement failure. So the algorithm must start re-searching of the next PM from the cluster. The objective is to minimize failure rate of VM placement in data center.

3.4. Power Consumption Model

Power consumption of a server is proportional to the utilization of CPU on the server. So, it can be expressed in terms of CPU utilization [19] as in equation 6.

$$P(U) = P_{idle} + U * (P_{max} - P_{idle}) \quad (6)$$

Where P_{max} is the power consumption by server, when CPU is fully utilized (U=1) and P_{idle} is power consumption by server, when it is idle (U=0).

$$Total\ power\ consumed = \sum_{i=1}^{n} Pi(U) \quad (7)$$

Where n be number of Physical Servers. The objective is to minimize the total power consumed by data center.

4. Proposed Approach

The proposed approach of VM placement in cloud data centre presented here addresses issues such as (1) maximum resource requirement during initial setup of VMs, (2) future demand of free resources by VMs at peak load, (3) avoiding live VM migration as long as possible, which affects on application performance (4) energy saving by keeping idle PMs at offline state. To the best of our understanding, existing research works related to this problem have not included these issues altogether.

In our approach, physical machines available in data center are categorized into four distinguished classes.

- **Offline class**: It contains PMs which are idle (not allocated any resources). These PMs are kept in offline state for energy saving.
- **Target class:** During initial placement of VM, it demands maximum resources for initialization, setup guest operating system as well as user applications with set of functionalities, therefore t>=1 number of PMs are kept online in this class. Initial deployment of every VM will be done in any appropriate PM of this class. Once VM is setup and initialized, it becomes ready VM and it is immediately migrated to appropriate PM of greedy class.
- **Greedy class:** To fulfill future demand of applications, only 70% of total resources on PMs are allocated. This class involves such PMs, whose resources are not allocated up to 70%. If any PM of this class has allocated about 70% of its resources then it is automatically migrated to the contented class.
- **Contented class:** This class contains PMs with resource allocation is almost 70%. Physical machines of this class are considered to be complete and no additional VM will be placed on them.

The proposed algorithm (see Algorithm 4.1) works as follows.

At the start up every VM v is to be placed and setup in any PM of the target class. It selects PM p from target class that satisfies Acceptance State (equations 3 and 4). If sufficient resources are not available on any target PM then algorithm searches for the

offline PM p which satisfies the Acceptance State and brings it to online, which is turn become as a target PM. VM v is then placed at PM p by allocating its resources and starts setup of operating system as well as user applications in it. When VM v becomes ready after its setup & initialization, it is migrated to an appropriate PM of greedy class. If any PM in greedy class does not satisfy the Acceptance State then PM p is shifted from target class to greedy class. If about 70% resources (resource-cap) of PM p in greedy class are allocated then that it is promoted to contented class. That means PM p is declared as full because 30% resources are kept for scaling up resources on the fly as per future demand of applications. Number of PMs in target class is always maintained to some threshold value t, to save start up time of offline PMs. So, if number of PMs in target class is below threshold value t then turn on one of the offline PM and promote it to target class.

Algorithm 4.1: Proposed Algorithm for Efficient Virtual Machine Placement

```
    //m is number of VMs to be placed at current time slot
1   For v=1 to m do
2   {
3    While VM v not placed do
4    {
5     Find PM p from Target_class that satisfies Acceptance State
6     If  no such PM found then
7     { Turn on next PM from Offline_class.
8       Continue While
9     }
10    Else //if PM p found
11      Place VM v at PM p; setup of OS, applications
      //When VM v is ready in Target_class migrate it into
      //PM of Greedy_class
12    If VM v is ready then
13    { Find PM q in Greedy_class satisfies Acceptance State
14      If no such PM found then
15      { Shift PM p from Target_class to Greedy_class
16        If number of PMs in Target_class< threshold t then
17          Turn on next PM and shift into Target_class
18      }
19      Else //PM q found in Greedy class
20       { Migrate VM v from PM p of Target_Class into PM q of Greedy_class.
21       }
22     }//If
23    } //while
      //If 70%(Rcap) resources of PM in Greedy class are full
      //then shift it into contented_class
24    If (Alloc_pc >= Tot_pc* Rcap_c OR Alloc_pm >= Tot_pm* Rcap_m )then
25      Shift PM p from Greedy_Class to Contented_Class
26  }//End for
```

5. Experiment Results

A proposed approach is based on Infrastructure as a Service (IaaS) model and it is essential to evaluate it on a virtualized data centre infrastructure. However, it is extremely difficult to conduct large scale experiments on a real infrastructure in cloud data centre. Therefore, simulation model has been developed in C and the performance is evaluated by comparing with standard algorithms - first fit, next fit and random selection. We have considered only two dimensions of resources- CPU cores and Memory to characterize a VM placement problem. A cloud data centre is simulated with 20 physical server machines and the number of CPU cores (in range 8 to 24) and memory capacity (in range 16 to 256 GB) of each server are randomly generated. In order to evaluate the

performance of our algorithm, load of 160 virtual machines in the range 1 to 8 VCPUs and 1 GB to 16 GB memory capacity is randomly considered for placement in data centre at a single time slot. In order to do a comparative study between standard and our proposed algorithm, parameters such as failure rate, number of active servers, application performance and power consumption are taken into consideration. Failure rate indicates the count regarding number of unsuccessful PM selections for consolidation of all VMs in current time slot. Power consumption is the total power consumed by PMs in data centre after consolidation of all VMs available in the current time slot.

Table 5.1 reports the experimental result obtained from different algorithms related to VM placement failure rate for four different random data sets and Figure 5.1 shows the comparative results between them. First-Fit algorithm selects first PM from cluster of PMs which satisfies the acceptance state. Next-Fit algorithm is similar to First-Fit with difference is that instead of always searching PM from the beginning of the list, it starts searching from the next PM after which the last search terminates. Random Selection algorithm selects PM randomly from the cluster of available PMs in data centre, which results in different failure rate on every instance of execution for the same data set. Random selection algorithm results in less failure rate as compare to our proposed algorithm; however, it consumes the significant power of data centre because all physical machines are kept in power on state.

Table 5.1. Failure Count for Different Algorithms

Algorithm	Failure Count			
	DataSet1	DataSet2	DataSet3	DataSet4
First Fit	3889	3884	4069	3912
Next Fit	1707	1825	1997	2100
Random	14	13	16	18
Proposed	87	91	98	102

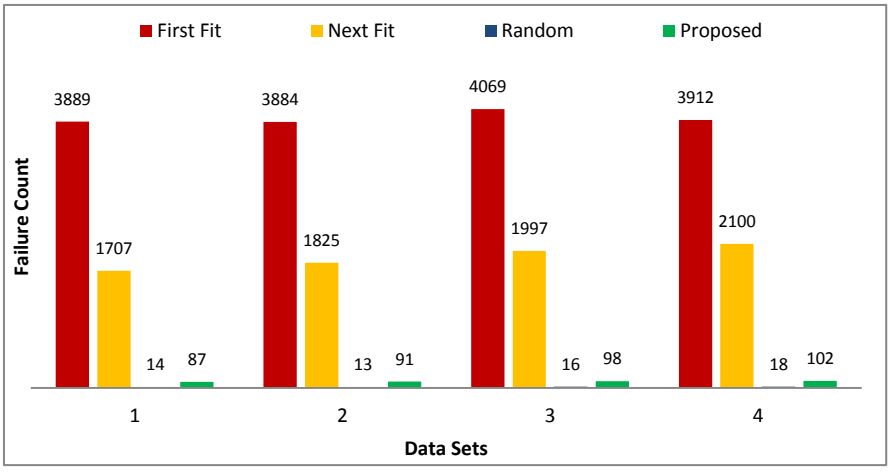

Figure 5.1. Comparative Result of Failure Counts

Table 5.2 shows the result of cumulative power consumed by Physical Machines during consolidation of every next 10 Virtual Machines as per the selection made by different algorithms and Figure 5.2 depicts their comparison result. The proposed technique saves 60 to 70% of energy consumption of data centre by keeping idle PMs at offline state as long as possible. As per the report of power monitoring system in eNlight cloud [9] data centre, the power consumption by a physical server is around 450 watt on

peak load, 120 watt at idle state and 5 watt at offline state, which is given as input to our simulation system. Proposed approach defines a resource-cap and reserves about 30% of resources in every physical server for dynamic vertical scaling as per the load, which postpones live virtual machine migration. Hence the performance of applications may be improved.

Table 5.2. Cumulative Power Consumption during VM Load

# of VMs Loaded	Power Consumption(Watt)			
	First Fit	Next Fit	Random selection	Proposed algorithm
10	4920	4800	4800	430.00
20	6120	4800	4920	890.00
30	7320	4800	5520	1028.45
40	8520	4800	6360	1373.45
50	9720	4920	7200	1488.45
60	10920	6120	7680	1718.45
70	12120	7320	8640	2063.45
80	13320	8520	9360	2408.45
90	14520	9720	10320	2877.32
100	15720	10920	11520	2992.32
110	16920	12120	12720	3357.91
120	18120	13320	13800	3591.05
130	19320	14520	14760	3821.05
140	20520	15720	15960	4166.05
150	21720	16800	17040	4539.10
160	22920	17520	18240	4769.10

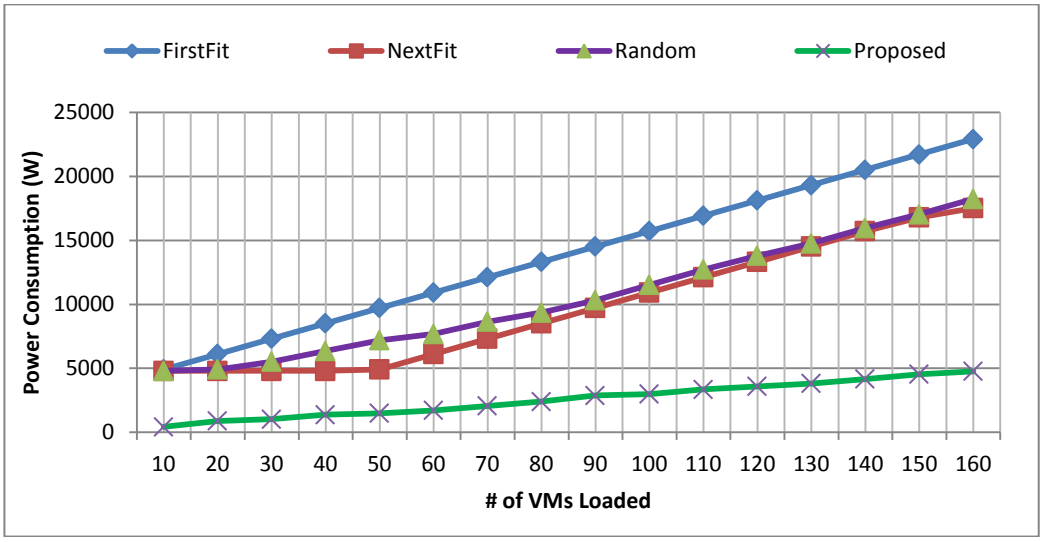

Figure 5.2. Comparative Results of Power Consumption by Different Algorithms

6. Conclusion and Future Work

In server virtualization, virtual machine placement in appropriate physical server is the key challenge for cloud service providers. This paper presents an autonomic solution for initial placement of VMs in cloud data centre which handles different issues such as maximum resource requirement during initial setup of VMs, dynamic resource scaling for VMs at peak load, improving the performance of applications and energy savings. The proposed technique is evaluated on parameters like failure rate of placing VM on server,

number of active servers, performance of applications and power consumption. Simulation model proved that, using proposed algorithm less number of failure occurs during VM placement as compare to the existing standard algorithms. The proposed technique saves 60 to 70% energy consumption of data centre by keeping idle PMs at offline state as long as possible. It also reserves about 30% CPU and memory resources in every PM for future resource scaling of applications. Obviously the dynamic live VM migration may be avoided, which improves the applications performance. This research work can be extended for the performance evaluation by dynamically increasing or decreasing the resource requirement of running applications. As a future work, in addition to CPU and memory, network bandwidth as another resource dimension can also be considered during VM placement.

References

[1] T. Alain, S. T. Giang, B. Laurent, D. Noel and H. Daniel, "Two Levels Autonomic Resource Management in Virtualized IaaS", Future Generation Computer Systems, Elsevier, vol. 29, no. 6, **(2013)**, pp. 1319–1332.
[2] Amazon Web Services, "Amazon Elastic Computer Cloud (EC2)", Retrieved on Feb 2014 from http://aws.amazon.com/ec2/, (**2014**).
[3] A. Beloglazov and R. Buyya, "Energy Efficient Resource Management in Virtualized Cloud Data Centers". In Proceedings of IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGRID '10), (**2010**), pp. 826-83.
[4] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility", Future Generation Computer Systems,Elsevier, vol. 25, no. 6, (**2009**), pp. 599-616.
[5] D. Carrera, M. Steinder, I. Whalley, J. Torres and E. Ayguade, "Utility-based placement of dynamic Web applications with fairness goals", In Proceedings of IEEE Conference on Network Operations and Management Symposium(NOMS), (**2008**), Salvador, Bahia.
[6] S. Chaisiri, B. Lee and D. Niyato, "Optimal Virtual Machine Placement across Multiple Cloud Providers", In Proceedings of IEEE Conference on Services Computing (APSCC), (**2009**), Singapore.
[7] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt and A. Warfield, "Live Migration of Virtual Machines", In Proceedings of ACM Second Conference Symposium on Networked Systems Design and Implementation, (**2005**), (NSDI '05), pp. 273-286. Berkeley, CA, USA.
[8] C. Dupont, G. Giuliani, F. Hermenier, T. Schulze and A. Somov, "An Energy Aware Framework for Virtual Machine Placement in Cloud Federated Data Centres", In proceedings of IEEE Third International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), (**2012**), Madrid, pp. 1-10.
[9] ESDS Solutions Private Limited, "eNlight Cloud Hosting", Retrieved on April 2014 from https://www.esds.co.in/enlight-cloud-hosting.php, (**2014**).
[10] Y. Gao, H. Guan, Z. Qi, Y. Hou and L. Liu, "A Multi-objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing", Journal of Computer and System Sciences, Elsevier, vol. 79, no. 8, (**2013**), pp.1230-1242.
[11] H. Goudarzi and M. Pedram, "Energy Efficient Virtual Machine Replication and Placement in Cloud Computing System", In Proceedings of 5th International Conference on Cloud Computing (CLOUD), (**2012**), Honolulu, HI, pp.750-757.
[12] A. Gupta, D. Milojicic and L. V. Kale, "Optimizing VM Placement for HPC in the Cloud", In Proceedings of ACM workshop on Cloud services, federation, and the 8th open cirrus summit, FederatedClouds'12, (**2012**), New York, NY, USA, pp. 1-6.
[13] S. He, L. Guo, Y. Guo, C. Wu, M. Ghanem and R. Han, "Elastic Application Container: A Lightweight Approach for Cloud Resource Provisioning", In proceedings of IEEE 26th International Conference on Advanced Information Networking and Applications (AINA), (**2012**), Fukuok, pp.15-20.
[14] C. Hyser, B. McKee, R. Gardner and B. J. Watson, "Autonomic Virtual Machine Placement in the Data Center", HP Laboratories, (**2008**).
[15] D. Jayasinghe, C. Pu, T. Eilam, M. Steinder, I. Whalley and E. Snible, "Improving Performance and Availability of Services Hosted on IaaS Clouds with Structural Constraint-aware Virtual Machine Placement", In Proceedings of IEEE International Conference on Services Computing, (**2011**), Washington, DC, pp. 72-79.
[16] B. Li, J. Li, J. Huai, T. Wo, Q. Li and L. Zhong, "EnaCloud: An Energy-saving Application Live Placement Approach for Cloud Computing Environments", In Proceedings of IEEE International Conference on Cloud Computing, (2009), Banglore, pp. 17-24.

[17] X. Li, Z. Qian, S. Lu and J. Wu, "Energy Efficient Virtual Machine Placement Algorithm with Balanced and Improved Resource Utilization in Data Center", Mathematical and Computer Modeling, Elsevier, vol. 58, nos. 5-6, (**2013**), pp. 1222-1235.
[18] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang and A. Ghalasi, "Cloud Computing - The Business Perspective", Decision Support Systems, Elsevier, vol. 51, no. 1, (**2011**), pp. 176-189.
[19] C. Mastroianni, M. Meo and G. Papuzzo, "Probabilistic Consolidation of Virtual Machines in Self Organizing Cloud Data Centers", IEEE Transactions on Cloud Computing, vol. 1, no. 2, (**2013**), pp. 215-228.
[20] X. Meng., V. Pappas and K. Zhang, "Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement", In Proceedings of IEEE Conference INFOCOM, (**2010**), San Diego, CA, pp. 1-9.
[21] Rackspace, Retrieved Feb 2014 from http://www.rackspace.com, (**2014**).
[22] M. Shelar, S. Sane, V. Kharat and R. Jadhav, "Virtual Machine Placement in Cloud Data Centers: A Survey and Proposed Approach", In Proceedings of Conference PMACC, vol. 13, (**2013**), pp. 22-28.
[23] Z. Shen, S. Subbiah, X. Gu and J. Wilkes, "CloudScale: Elastic Resource Scaling for Multi-Tenant Cloud Systems", In Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC'11), (**2011**).
[24] S. Shouraboura and P. Bleher, "Placement of Applications in Computing Clouds using Voronoi Diagrams. Journal of Internet Services and Applications", Springer, (**2011**), pp. 229-241.
[25] V. Shrivastava, P. Zerfos, L. Kang-Won, H. Jamjoom, Y. Liu and S. Banerjee, "Application-aware Virtual Machine Migration in Data Centers", In Proceedings of IEEE Conference INFOCOM, (**2011**), Shanghai.
[26] M. Stillwell, F. Vivien and H. Casanova, "Virtual Machine Resource Allocation for Service Hosting on Heterogeneous Distributed Platforms. In Proceedings of 26th International Conference on Parallel & Distributed Processing Symposium (IPDPS)", (**2012**), Shanghai, pp. 786-797.
[27] H. N. Van, F. D. Tran and J. Menaud, "Autonomic Virtual Resource Management for Service Hosting Platforms", In Proceedings of ACM ICSE workshop on Software Engineering Challenges of Cloud Computing (Cloud'09), (**2009**), Canada, pp. 1-8.
[28] L. M. Vaquero, L. Rodero-Merino and R. Buyya, "Dynamically Scaling Applications in the Cloud", ACM SIGCOMM Computer Communication Review,vol. 1, no. 2, (**2011**), pp. 45-52.
[29] W. Wang, H. Chen and X. Chen, "An Availability-aware Virtual Machine Placement Approach for Dynamic Scaling of Cloud Applications", In Proceedings of Ubiquitous Intelligence & Computing and 9th International IEEE Conference on Autonomic & Trusted Computing (UIC/ATC), (**2012**), Fukuoka, pp. 509-516.
[30] Z. Zhuang and C. Guo, "OCPA: An Algorithm for Fast and Effective Virtual Machine Placement and Assignment in Large Scale Cloud Environment", In Proceedings of International IEEE Conference on Cloud Computing and Big Data, (**2014**), Fuzhou, pp. 54-259.

Authors

Madhukar N Shelar, he received Master of Computer Science degree from Savitribai Phule Pune University, formerly known as University of Pune, India in 1993 and also qualified State Eligibility Test (S.E.T.) for Lectureship. He is Pursuing Ph.D. degree in Computer Science at Department of Computer Science, Savitribai Phule Pune University. Currently, he is working as a Head of Computer Science Department, KTHM College, Nashik. Madhukar has more than 20 years of teaching experience for undergraduate and postgraduate programmes in Computer Science and Information Technology and he has been member of Board of Studies in Computer Science and Faculty of Science at Savitribai Phule Pune University. He has authored five books includes System Programming and Operating System, OOP and Java Programming. He has successfully completed the research work funded by Savitribai Phule Pune University under minor research scheme and also published research papers at conferences. His research interest includes Cloud Computing, Operating System and Distributed Database Management Systems.

Shirish S Sane, he obtained his Diploma in Electronics and Radio Engineering (DERE) in 1984, Bachelors Degree in Computer Engineering from the Pune Institute of Computer Technology (PICT), Pune in the year 1987 and Masters Degree M. Tech in Computer Science & Engineering from Indian Institute of Technology (IIT), Mumbai in 1995. He is the first candidate being awarded the Ph. D. in Computer Engineering from Savitribati Phule Pune University, formerly known as University of Pune. Dr. Shirish is working as the Head of the Computer Engineering Department and Vice Principal at K K Wagh Institute of Engineering Education & Research, Nashik. He is a member of several boards of studies such as Computer Engineering and also member of Faculty of Engineering, Savitribai Phule Pune University, North Maharashtra University, Marathwada University. Currently, he is as Regional Vice President (RVP) for CSI Region VI(Maharashtra & Goa). He has published more than 35 research papers at the National and International Conferences and Journals. He has also authored books on the subjects "Data Structures" and "Theory of Computer Science". His areas of interests include Data Mining, Databases, Compilers and Cloud Computing.

Vilas S Kharat graduated with Physics, Chemistry and Mathematics, subsequently did his Master of Science from the Dr. B. A. Ambedkar Marathwada University, Aurangabad and Doctor of Philosophy from the University of Pune. During late Eighties he joined as a lecturer and in mid Nineties University of Pune, became full Professor in 2005. He is recipient of four consecutive best research papers awards by Indian Mathematical Society for the years 1998, 1999, 2000 and 2001. Professor Kharat has published number of research papers in different international journals. Five students have obtained their Ph.D. degrees and work of seven is in progress and importantly he has to his credit a partial solution to a very famous Frankl's Conjecture in collaboration with his research student. Kharat is also member of various learned societies including IEEE, American Mathematical Society member of various academic bodies which includes Board of studies, Research and Recognition Committee, Faculty of Science of various universities.

Rushikesh Jadhav, he is a Research Team Lead at ESDS Software Solution Pvt Ltd, Nasik. He has pioneered the cloud computing technology by giving ESDS its innovative Cloud Computing platform – eNlight Cloud. Rushikesh introduced himself in the cloud computing world immediately after his Graduation in Computer Engineering from K.K.Wagh Research and Engineering Institute of University of Pune. He deeply explored the Cloud technology, which remains his forte and he has been awarded the "CSI Young IT Professional" Award. With the right blend of expertise in software development, operating systems, storages and networking technologies, he innovated the eNlight Cloud Product. eNlight Cloud is an innovating computing approach in cloud technology built with the aim to introduce dynamism in servers to adopt varying computational needs and

increase the server uptime to 100%. eNlight Cloud product has also received “ET Telecom Awards 2014 – Innovation in Cloud Technology”. He and his group members have made key contribution in ESDS technologies for Compute, Storage and Networking. He is keen to gain a greater knowledge by pursuing his Masters in Computer Engineering from MET BKC Institute of Engineering of University of Pune. Rushikesh has also published patent with the Indian, US and UK patent offices around cloud computing.

