Ant Colony Optimization Based Cloud Surrogation Placement Algorithm for Content Delivery Network

Komal Rane [1], Gagan Sharma [2], Rajesh Sharma [3]
[1,2,3]*Computer Science & Engineering Department,*
RKDF University, Bhopal, India

***Abstract*—The cloud computing is well known for its "on-demand" service models, supported by set of hardware, networking and software resources. Cloud computing services host huge virtual machines (VMs) for demanding situations as virtualization. Virtualization efficiently accomplish increasing demand for computing, storage and network resources in the large-scale cloud data centers. By virtual machine (VM) migration, modern situation come across the various resource management representation considering load balancing, proactive server maintenance, power management, distributive service availability and fault tolerance. In this paper, an efficient energy consumption technique in cloud load balancing and consolidation based on Ant Colony Optimization (ACO) algorithm has been presented for cloud surrogation placement in content delivery network. Experimental results represents that, the proposed method performs better than static and dynamic cloud surrogation methods.**

***Keywords*— Cloud Computing, $\mathcal{MAX} - \mathcal{MIN}$ ACO, Virtualization, VM Migration, Cloud Surrogation Placement, Consolidation, Content Delivery Network.**

I. Introduction

The cloud computing is computing technique in which large groups of remote servers are networked to allow centralized data storage and online access to computer services or resources. It provides flexibility and different computing platform for organizations. The "ShareLaTeX" is an great example of cloud computing, it is a free website that requires registration to access, where users can typeset LaTeX documents, save, sort, manage and generate pdf documents through a web browser without installing TexLive, LaTeX IDE and Linux operating system. It is used by approximately 8 million people per month and has a great storage capacity and computing environment. It is collaborated by top Universities [1].

Initially, due to absence of cloud computing, maintaining security of the information had been very difficult task. However, first appearance of the cloud computing has made life easier. The cloud computing consists of certain elements such as clients, servers and the main center where all servers are managed [2]. Figure 1.1 shows the architecture of cloud computing system, where the data owner stores the data and applications to the cloud storage. Whenever users need to access data and applications, the cloud service provider establish an access to the user through the cloud.

A. Types of Services

The cloud computing categories as follows [3].

- Software as a service (SaaS) applications are intended for end-users, delivered over the network.
- Platform as a service (PaaS) is the set of tools and services developed to perform coding and organizing the applications.
- Infrastructure as a service (IaaS) is the software and hardware that commands all servers, storage, networks and operating systems.

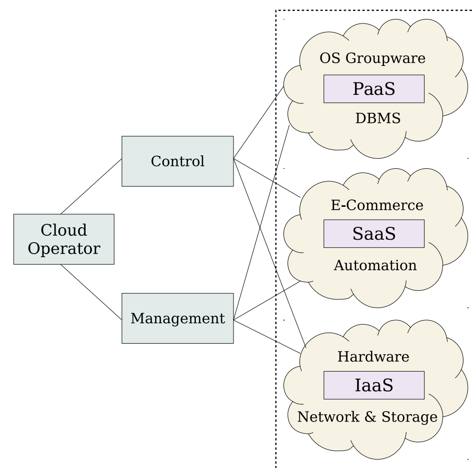

Fig. 1. Types of Services in Cloud Computing

The different types of services described in the Figure 1, description of each service in the cloud computing described in next section.

(*a*). *Software as a Service*

A SaaS [4] provider gives subscription option for an application, to the customers either as a service on demand, or at no charges. Some decidable features of SaaS include

- Web access to commercial software.
- Software is managed from a central location.
- Software delivered in a "one to many" model.
- Users not need to handle software upgrades and patches.

(*b*). *Platform as a Service*

PaaS [5] has the similar feature as SaaS which gives platform to run the applications, but in the software development area PaaS used as a computing platform. It grants the development of web applications fast and easily without any complexity of purchasing and maintaining the software and infrastructure for its organization. Characteristics of PaaS are as follows:

- PaaS provides services to develop, test, deploy, host and keep it in the similar unified evolution environment. It accomplished the application development process.

- Network base user interface, helps to create, modify, test and deploy different UI scenarios.
- It has sorted architecture where severe parallel users employ the same development application.
- Built in scalability of arranged software including load balancing and fall over.
- Incorporation of web services and databases via regular pattern.
- It adjusts for maturity group cooperation; some PaaS solutions contain project forecasting and communication tools.
- It also works as a tool that handles fee and subscription management.

(*c*). *Infrastructure as a Service*

IaaS [6] is a mode of carrying cloud computing infrastructure servers, storage, network and operating systems, as an on-demand service. Other than buying servers, software and datacenter space or network equipment, clients rather buy those resources as fully outsourced services on demand [7]. Features of IaaS described as,

- IaaS is generally established to observe the resources those are distributed as a service.
- It allows for dynamic leveling also has an erratic cost, utility pricing model.
- Generally IaaS includes various clients on a single piece of hardware.

II. Related Work

Wu *et al.* [8] presented an admission control and scheduling mechanism proposes which not only maximizes the resource utilization and profit, but also ensures that the QoS requirements is proposed. Mixed Workload Aware Policy (MWAP) is implemented to consider the workload of different types of application such transactional and non-interactive batch jobs. The proposed mechanism provides substantial improvement over static server consolidation and reduces SLA violations.

Ferreto *et al.* [9] proposed a new approach for Dynamic VM Consolidation with Migration Control. Authors have discussed that current techniques like static consolidation and dynamic consolidation does not consider the steady usage of virtual machines. Due to which problems may arise like migration cost and penalty to physical server. So, Authors claimed that for steady usage migration can be avoided but for variable usage migration can be performed. Authors had used Linear Programming Formulation and Heuristics approach for the same. Huang *et al.* [10] provided VM consolidation problem which is a NP Hard problem is solved by applying meta-heuristic algorithm ACO. The objective is to lower down the energy consumption of the overall algorithm. And the algorithm also reduces VM migrations.

Cao *et al.* [11] presented a novel allocation and selection policy for the dynamic virtual machine (VM) consolidation in virtualized data centers to reduce energy consumption and SLA violation. Firstly, it detect overloading hosts in virtual environments and then apply a method to select VMs from those overloading hosts for migration. VM Provisioning Method to Improve the Profit and SLA Violation of Cloud Service Providers. Li*et al.* [12] presented an approach for virtual machine consolidation based on energy efficient migration and live virtual machine migration. Authors have implemented the same approach using Eucalyptus which is an open source clone of the Amazon Elastic Compute Cloud (Amazon EC2).

Patel *et al.* [13] proposed an Threshold based algorithm for VM provisioning among multiple service providers that reduces SLA Violation. It uses two threshold values and two type of VMs (on-demand and reserved),These threshold values will be decided by the cloud federation depending on the environmental conditions like current workload, idle capacity of each cloud provider, etc. Aikema *et al.* [14] presented a power friendly algorithm is proposed. This paper compared live and non live VM migration in terms of power consumption.

Mills *et al.* [15] developed an objective method to facilitate the comparison of different virtual machine placement algorithms in the cloud. Xu *et al.* [16] proposed stable matching framework to decouple policies from mechanisms when mapping virtual machines to physical servers are presented and a general resource management architecture called Anchor is proposed. The resource allocation problem to be a convex optimization problem and a self-organizing cloud architecture is discussed. Speitkamp *et al.* [17] studied the static consolidation problem with a mathematical programming approach. They modelled the consolidation as a modified bin-packing problem. These works focus on the initial VM deployment or static consolidation problem based on resource utilization and do not consider VM migration overhead.

A. Ant System

Ant System is the original Ant Colony Optimization (ACO) algorithm presented by Dorigo *et al.* [18]. 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:

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

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:

$$p^k_{i,j} = \begin{cases} \dfrac{\tau^{\alpha}_{i,j}.\eta^{\beta}_{i,j}}{\sum_{c_{i,l}\in \mathbf{N}(s^p)} \tau^{\alpha}_{i,l}.\eta^{\beta}_{i,l}} & \text{if } c_{i,j} \in \mathbf{N}(s^p), \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $\mathbf{N}(s^p)$ 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.

III. Proposed Approach

VM Migration is considered as $\mathcal{NP}$−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.

A. $\mathcal{MAX} - \mathcal{MIN}$ Ant System (MMAS)

This method [19] 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:

$$\tau_{i,j} \leftarrow \left[(i-\rho).\tau_{i,j} + \Delta\tau^{\text{best}}_{i,j}\right]^{\tau_{\text{miax}}}_{\tau_{\text{min}}} \quad (5)$$

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

$$[x]^a_b = \begin{cases} a & \text{if } x > a, \\ b & \text{if } x < b, \\ x & \text{otherwise;} \end{cases} \quad (6)$$

and $\Delta\tau^{\text{best}}_{i,j}$ is:

$$\Delta\tau^{\text{best}}_{i,j} = \begin{cases} 1/L_{\text{best}} & \text{if } (i,j) \text{ belongs to the best tour,} \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

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

B. 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:

$$p^k_{ij} = \frac{[\tau_{ij}]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{l\in N^k_i} [\tau_{ij}]^{\alpha} [\eta_{ij}]^{\beta}} \quad \text{if } j \in N^k_i \quad (8)$$

Where $\eta_{ij} = 1/d_{ij}$ 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_l is the nodes which are available. The Figure 3 represents the procedure of our proposed method. All implementations is effectively simulated using a tool called CloudSim within Eclipse IDE. The

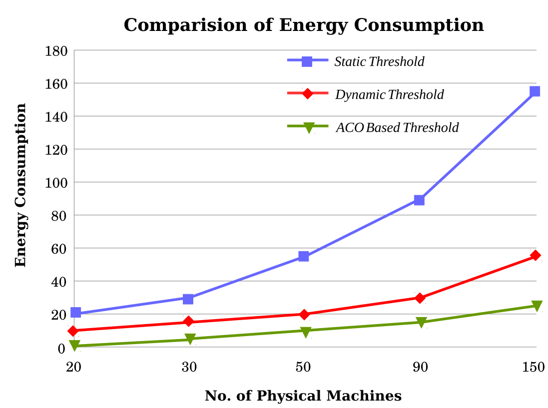

Fig. 2. Result

Table I represents the number of VM Migrations by existing and proposed method considering various values, the Table III represents the energy consumption by existing and proposed method considering various values and Figure 2 illustrates the the graph as a result which compares the proposed method ($\mathcal{MAX} - \mathcal{MIN}$ ACO) with static and dynamic methods which represents that the proposed method performs better as compared to existing methods.

IV. Result

In this research a cloud is simulated using Cloudsim having fixed number of physical machines and virtual machines.

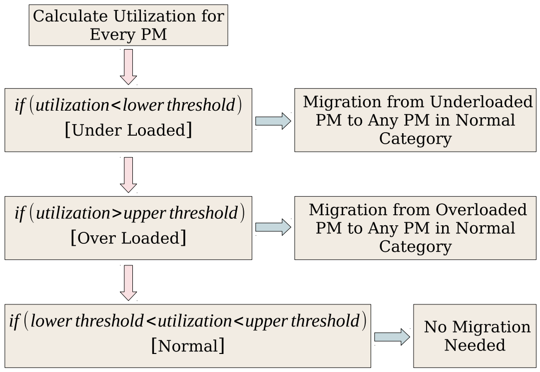

Fig. 3. Proposed Method

Configuration of physical and virtual machi.nes is measured in terms of MIPS (million instructions per second). VM Migrations, VM consolidation and energy consumption are recorded.

Physical machines are created based on the following MIPS list as shown below:

$$\{750, 1000, 1500, 2000\}$$

Virtual machines are created based on the following MIPS list as shown below:

$$\{250, 500, 750, 1000\}$$

No. of VM Migrations are compared for traditional VM Migration approach (TVMM), ACO based approach (ACOVMM) and Max-Min based approach (MMVMM). The above table clearly shows that number of VM migrations increase with the increase in number of VM. In traditional VM migration (TVMM) number of VM migration is more as compared to base approach ACO based VM migration. And proposed VM migration approach Max-Min based VM migration (MMVMM) offers least number of VM migrations. VM consolidation means number of PMs having less utilization switched to sleep mode by migrating all its VMs to other VMs having normal utilization. Number of VM consolidated should be more. And in proposed approach number of VM consolidated are more.

TABLE I
NUMBER OF VM MIGRATIONS FOR DIFFERENT APPROACHES

No. of VM Migrations				
No. of PM	**No. of VM**	**S-VMM**	**D-VMM**	$\mathcal{MAX}$-$\mathcal{MIN}$ 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 II
NUMBER OF VM CONSOLIDATIONS FOR DIFFERENT APPROACHES

No. of VM Consolidations				
No. of PMs	**No. of VMs**	**TVMM**	**ACOVMM**	$\mathcal{MAX}$-$\mathcal{MIN}$ ACO-VMM
10	15	2	5	7
15	20	3	3	6
20	25	5	7	10
25	30	8	11	14
30	35	11	13	18

TABLE III
ENERGY CONSUMPTION FOR DIFFERENT APPROACHES

Energy Consumption in KWh				
No. of PM	**No. of VM**	**S-VMM**	**D-VMM**	$\mathcal{MAX}$-$\mathcal{MIN}$ 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

Energy consumption should be as minimum as possible. And in proposed approach energy consumption is also minimum. Table I shows number of VM migrations in traditional approach, ACO based VM migration approach and Max-Min based approach which clearly shows Max-Min based approach gives the best results.

Similarly in Table II number of VM consolidations are compared for the three approaches and it is clear that VM consolidations in Max-Min are greater than other two. In Table III energy consumption is compared for all three approaches. Max-Min gives comparable results as ACO.

V. CONCLUSION AND FUTURE WORK

In this dissertation report, an $\mathcal{MAX} - \mathcal{MIN}$ 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 dissertation report, $\mathcal{MAX} - \mathcal{MIN}$ 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 $\mathcal{MAX} - \mathcal{MIN}$ ant system approach gives best results as compared to static and dynamic methods. The following works may be extended in future:

- The development of a new service which is “security as a service” to adapt to threat scenarios in both public cloud computing and virtualization.

- There is a need to design a tool that control and understand privacy leaks, perform authentication and guarantee availability in the face of cloud denial-of-service attacks.
- 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*. London: Springer London, 2010, pp. 3–20. [Online]. Available: http://dx.doi.org/10.1007/978-1-84996-241-4_1

[2] E. M. Guerra and E. Oliveira, *Metadata-Based Frameworks in the Context of Cloud Computing*. London: Springer London, 2013, pp. 3–24. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-5107-4_1

[3] 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)*, Oct 2016, pp. 1–5.

[4] R. Sharma and M. Sood, *Cloud SaaS: Models and Transformation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 305–314. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24055-3_31

[5] D. Beimborn, T. Miletzki, and S. Wenzel, "Platform as a service (paas)," *WIRTSCHAFTSINFORMATIK*, vol. 53, no. 6, pp. 371–375, 2011. [Online]. Available: http://dx.doi.org/10.1007/s11576-011-0294-y

[6] S. K. Panda and P. K. Jana, *An Efficient Resource Allocation Algorithm for IaaS Cloud*. Cham: Springer International Publishing, 2015, pp. 351–355. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-14977-6_37

[7] 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)*, Feb 2016, pp. 1–4.

[8] L. Wu, S. K. Garg, S. Versteeg, and R. Buyya, "Sla-based resource provisioning for hosted software-as-a-service applications in cloud computing environments," *IEEE Transactions on Services Computing*, vol. 7, no. 3, pp. 465–485, July 2014. [Online]. Available: https://doi.org/10.1109/TSC.2013.49

[9] T. C. Ferreto, M. A. S. Netto, R. N. Calheiros, and C. A. F. De Rose, "Server consolidation with migration control for virtualized data centers," *Future Gener. Comput. Syst.*, vol. 27, no. 8, pp. 1027–1034, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.future.2011.04.016

[10] Z. Huang and D. H. K. Tsang, "M-convex vm consolidation: Towards a better vm workload consolidation," *IEEE Transactions on Cloud Computing*, vol. 4, no. 4, pp. 415–428, Oct 2016. [Online]. Available: https://doi.org/10.1109/TCC.2014.2369423

[11] Z. Cao and S. Dong, "Dynamic vm consolidation for energy-aware and sla violation reduction in cloud computing," in *2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, Dec 2012, pp. 363–369. [Online]. Available: https://doi.org/10.1109/PDCAT.2012.68

[12] H. Li, G. Zhu, C. Cui, H. Tang, Y. Dou, and C. He, "Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing," *Computing*, vol. 98, no. 3, pp. 303–317, Mar. 2016. [Online]. Available: http://dx.doi.org/10.1007/s00607-015-0467-4

[13] K. S. Patel and A. K. Sarje, "Vm provisioning method to improve the profit and sla violation of cloud service providers," in *2012 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, Oct 2012, pp. 1–5. [Online]. Available: https://doi.org/10.1109/CCEM.2012.6354623

[14] D. Aikema, A. Mirtchovski, C. Kiddle, and R. Simmonds, "Green cloud vm migration: Power use analysis," in *2012 International Green Computing Conference (IGCC)*, June 2012, pp. 1–6. [Online]. Available: https://doi.org/10.1109/IGCC.2012.6322249

[15] K. Mills, J. Filliben, and C. Dabrowski, "Comparing vm-placement algorithms for on-demand clouds," in *Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science*, ser. CLOUDCOM '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 91–98. [Online]. Available: https://doi.org/10.1109/CloudCom.2011.22

[16] H. Xu and B. Li, "Anchor: A versatile and efficient framework for resource management in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1066–1076, June 2013. [Online]. Available: https://doi.org/10.1109/TPDS.2012.308

[17] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," *IEEE Transactions on Services Computing*, vol. 3, no. 4, pp. 266–278, Oct 2010. [Online]. Available: https://doi.org/10.1109/TSC.2010.25

[18] 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, no. 1, pp. 29–41, Feb 1996.

[19] T. Stützle and H. H. Hoos, "Max-min ant system," *Future Gener. Comput. Syst.*, vol. 16, no. 9, pp. 889–914, Jun. 2000. [Online]. Available: http://dl.acm.org/citation.cfm?id=348599.348603