EFFICIENT ENERGY CONSUMPTION IN CLOUD LOAD BALANCING AND CONSOLIDATION USING MAX -MIN ANT COLONY OPTIMIZATION

Pankaj Singh Sisodiya

Department of Computer Science & Engineering,
Sagar Institute of Research & Technology, Bhopal, India
sisodiya.pankaj90@gmail.com

Dr. Vipin Tiwari

Department of Computer Science & Engineering,
Sagar Institute of Research & Technology, Bhopal, India
vipintiwari1@gmail.com

Abstract

Cloud computing provides multiple services in various data-centers where "on-demand" re- sources are provided to the users. With the high utilization standard of cloud data-centers, the amount of data flow in data-center get over larger. An efficient replacement of virtual machine (VM) among physical machines (PMs) improves resource utilization and efficiency. The various conventional load balancing methods are not performing up to the mark, and they are not deliberating the parameters of "service level agreement" (SLA) while deciding virtual machine migration. In this paper, we are presenting an efficient energy consumption technique in cloud load balancing and consolidation based on MAX − MIN Ant Colony Optimization (ACO) algorithm. Experimental results represents that, the proposed MAX − MIN (ACO) method performs better than static and dynamic virtual machine migration methods in context of energy consumption.

***Keywords*:** Cloud Computing, MAX − MIN ACO, Virtualization, VM Migration, Consolidation, Data- Centers.

1. Introduction

Cloud computing is a service which present the cooperation among cluster of physical machines and their services through the network and these powerful services are provided to end users [1]. The fundamental concept of cloud computing is combining distributed computing with grid computing. It has become truly significant to manage and control the data-centers as well as its resources due to high usage of large data information over the Internet [1][2].

The various virtualization methods have been proposed for suitable utilization of cloud resources. The huge number of virtual machines (VMs) are generated on the limited number of servers and each user works on their autonomous machine by using this technology. The data-centers are unequally loaded of tasks according to the request of end users. So there are the crucial issues such as load balancing, consolidation, power management, efficient resource utilization and service level agreement (SLA) which have to be controlled and managed [3][4].

The virtual machine (VM) migration supports management of the high load within cloud computing environment using live migrations. The VM migration is the process of moving the virtual machines from one physical machine into another within same data-centers or other and also balance the resource utilization among all the servers. The process of VM migration within cloud computing illustrated in Figure 1. The live migration is considered as the primary feature of VM migration, it essentially transfer the overall working state virtual machine from one server host into another. In cloud computing network, the live migration is essentially used for load balancing, consolidation, task scheduling, fault tolerance, maintenance of system, energy consumption

and green computing [5][6]. The methods are applied in live migration are pre-copy, post-copy and hybrid live VM migration [7].

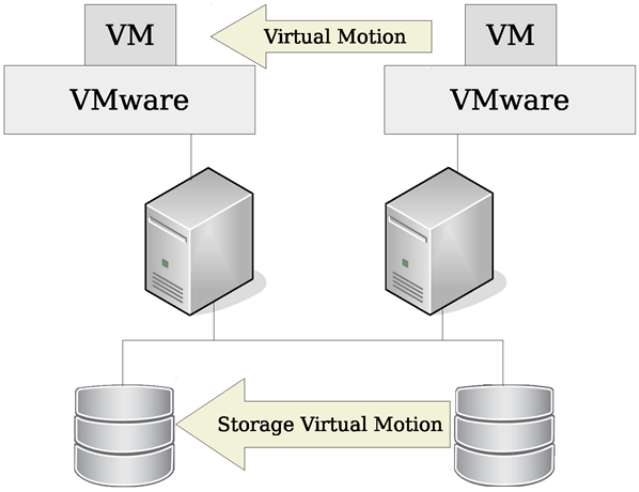

Figure 1: Virtual Machine Migration within Cloud Computing

The original motivation for deploying cloud system has been the dedication of minimum capital and operating costs, and the facilities of dynamically enhancing and deploying modern services without preserving a dedicated computation infrastructure [8]. Now, cloud computing has been set out to rapidly modify the organizations view for their growth and IT resources. The scenario of a single system having single operating system, application have already been moved into cloud computing, so plenty of resources are available to the user with a wide range selections [9].

Cloud computing is basically considered as a model for providing advantageous, request of "on-demand" access to a common set of configurable computational resources which are provisioned and serviced with minimal management exploitation. Here, the details of particular technology is completely hidden to the end-users while hosting and servicing their applications, as all the service is totally managed and controlled by the Cloud Service Provider (CSP) [10].

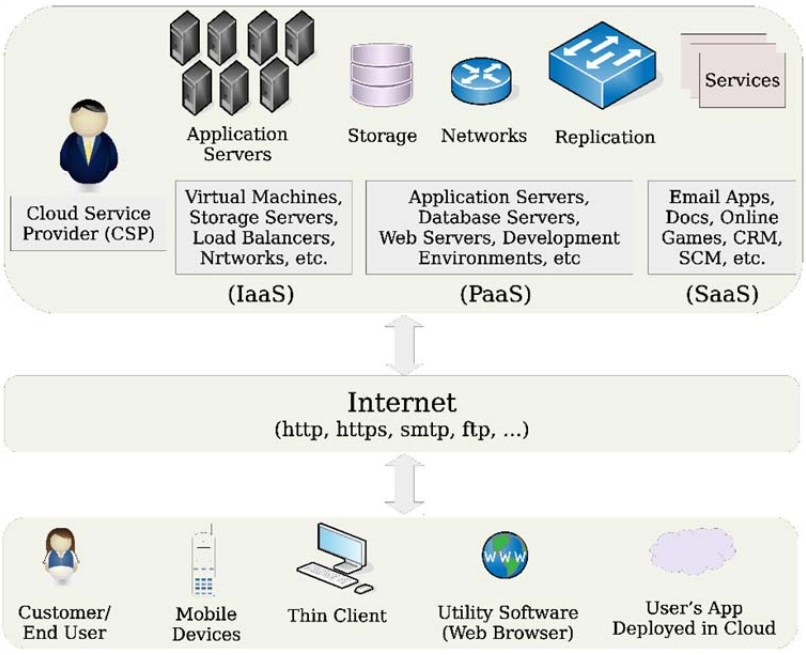

Figure 2: Architecture of Cloud Computing

The end-users consumes cloud services at the rate set by their specific requirements. This "on-demand" service is provided all time by CSP and all the essential complex activities and computations on behalf of the end-user is taken care [11]. For execution of end-user applications, it provides the complete environment which allocates the resource requirements and manages of the entire system work-flow. Figure 2, depicts the graphical representation of the sample architecture of cloud computing and its services [12]. Additionally, modern cloud model has assembled many proponents because its existence and exhibition as a "Greener Computing Alternative" [13]. The analytical aspects represents that the clustering of resources and facilities significantly reduces the costs for an organization. Moreover, as an AT&T supported analysis proposes that, it also has an highly positive effect on our environment. It is estimated that large organizations which use cloud computing can succeed for heavy annual energy savings and carbon reductions in large scale [14].

There has always been sources of state-of-mind between grid computing and cloud computing. The same visions are shared in clouds as well as grid such as: minimum computing cost, high flexibility and reliability. However, they generally differ in the following aspects:

Resource Sharing: The grid computing utilizes the sharing of resources through the organizations, whereas cloud computing provides "on-demand" resources based of the end-user. Because of isolation provided through the virtualization, so there is no true sharing [15].

Virtualization: Grids have potentiality to virtualize the integrated parts into a singular wide-area resource array. The virtualization generally comprehend both databases and files in their computing resources. Whereas, the cloud computing integrates virtualization of hardware also [16][17].

Coordination: To perform in grid computing system, the coordination of location and services work-flow are required, whereas in cloud it is not required [18].

Security: Cloud service end-users have unique access to their virtualized environment, as virtualization is highly related to security, whereas grid do not care about end user security [19].

Scalability: The scalability in grid is mainly enabled by maximizing the amount of operating nodes, whereas cloud automatically resizes the virtualized hardwares [20].

There are a wide scale of benefits for using the cloud computing technique which it dominates economically low cost services, remote accessibility and re-provisioning of resources. Cloud computing minimizes the value by canceling cost in dealing with physical infrastructure from the third party service provider.

In cloud computing, the resource managements and allocation dominates significantly crucial role in the performance of whole system with the degree of end-user satisfaction provided by the cloud system. But whereas providing the utmost consumer satisfaction the service provider wants to confirm the profits that incur to them jointly. The resource allocation ought to be economical on each read i.e. on the tip user and therefore the service provider perspective. Thus on get such a system the new technologies insist that the system ought to be with minimum SLA (Service Level Agreements) violation [3].

SLA: The service level agreement is a a part of the terms that's offered by the service giver to allow assurance to the end user concerning the extent of service that it will provide to the end user. In short, for a customer high QoS suggests few SLA violations. Virtualization is a popular answer that acts as a backbone for provisioning necessities of a cloud-based solution.

Virtualization: Virtualization is the use of hardware and software resources to form the perception that one or additional entities exist, although the entities, in actuality don't seem to be physically present. Using virtualization, we will create one server seem to be several, a desktop computer seem to be running multiple software system at the same time, many network association seem to exist, or huge amount of space or a vast variety of drives to be obtainable. The ability to form virtual machines (VMs) dynamically on demand may be a popular answer for managing resources on physical machines [16].

Virtualization usually provides a "virtualized" appearance of resources used to represent virtual machines (VMs). A hypervisor or A VM monitor manages, controls and multiplexes user access to the resources, maintaining the isolation among all VMs every time. As the cloud resources are virtualized, then several virtual machines can be executed on a physical machine (PM), each of which is self-contained within its own operation mode. The VM monitor, which intermediates the user access to the cloud resources and it can manipulate the degree of access to a resource like memory and CPU allotted to a VM.

2. Technical Aspects of Resource Management

The resource management [21] of cloud service provider work toward resource usage minimization and SLA adherence simultaneously, it is classified as follows:

Load Balancing: There are varied resource management policies for balance load in datacenter. The goal of load balancing is to avoid a state of affairs wherever there is an oversized discrepancy in resource utilization levels of the PMs [22]. A desired scenario might be to possess equal residual resource capability across PMs (to

facilitate increase native resource allocations throughout increase demands). Virtual machine migrations can be used to attain this balance [23]. Load balancing is of two types:

Static Load Balancing: In this method of load balancing, the static information of cloud system are taken into account for preference of node with the smallest amount loaded. In terms of complexness emergence, it performs better, but compromise with the degraded result as call is generated on statically collected knowledge.

Dynamic Load Balancing: In this method, current cloud system condition plays leading role while generating options. Neglecting the reality that dynamic load balancing contains greater run rime complexness than static load balancing method, dynamic has higher performance report as it deliberate current load of the cloud system for choosing upcoming data-center to serve the end-user request. This significantly provides the best choice from the available options for that system condition.

Power Saving: The utilization of the minimum power consumption at data-center is one of the prominent aspects of cloud resource management techniques. For achieving energy efficiency in cloud the following techniques are useful.

Server Consolidation: The main aim of server consolidation is to neglect low-resource-usage of the host. As illustrated in Figure 3, virtual machines (VMs) on lightly loaded server hosts are "packed" into fewer machines to fulfill resource requirements. After this swapping process, the freed-up physical machines (PMs) can be either switched off for energy saving or they are presented as higher-resource bins available for new VMs.

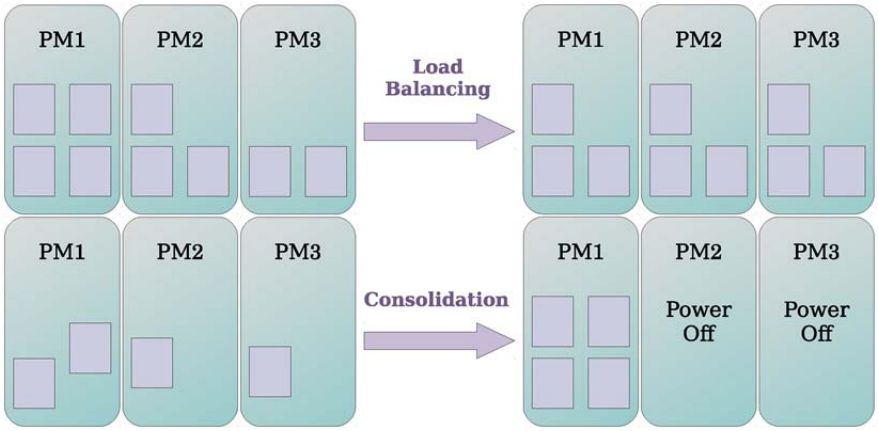

Figure 3: Load Balancing

In the first circumstance, the aim is either to distribute "load" equally across PMs or a VM requires many resources and so it is migrated to another machine. In consolidation process, machines are migrated to the fewer physical machines (PMs) to attain server position back. Many VMs can be supported by a single physical, allowing several applications which ordinarily require dedicated servers for a single physical server sharing [24]. This helps minimizing the number of physical servers within the knowledge center but simultaneously enhancing average server utilization from as low as 5 − 10% up to 60 − 70%. There are two varieties of server consolidation:

Static Consolidation: In static consolidation, the process is performed in a single step by using the high load demands of every usage to exploit virtual machine capacities, so the virtual machines (VMs) stay within the same servers throughout their life cycle. The high load utilization demand confirms that the virtual machine is not overloaded. However, idleness can be occurred since the workloads represent changeable demand patterns.

Dynamic consolidation: In dynamic consolidation, load reevaluates periodically in every virtual machine and acts the desired configuration alteration for workload and generally ends up with higher consolidation, so it dynamically changes the capacities of virtual machines according to the demanded workload. However, it requires migrating virtual machines among physical servers so as to avoid physical servers from an overloaded state. With addition of capacities of virtual machines mapped into a physical server get over higher than its original capability. A physical machine is switched off once the virtual machines are mapped, so they are often affected to other physical machines. Dynamic consolidation basically includes VM migration from one host into another.

3. Related Work

3.1 Ant System

Ant System is the original ACO algorithm presented by M. Dorigo *et al.* [25]. Its main feature is that, the pheromone amount and values are updated at each iteration by all the *m* ants which have developed a solution with the iteration itself. Pheromone $\tau_{i,j}$, related with the edges connecting cities *i* and *j*, are updated as follows:

$$\tau_{i,j} \leftarrow (1-\rho).\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i.j}^{k} \quad (1)$$

where m is the number of ants, ρ is the pheromone evaporation rate and $\Delta\tau_{i,j}^{k}$ is the amount of pheromone set on edge (i, j) by ant k:

$$\Delta\tau_{i,j}^{k} = \begin{cases} \frac{Q}{K_k} & if\ ant\ k\ used\ edge\ (i,j) in\ its\ hour \\ 0 & otherwise \end{cases} \quad (2)$$

where Q is a constant and L_k is the edge length of tour developed by ant k. In the development of a solution, ants choose the following city which is to be visited by a stochastic method. When ant k is in city i and has constructed a partial solution s^p, the probability of visiting to city j is given by:

$$P_{i,j}^{k} = \begin{cases} \frac{\tau_{i,j}^{\alpha}.\eta_{i,j}^{\beta}}{\sum_{c_{i,l} \in N(s^p)} \tau_{i,j}^{\alpha}.\eta_{i,j}^{\beta}} & if\ c_{i,l} \in N(s^p), \\ 0 & otherwise \end{cases} \quad (3)$$

where **N**(s^p) is a set of feasible components, which is, the edges (*i, l*) where *l* is a city has not visited yet by the ant *k*. The α and β are control parameters, the relative significance of pheromone versus heuristic information $\eta_{i,j}$ which is usually given by:

$$\eta_{i,j} = \frac{1}{d_{i,j}}, \quad (4)$$

where $d_{i,j}$ is the distance between city *i* and city *j*.

3.2 Multi-Objective Optimization Based on ACO in Cloud Computing

L. Zuo *et al.* [26] proposed a multi-objective optimization scheduling algorithm based on ant colony optimization (ACO) resource cost model in cloud computing. This algorithm considers the total length of the scheduling and the budget costs of use as constraints of this optimization problem, estimating multi-objective optimization for both cost and performance. The improved ant colony method is proposed to determine this problem solution.

The cost of the resource contains two parts of CPU and memory which is defined as

$$C_{cost(j)} = C_{base} \times C_j \times t_{ij} + C_{trans} \quad (5)$$

Here, C_{base} is base cost when the resources are used by the minimum CPU utilization. t_{ij} is the time duration to run the task T_i in the resource R_j . C_{Trans} is the CPU transmission cost.

The cost of memory is calculated as

$$M_{cost(j)} = M_{base} \times M_j \times t_{ij} + M_{Trans} \quad (6)$$

likewise, M_{base} is base cost for 1 GB memory. t_{ij} is the time duration to run the task T_i in resource R_j . M_{Trans} is the memory transmission cost.

The cost functions is obtained as based on the above CPU and memory cost models.

$$C(j) = \sum_{j-1}^{N} C_{cost}(j) \qquad (7)$$

$$M(j) = \sum_{j-1}^{N} M_{cost}(j) \qquad (8)$$

4. Proposed Approach

VM Migration is considered as NP−Hard problem and this problem can be solved in less time using some meta-heuristic algorithm. With the help of nature inspired Ant Colony Optimization (ACO) algorithm, the VM migration techniques can be simulated similarly. We consider each physical machine as represented by a node in graph and each edge defines similarity to VM migration from one physical machine (PM) to another. The generated graph is directed and also completely connected which have positive edge weights.

MAX − MIN Ant System (MMAS)

This method [27] is modification of the original ant colony system. Its feature elements are the only updates of the best ant pheromone trails which is bounded value of the pheromone. The pheromone update is defined as follows:

$$\tau_{i,j} \leftarrow \left[(i-\rho).\tau_{i,j} + \Delta\tau_{i,j}^{best}\right]_{\tau_{min}}^{\tau_{max}} \qquad (9)$$

where $\tau_{\max}$ and $\tau_{\min}$ are the upper and lower bounds respectively on the pheromone, the operator $[x]_b^a$ is estimated as:

$$[x]_b^a = \begin{cases} a & if\ x > a, \\ b & if\ x < b, \\ x & otherwise \end{cases} \qquad (10)$$

and $\boldsymbol{\Delta\tau_{i,j}^{best}}$ is:

$$\Delta\boldsymbol{\tau}_{i,j}^{\boldsymbol{best}} = \begin{cases} 1/L_{best} & if\,(i,j)belongs\ to\ the\ best\ tour, \\ 0 & otherwise, \end{cases} \qquad (11)$$

where L_{best} is the tour length by the best ant.

Dynamic Load Balancing
Dynamic Load Balancing
{
1. Set upper threshold (UT) to 75% and lower threshold (LT) to 25% as per standard.
2. Take lower window (*LW*) = 70 and upper window (*UW*) = 90.
3. Take delta = 2
4. Calculate average load (AvgLoad) of cloud data center if *AvgLoad* > *UT*
{
if *AvgLoad* + *delta* < *UW*
UT = *AvgLoad* + *delta*;
}
else if (*AvgLoad* < *UT*)
{
if (*AvgLoad* + *delta* > *LW*)
UT = *AvgLoad* + *delta*
}
}

In ACO, ants concurrently develop the solution for Cloud VM. Initially ants are usually put on randomly selected nodes which represent PM. At each iteration construction step, ant *k* applies probabilistic action choice rule, which is called random proportional rule, to decide to which PM given VM should be migrated. P_{ij} is the probability of migrating VM to *j* which is currently at *i* is:

$$P_{i,j}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\Sigma_{l \in N_i^k}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} \quad \text{if } j \in N_i^k \qquad (12)$$

Where $\eta ij = 1/dij$ is a heuristic. α and β are two parameters which determine the relative influence of the pheromone trail and the heuristic information, and where N_k is the nodes which are available. The Figure 4 represents the procedure of our proposed method. All implementations is effectively simulated using a tool called CloudSim within Eclipse IDE. The Table 1 represents the number of VM Migrations by existing and proposed method considering various values, the Table 2 represents the energy consumption by existing and proposed method considering various values and Figure 5 illustrates the graph as a result which compares the proposed method (MAX − MIN ACO) with static and dynamic methods which represents that the proposed method performs better as compared to existing methods.

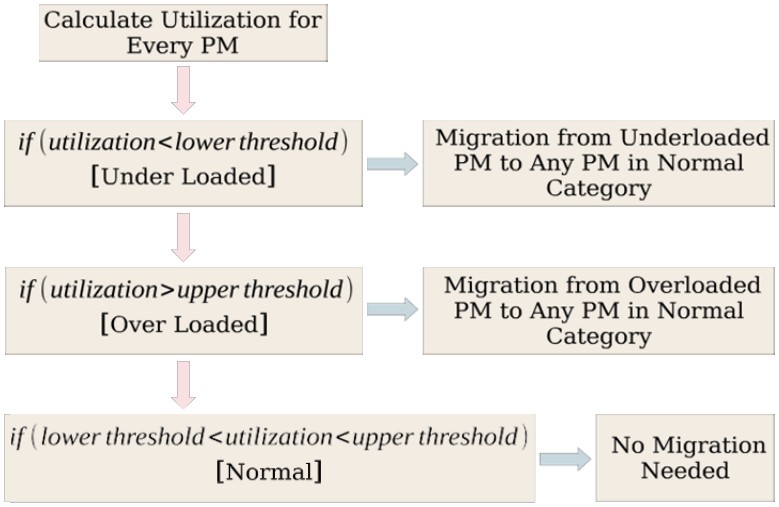

Figure 4: Proposed Method

Table 1: No. of VM Migrations by Existing and Proposed Method

No. of VM Migrations				
No. of PM	No. of VM	S-VMM	D-VMM	MAX − MIN ACO-VMM
10	15	10	5	2
15	20	9	6	3
20	25	11	8	4
25	30	15	9	5
30	35	11	7	5
35	40	13	8	6
45	50	18	13	8

Table 2: Energy Consumption by Existing and Proposed Method

Energy Consumption in *KWh*				
No. of PM	No. of VM	S-VMM	D-VMM	MAX − MIN ACO-VMM
10	15	4.14	3.30	1.14
15	20	6.44	4.60	1.43
20	25	8.18	5.94	2.02
25	30	9.04	7.05	2.21
30	35	10.91	7.88	2.57
35	40	12.81	8.93	3.22
45	50	14.91	11.74	3.46

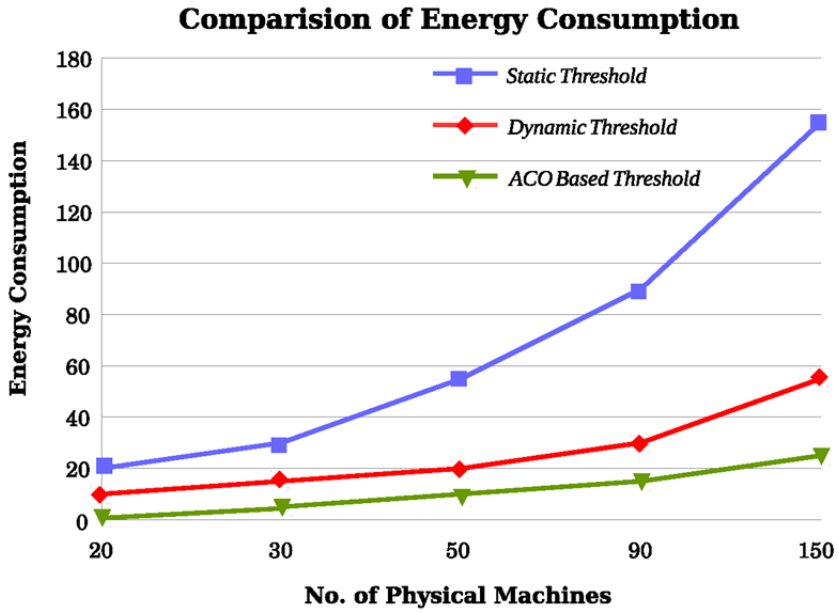

Figure 5: Result

5. Conclusion

In this paper, an MAX − MIN Ant Colony Optimization (ACO) approach is proposed for VM migration in the cloud system. As described in this paper, VM Migration is considered as NP-Hard problem and this problem is generally solved in less time using some meta-heuristic algorithm as Ant Colony Optimization (ACO). All such implementations is effectively simulated using a tool called CloudSim within Eclipse IDE. In this paper, MAX − MIN ant system is applied and performance of all these variants is compared with each other in terms of No. of VM Migrations, Energy consumption and VM consolidation. It is concluded that MAX − MIN ant system approach gives best results as compared to static and dynamic methods. In future, other nature inspired optimization methods and soft computing methods can be applied in cloud computing system.

References

[1] H. Jin, S. Ibrahim, T. Bell, L. Qi, H. Cao, S. Wu, and X. Shi, *Tools and Technologies for Building Clouds*, pp. 3–20. London: Springer London, 2010.
[2] R. Vasan, "A venture perspective on cloud computing," *Computer*, vol. 44, pp. 60–62, March 2011.
[3] S. Murugesan and I. Bojanova, *Cloud Service Level Agreement*, p. 744. Wiley-IEEE Press, 2016.
[4] A. Teshome, L. Rilling, and C. Morin, "Including security monitoring in cloud service level agreements," in *2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS)*, pp. 209–210, Sept 2016.
[5] E. Rodriguez, G. P. Alkmim, N. L. S. da Fonseca, and D. M. Batista, "Energy-aware mapping and live migration of virtual networks," *IEEE Systems Journal*, vol. 11, pp. 637–648, June 2017.
[6] T. Arthi and H. S. Hamead, "Energy aware cloud service provisioning approach for green computing environment," in *2013 International Conference on Energy Efficient Technologies for Sustainability*, pp. 139–144, April 2013.
[7] A. J. Younge, G. von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for cloud computing environments," in *International Conference on Green Computing*, pp. 357–364, Aug 2010.
[8] E. M. Guerra and E. Oliveira, *Metadata-Based Frameworks in the Context of Cloud Computing*, pp. 3–24. London: Springer London, 2013.

[9] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 931–945, June 2011.
[10] V. V. Rajendran and S. Swamynathan, "Parameters for comparing cloud service providers: A comprehensive analysis," in *2016 International Conference on Communication and Electronics Systems (ICCES)*, pp. 1–5, Oct 2016.
[11] N. R. Patil and R. Dharmik, "Secured cloud architecture for cloud service provider," in *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, pp. 1–4, Feb 2016.
[12] V. A. A. Quirita, G. A. O. P. da Costa, P. N. Happ, R. Q. Feitosa, R. d. S. Ferreira, D. A. B. Oliveira, and Plaza, "A new cloud computing architecture for the classification of remote sensing data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, pp. 409–416, Feb 2017.
[13] J. A. Pascual, T. Lorido-Botra´n, J. Miguel-Alonso, and J. A. Lozano, "Towards a greener cloud infrastructure management using optimized placement policies," *Journal of Grid Computing*, vol. 13, no. 3, pp. 375–389, 2015.
[14] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *Journal of Network and Computer Applications*, vol. 59, pp. 46 – 54, 2016.
[15] Z. Ali, R. U. Rasool, and P. Bloodsworth, "Social networking for sharing cloud resources," in *2012 Second International Conference on Cloud and Green Computing*, pp. 160–166, Nov 2012.
[16] U. Gurav and R. Shaikh, "Virtualization: A key feature of cloud computing," in *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*, ICWET '10, pp. 227–229, 2010.
[17] D. M. Leite, M. L. M. Peixoto, B. G. Batista, B. T. Kuehne, and C. H. G. Ferreira, "The influence of resource allocation on cloud computing performance," in *Proceedings of the Symposium on Applied Computing*, SAC '17, pp. 1516–1521, 2017.
[18] S. L. Bowman, C. Nowzari, and G. J. Pappas, "Coordination of multi-agent systems via asynchronous cloud communication," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 2215–2220, Dec 2016.
[19] H. Xiangyi, M. Zhanguo, and L. Yu, *The Research of the Cloud Security Architecture*, pp. 379–385. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
[20] M. F. Ali, O. A. Batarfi, and A. Bashar, "A simulation-based comparative study of cloud datacenter scalability, robustness and complexity," in *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 547–551, Dec 2015.
[21] D. Xu, X. Liu, and A. V. Vasilakos, "Traffic-aware resource provisioning for distributed clouds," *IEEE Cloud Computing*, vol. 2, pp. 30–39, Jan 2015.
[22] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 50 – 71, 2017.
[23] X. Liu, S. M. Yuan, G. H. Luo, H. Y. Huang, and P. Bellavista, "Cloud resource management with turnaround time driven auto-scaling," *IEEE Access*, vol. 5, pp. 9831–9841, 2017.
[24] B. Xu, Z. Peng, W. Ke, M. Zhong, and A. M. Gates, "Deployment method of VM cluster based on graph theory for cloud resource management," *IET Communications*, vol. 11, no. 5, pp. 622–627, 2017.
[25] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, pp. 29–41, Feb 1996.
[26] L. Zuo, L. Shu, S. Dong, C. Zhu, and T. Hara, "A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing," *IEEE Access*, vol. 3, pp. 2687–2699, 2015.
[27] T. Stu¨tzle and H. H. Hoos, "Max-min ant system," *Future Gener. Comput. Syst.*, vol. 16, pp. 889–914, June 2000.