

Task Scheduling Algorithm Based on Bacterial Foraging Optimization (BFO) in Cloud Computing

Anupama Gupta¹, Kulveer Kaur² and Rajvir Kaur³

¹HOD, ^{2&3}Student, Department of Computer Science,
Lala Lajpat Rai Institute of Engineering and Technology, Punjab, India
E-Mail: srankulveer@gmail.com

(Received 16 December 2017; Revised 31 December 2017; Accepted 22 January 2018; Available online 29 January 2018)

Abstract - Cloud computing is the architecture in which cloudlets are executed by the virtual machines. The most applicable virtual machines are selected on the basis of execution time and failure rate. Due to virtual machine overloading, the execution time and energy consumption is increased at steady rate. In this paper, BFO technique is applied in which weight of each virtual machine is calculated and the virtual machine which has the maximum weight is selected on which cloudlet will be migrated. The performance of proposed algorithm is tested by implementing it in CloudSim and analyzing it in terms of execution time, energy consumption

Keywords: VM migration, Cloudlet, virtual machines, CloudSim

I. INTRODUCTION

A large pool of systems which gather together in a public or private environment for providing resources in a shared manner is known as the cloud computing system. This provides a dynamic scalable infrastructure for the organization as well as the data stored. There is a reduction in the cost computation, application hosting, content storage and delivery with the help of cloud computing technique. There are direct cost advantages achieved due to the utilization of this technique [1]. Also, the data center is to be transformed from a capital-intensive set up to a variable cost environment.



Fig. 1 Cloud Computing Environment

The availability of cloud computing requires Software, hardware, application, platform, infrastructure and storage with an internet connection. There are certain steps involved within the cloud computing systems [2]. In grid computing

large scale problems are solved with the help of parallel computing. Utility computing provides the resources on the basis of metered services which means that the users will have to pay on the basis of amount they consumed the services. SAAS is a part of the service model of cloud computing. It provides the network based subscription to the applications. Cloud Computing involves the services from all of the above given environments. In other terms it provides IT resources anywhere and at anytime to the user [3].

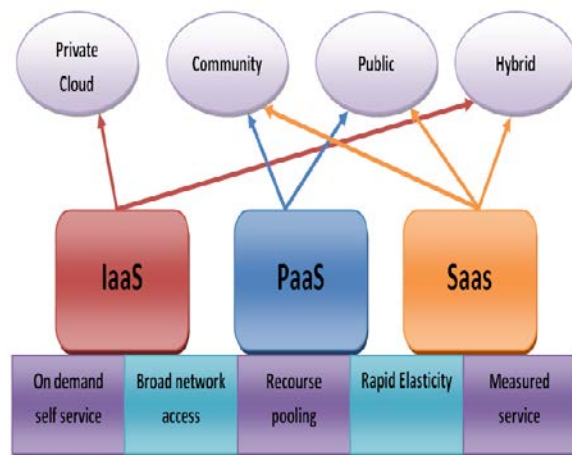


Fig. 2 Cloud Computing Service Models

Enterprises can choose to deploy applications on Public, Private or Hybrid clouds. Cloud Integrators can have a crucial impact on determining what the right cloud path for each organization is. Public clouds are owned and operated by outsiders. They convey better economies of scale than customers, as the infrastructure costs are spread among a mix of users, giving every individual customer an attractive minimal cost, "Pay-as-you-go" model. Private clouds are built exclusively for a single enterprise [4]. They aim to address concerns on data security and offer greater control, which is typically lacking in a public cloud. Hybrid Clouds join both public and private cloud models. With a Hybrid Cloud, service providers can use outsider Cloud Providers in a full or partial manner which helps in expanding the flexibility of computing. The Hybrid cloud environment is capable of giving on-demand externally provisioned scale. The ability to augment a private cloud with the resources of

a public cloud can be utilized to manage any unexpected surges in workload [5].

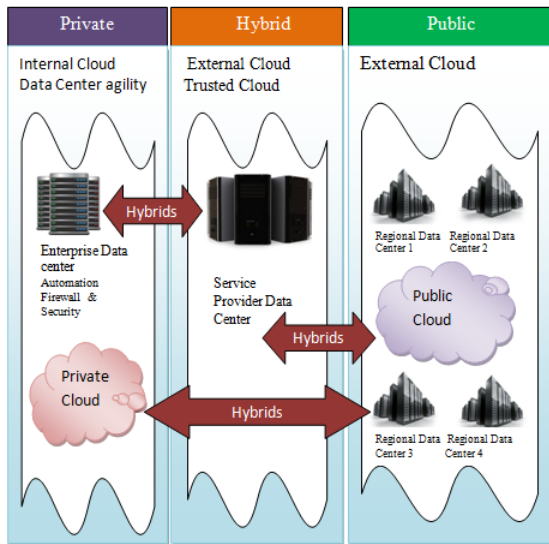


Fig. 3 Deployment Models

The cloud resources are to be allocated in a proper manner and this is a prior objective of this system. It is to be made sure that the financial profit is the highest in the selected architecture [6]. One of the major functions of cloud computing is the allocation of resources in which the resources are distributed in a proper manner. For the purpose of satisfying the requirements of users, resource allocation relates to providing commodities that are services. It is done keeping in mind the perspective of the consumer. The economy of the industries is increased by the proficient resource allocation results provided. The clients enter the resources at very less chances for a small job when the skill is to be deployed as a service. In certain systems, the resource allocation as well as the scheduling for providing proper coordination for the job resources is done by including a key [7]. There should be no violation of the set of constraints during this resource allocation process. There is no prediction of which resources are perfect for the allocation in a particular job provided by the cloud users. The perfect resource allocation procedure involves the proficient allocation along with the least number of resources to be used. This will help in providing maximum profit for the users. The main aim of resource allocation strategy is to provide resources from a team for the purpose of concurrent allocation in the cloud computing environment [8].

II. LITERATURE REVIEW

The authors introduced in this paper [9], an interval number theory to portray the uncertainty of the computing environment and a scheduling engineering to mitigate the impact of uncertainty on the task scheduling quality for a cloud data center. Based on this design, a novel scheduling algorithm (PRS1) is presented. The experimental results

demonstrate that PRS performs superior to those algorithms, and can effectively improve the performance of a cloud data center.

The authors proposed in this paper [10], an algorithm for allocating tasks to resources that minimizes the infringement of the tasks' time requirements. The exact time scheduling of the tasks on the resources is then decided by considering the time constraints. Experimental results demonstrate that the proposed algorithm outperforms other scheduling algorithms for various values of the granularity and the load of the task requests.

The authors discussed in this paper [11], open challenges connected with energy efficient resource allocation. Accessible techniques presented are summarized based on the energy-efficient research dimension taxonomy. The focal points and inconveniences of the existing techniques are comprehensively analyzed against the proposed research dimension taxonomy. On the basis of this study various techniques are enlisted which can be utilized as per their benefits.

The authors proposed in this paper [12], two energy-cognizant task consolidation heuristics, which aim to maximize resource utilization and explicitly consider both dynamic and mobility energy consumption. The heuristics assign every task to the resource on which the energy consumption for executing the task is explicitly or implicitly minimized without the performance degradation of that task. Based on the experimental results, the heuristics demonstrate their technique provides higher efficient energy-saving capability.

The authors proposed in this paper [13], a dynamically hierarchical, resource-allocation algorithm for multiple cloud nodes collaborating in a big data environment. Both theoretical and experimental results represent that the proposed algorithm outperforms the MinMin algorithm in terms of communication traffic and makespan. The results demonstrate that DHRA can reduce message number and communication traffic significantly, with the equal or even less tasks finishing time as compared with MinMin.

The authors proposed in this paper [14], a pragmatic approximated solution with the accompanying two steps. Every participant solves its optimal problem independently, without consideration of the multiplexing of resource assignments. It is demonstrated that Nash equilibrium dependably exists if the resource allocation game has feasible solutions. The advanced problem considered in this paper which is related to a large proportion of cloud-based computing services.

III. PROPOSED METHODOLOGY

In this research work, BFO algorithm is applied to resolve node failure in the cloud network. In the present algorithm there are a number of nodes available. From these

nodes, the candidate node will be chosen on the basis of failure rate and minimum execution time. Here Master node sets threshold value which includes two parameters amongst which one is failure rate and other is maximum execution time. The nodes which have equal to and less failure rate and minimum execution time are elected as candidate nodes by the master node. The node which has value equal to and less than this threshold value is selected as a candidate node. N1 has smaller value than threshold value so it will be a candidate node. N2 has one parameter less and other is higher so it will not be chosen as a candidate node. N3 has value equal to the threshold so, it will be selected as a candidate node 2. Again N4 has a greater value than threshold value and so it will not be selected as candidate node. After the selection, candidate node will start performing its tasks. We will also enter number of tasks in this scenario. Suppose, during execution of task one node moves from its location, and then failure occurs at that point. To overcome this problem a novel technique has been proposed which overcome the problem of failure due to mobility of the node. In the proposed algorithm, we have added a new parameter in the present algorithm that is master node time. Master node time is the result time to join the end users. It is for node collaboration. For this we have formulae which are as follows:

1. $E\text{-cost} = \text{maximum execution time} + \text{Time taken by the master node (master node time)}$. After that we will calculate profit of each node.
2. $\text{Profit of each node} = E\text{-cost} + \text{Failure node of each node}$
3. $\text{Weight of each node} = \frac{\text{No. of tasks} + \text{maximum execution time}}{\text{Profit}}$

The node which has the highest weight is chosen. Weight will be calculated according to the above mention formulae.

IV. PSEUDO CODE

Begin

Step 1: Get list of all VMs working on all hosts.

Step 2: Initialize no migration is done.

Step 3: Get resource utilization, failure rate, and execution time of all machines.

Step 4: Built transition matrix for hosts and VMs.

Step 5: Loop will execute until all machines on over utilized hosts are migrated.

Step 5.1: Calculate the current utilization of each host for that particular VM that needs migration.

Step 5.2: Check creation history of the VM.

Step 5.3: Compare increase in utilization of selected hosts with other hosts.

Step 5.4: Select host for which increase in utilization is minimum End loop

Step 5.6: If maximum utilization exceeds upper utilization threshold go to step 5.1.

Step 6: Else choose that particular host for migration.

Step 7: return migration List

END

V. RESULTS AND DISCUSSION

The proposed algorithm can be implemented in CloudSim as it is very difficult to make real time simulation as large scale. In the performance of existing technique and proposed algorithm is compared in terms of execution time, energy consumption. The simulation parameters are given in the table 1

TABLE 1 SIMULATION PARAMETERS

Number of VM	10
Number of cloudlets	60
Host Memory	2 GB
Processor	Xenon
Number of Data centers	5

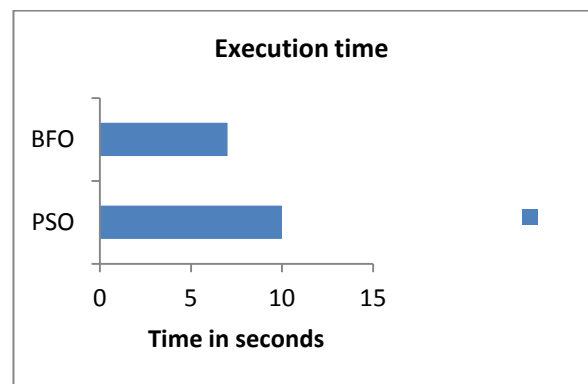


Fig. 4 Execution Comparison

As shown in the figure 4, the execution time of the proposed and existing algorithm is compared and it is been analyzed that due to fault recovery in the network, the execution time is reduced in the proposed technique.

$\text{Execution time} = \text{Time at the end of the algorithm} - \text{Time at the start of the algorithm} - (1)$

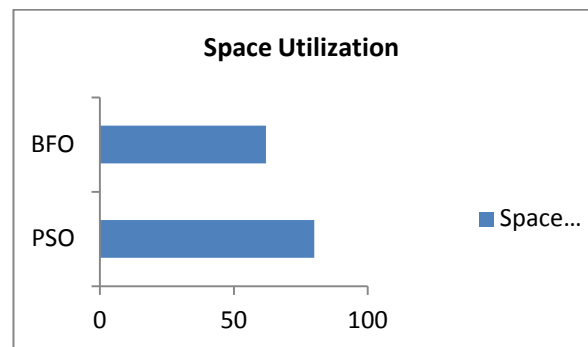


Fig 5 Space Utilization

As shown in the figure 5, the proposed and existing algorithms are compared in terms of space utilization and it is been analyzed that when the fault is recovery in the network, space utilization is reduced and network efficiency is increased

Space Utilization = Resource consumed per unit seconds * total number of seconds - (2)

The space utilization is calculated with the formula given in the equation 1, in which per unit bandwidth consumption is multiplied with the time

VI. CONCLUSION

It has been concluded that due to dynamic architecture of cloud, uncertainties may happen in the network. The most applicable virtual machines are selected on the basis of execution time and failure rate. The BFO algorithm is applied, which will migrate the task of virtual machine that gets overloaded. The performance of proposed algorithm is tested in CloudSim and it is analyzed that execution time is reduced with the energy consumption.

REFERENCES

- [1] Guiyi Wei, Athanasios V., Vasilakos, Yao Zheng, Naixue Xiong, "A game-theoretic method of fair resource allocation for cloud computing services", *J Supercomput*, Vol. 54, pp. 252–269, 2010
- [2] Young Choon Lee, Albert Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems", *J Supercomput*, Vol. 60, pp. 268–280, 2012.
- [3] Doulamis ND, Kokkinos P, Varvarigos E, "Resource selection for tasks with time requirements using spectral clustering", *IEEE Trans Comput*, Vol. 63, No. 2, pp. 461–474, 2014.
- [4] Abdul Hameed, Alireza Khoshkbarforoushha, Rajiv Ranjan, Prem Prakash Jayaraman, Joanna Kolodziej, Pavan Balaji, Sherali Zeadally, Qutaibah Marwan Malluhi, Nikos Tziritas, Abhinav Vishnu, Samee U. Khan, Albert Zomaya, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems", *Computing*, pp. 1–24, 2014.
- [5] Huangke Chen, Xiaomin Zhu, Hui Guo, Jianghan Zhu, Xiao Qin, Jianhong Wu, "Towards Energy-Efficient Scheduling for Real-Time Tasks under Uncertain Cloud Computing Environment", *J Syst Softw*, Vol. 99, pp. 20–35, 2015
- [6] Zhanjie Wang, Xianxian Su, "Dynamically hierarchical resource-allocation algorithm in cloud computing environment", *J Supercomput*, 2015.
- [7] Hamid Roomi Talkhaby, Reza Parsamehr, "Cloud Computing Authentication Using Biometric-Kerberos scheme based on Strong DiffieHellman-DSA Key Exchange", *International Conference on Control, Instrumentation, Communication and Computational Technologies*, vol.3, pp.104-110, 2016.
- [8] Krutika K. Shah, Vahida U. Vadiya, Rutvij H. Jhaveri, "A Survey Paper on Security in Cloud Computing: A Bibliographic Analysis", *Circulation in Computer Science*, vol.1, pp. 9-23, 2016.
- [9] Dr. S. S. Manikandasaran, "Security Attacks and Cryptography Solutions for Data Stored in Public Cloud Storage", *International Journal of Computer Science and Information Technology & Security*, vol.6, pp. 498-503, 2016.
- [10] Guodong Zhu, Yue Yin, Ruoyan Cai, Kang Li, "Detecting Virtualization Specific Vulnerabilities in Cloud Computing Environment", *2017 IEEE 10th International Conference on Cloud Computing*, Vol. 4, pp. 743-748, 2017.
- [11] M. Shamim Hossaina, Ghulam Muhammadb, Wadood Abdulc, Biao Songd, B. B. Gupta, "Cloud-assisted secure video transmission and sharing framework for smart cities", *ELSEVIER Future Generation Computer Systems*, Vol. 4, pp. 45-57, 2017.
- [12] Varun Mahajan, Sateesh K Peddoju, "Deployment of Intrusion Detection System in Cloud: A Performance-Based Study", Vol. 5, pp. 140-153, 2017.
- [13] Mohammad Taghi Adili, Amin Mohammadi, Mohammad Hossein Manshaei, and Mohammad Ashiqur Rahman, "Cost-Effective Security Management for Clouds: A Game-Theoretic Deception Mechanism", Vol. 5, pp. 98-106, 2017.
- [14] Tara Salman, Deval Bhamare, Aiman Erbad, Raj Jain, Mohammed Samaka, "Machine Learning for Anomaly Detection and Categorization in Multi-cloud Environments", *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing*, Vol. 3, pp. 97-103, 2017.