International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 3 Issue 5, August 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

# Fuzzy Based Algorithm for Cloud Resource Management and Task Scheduling

**Rashmi Singh Lodhi**[1], **Dr. R. K. Pateriya**[2]

[1]PhD Scholar, [2]Associate Professor
[1,2]Manit, Bhopal, Madhya Pradesh, India

***How to cite this paper***: Rashmi Singh Lodhi | Dr. R. K. Pateriya "Fuzzy Based Algorithm for Cloud Resource Management and Task Scheduling" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.1335-1340, **https://doi.org/10.31142/ijtsrd26604**

**IJTSRD26604**

**ABSTRACT**

This paper presents a Fuzzy Logic based approach to manage VM status and VM configuration within the cloud environment. The aim of the approach is to serve task requests efficiently with minimal use of resource and power. The proposed technique uses Fuzzy based approach to calculate the VM's status and VM's configurations, depending upon the cloud resource availability and job resource requirements, and then a fuzzy logic-based controller is used to control the status and configurations of the VM to serve the intended purpose afterwards. Controlling in this way reduces the active physical resources and the cloud's power requirements. The proposed controller is tested for different load conditions against the standard controlling algorithm to validate the concept. The test results obtained show that in terms of QoS, resource management, and power savings, the proposed fuzzy logic controller-based technique outperforms standard techniques.

***KEYWORDS:*** *Cloud Computing, Cloud Resource Management, Cloud VM Management, Fuzzy Logic Controller*

## 1. INTRODUCTION

In a cloud system the Virtual Machines (VMs) that are formed within the cloud system, are used to run the user services or applications. The VMs are formed by virtually allocating and configuring the physical resources. The virtualization platforms are used to manage these VMs. The cloud service providers need to maintain a minimum level of Quality of Service (QoS) bound by the Service Level Agreement (SLA) which is also known as the Service Level Objective (SLO).

Increasing application of cloud systems, imposing new challenges for cloud operators as they now have to deal with limited resources with greater loads of different operational and requirement features. To tackle such situations, an efficient cloud management algorithm is needed that could utilized cloud resources to meet the job requirements with guaranteed SLO. while keeping power requirements at the lowest possible level.

This paper presents a Fuzzy Logic based VM status and configuration management technique. The proposed technique uses task and cloud characteristics, then using this information to manage the VM's status and configuration through a fuzzy decision-making system based on logic. The proposed technique can manage the balance between conditions such as task requirements, SLO boundaries, sources available, power savings, etc. by dynamically configuring the VMs.

The rest of the paper is organized as follows. Section II provides a brief literature review. Section III provides an overview of Fuzzy's logic controller. Section IV explains the system architecture with simulation configurations, explains the proposed system in Section V. Section VI presents the results of the simulation and finally Section VII presents the conclusion and discusses the possibilities of future work.

## 2. LITERATURE REVIEW

The majority of proposed frameworks utilize resource controllers that are centralized in the sense that they have full control of the allocation of a given set of resources. A small number of frameworks incorporate decentralized controllers. Global scheduling involves a system-wide perspective on the allocation of the physical and virtualized resources that comprise a cloud environment. Recent years have seen a proliferation of proposals for frameworks and techniques for global scheduling. Most proposals originate from academic researchers; notable exceptions include the discussion by Gulati et al. [1] of the VMware Distributed Resource Scheduler and Distributed Power Management systems, which provide centralized control of a cluster of virtualized servers, performing VM placement, as well as load balancing and VM consolidation via live migration. Similarly, Wilkes [2] provides a high-level overview of two generations of Google's cluster management systems, focusing on design goals, management objectives and unsolved challenges.

Frameworks from the academic community that incorporate controllers directly or indirectly cooperating to achieve global scheduling include: vManage [3], which loosely couple's controllers for power and cooling management with controllers for virtualization management; the system proposed by Beloglazov and Buyya [4], which comprises controllers that cooperatively control VM placement and live migration; and TROPIC [5], which performs transactional cloud resource orchestrations that enforce safety, provide concurrency and increase system robustness. Whenever a request for the provisioning of one or more VMs is made by a Cloud User, the Cloud Provider's resource management system schedules the VMs by placing them onto PMs—the decision should help achieve the Cloud Provider's current management objective(s). Initial work on the VM placement problem typically assumed that VMs are assigned static shares of a PM's managed resources (typically CPU and/or memory). Given this, the placement of VMs onto PMs is related to the vector bin packing problem, which can be used to model static resource allocation problems where the resources used by each item are additive. To achieve server consolidation, for instance, the optimal placement is one where the items (i.e., VMs) are packed into a minimum number of bins (i.e., PMs), such that the vector sum of the items received by any bin does not exceed the bin's (resource) limit. The vector bin packing problem and its variants are NP-hard problems [6]; thus, heuristic algorithms are commonly proposed and evaluated. Most proposed heuristics are based on greedy algorithms using simple rules, such as First Fit Decreasing (FFD) and Best Fit Decreasing. Panigrahy et al. [7] study variants of the FFD algorithm and propose a new geometric heuristic algorithm which scales to large data centers without a significant decrease in performance. Wilcox et al. [8] propose a new generic algorithm, Reordering Grouping Genetic Algorithm (RGGA), which they apply to VM placement. Other works proposing bin packing heuristics for placement of VMs on PMs include Jung et al. [9], Gupta et al. [10], and Li et al. [140].

## 3. CLOUD COMPUTING

Cloud computing is a general term for the provision of hosted services over the Internet. These services are widely divided into three categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The name cloud computing was inspired by the cloud symbol which is often used in flowcharts and diagrams to represent the Internet.

A cloud service has three distinct features that distinguish it from traditional web hosting. It is sold on demand, typically by minute or hour; it is elastic— a user can have as much or as little service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and internet access). Increasing interest in cloud computing has been accelerated by significant innovations in virtualization and distributed computing as well as improved access to high-speed Internet.

### 3.1 Deployment Models

There are four different deployment models of cloud computing:

**Public Cloud:** Traditional, public or external cloud computing provides dynamic resources through an internet, VPN, or off-site third-party provider that accounts fine-grained. Public cloud: the external cloud or the public cloud.

**Community Cloud:** To benefit from cloud computing, if several organizations share the infrastructure, they can set up a community cloud. This is a more expensive option compared to the public cloud, as costs are spread over fewer users than public clouds. However, this option may provide increased privacy, security and/or policy compliance.

**Hybrid clouds:** Hybrid cloud means either two clouds (public, private, internal or external) or a combination of virtualized cloud server instances with real physical hardware. Using physical hardware and virtualized cloud server instances together, the most appropriate definition for "Hybrid Cloud" is likely to have one common service. Two clouds that were joined together call the "combined cloud."

**Private Clouds:** Private clouds describe private networking cloud services. It is made up of apps or virtual machines in the company's set of hosts. In the area of utility computing, they offer the advantages of shared hardware costs, the ability to restore fault and increase and decrease depending on demand.

## 4. FUZZY LOGIC SYSTEM

The term "fuzzy" refers to things that are either unclear or vague. Many times, in the real world we find a situation when we cannot determine whether the state is true or false, their fuzzy logic gives reasoning a very valuable flexibility. In this way, we can consider any situation's inaccuracies and uncertainties.

1 is the absolute value of truth in the Boolean system and 0 is the absolute false value. But there is no logic to absolute truth and absolute false value in the fuzzy system. But there is also an intermediate value in fuzzy logic that is partially true and partially false.

### 4.1 Fuzzy Architecture

Its Architecture contains four parts:

**Rule Base:** It contains the list of rules and the IF-THEN conditions on the basis of linguistic information provided by the experts to govern the decision-making system. Recent developments in fuzzy theory offer several effective methods for fuzzy controller design and tuning. Most of these developments reduce the number of rules.

**Fuzzification:** It is used to convert inputs to fuzzy sets, i.e. crisp numbers. Basically, crisp inputs are the exact inputs measured by sensors and transferred to the processing control system, such as temperature, pressure, rpm, etc.

**Inference Engine:** It determines the corresponding degree of the current fuzzy input relative to each rule and determines which rules are to be fired according to the field of input. Next, to form the control actions, the fired rules are combined.

**Defuzzification:** It is used to convert the fuzzy sets obtained by the inference engine to a crisp value. Several defuzzification methods are available and the best suited one is used to reduce the error with a specific expert system.

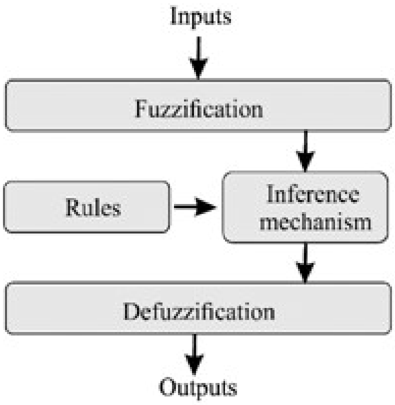

**Figure 1: Architecture of Fuzzy Logic Controller (FLC) system.**

### Membership Function

A graph that defines how to map each point in the input space to the membership value between 0 and 1. Input space is often referred to as the universe of discourse or universal set (u), which contains all possible elements of concern in each particular application.

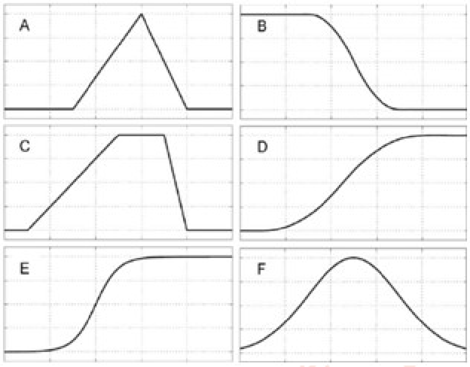

**Figure 2: Six types of fuzzy membership functions: triangular (A), z-shape (B), trapezoidal (C), s-shape (D), sigmoid (E) and Gaussian (F), (MATLAB user's guide, 2007).**

## 5. PROPOSED ALGORITHM

### 5.1. System Architecture

The proposed system architecture is presented in Figure 3. The system contains four Fuzzy Logic Decision Blocks, two VM Controlling Blocks and five Information Extraction Blocks. The working details of each block are as follows:

Task Analyzer: This block is used to extract useful information from task queue.

VM Analyzer: This block is used to extract useful information from VMs.

- ➢ Task Length: the length of current task in MIPS.
- ➢ Task Priority: Execution priority of current task.
- ➢ VM Utilization: how many times the particular VM has accessed during predefined time interval.
- ➢ VM Configuration: the resources used by VM.
- ➢ VM Load: current load on VM.

Fuzzy Logic Estimator and Controller Blocks: These blocks are used to make specific decisions based on the inputs provided using fuzzy logic.

- ➢ Fuzzy Task Weight Score Estimator: This block estimates the task requirements based on the length and priority of the task.
- ➢ Fuzzy VM Capability Estimator: This block estimates the ability of VM to handle basic VM configuration and load tasks.
- ➢ Fuzzy Task-VM Compatibility Estimator: This block estimates fitness between VM capabilities and task requirements.
- ➢ Fuzzy VM Status Controller: This block is used to determine the operational status of VM based on the resource used by VM and the VM access rate.

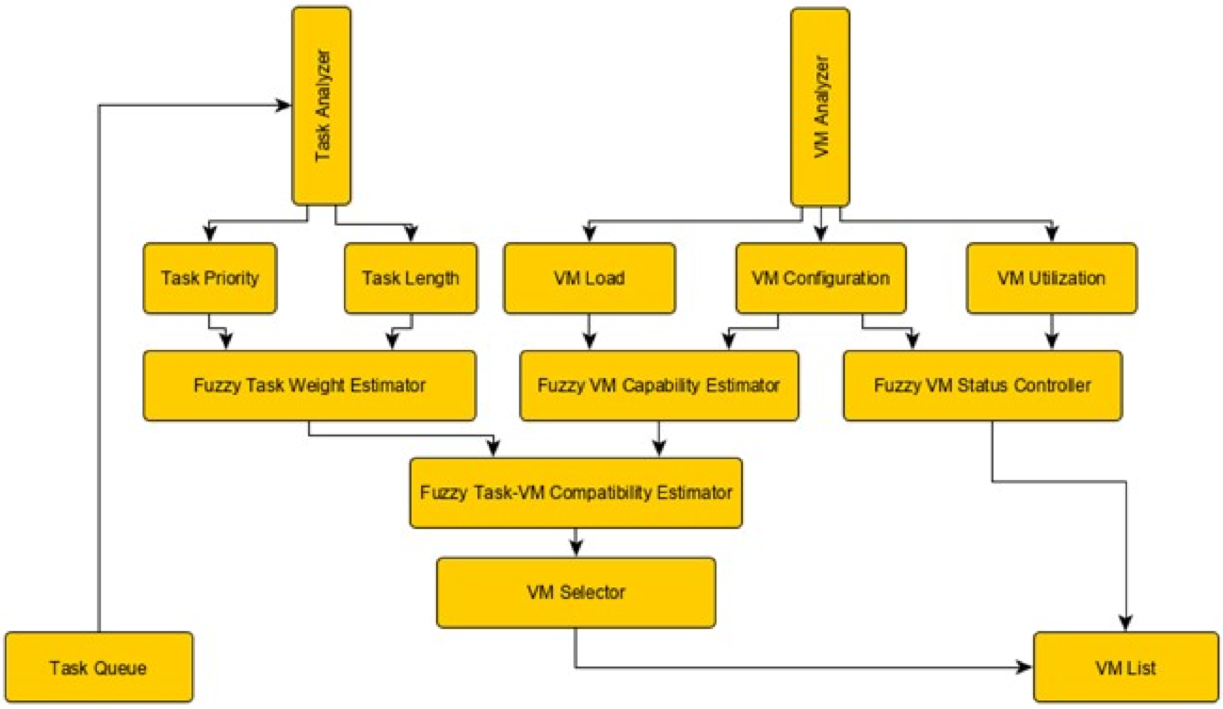

**Figure 3: Proposed System Architecture**

**VM Selector:** This block is used to select the best VM for the current task; it takes the Fuzzy Relative Score Estimator input for all VMs and selects the VM with the highest relative score.

**Fuzzy VM status Controller:** it keeps the states of all VM on the basis of VM status score ($VM_{score}$). This blocks uses two different thresholds $t_{sleep}$ and $t_{dissolve}$ where $0 < t_{sleep} < t_{dissolve} < 1$ which are compared against VM status score to decide the VM status as follows:

$$VM_{status} = \begin{cases} Keep\ Running, & if\ VM_{score} < t_{sleep} \\ Sleep, & elseif\ t_{sleep} \geq VM_{score} > t_{dissolve} \\ Dissolve, & else\ VM_{score} \geq t_{dissolve} \end{cases}$$

## SIMULATION RESULTS

Following consideration is taken into account when the numerical models are developed.

- It is assumed that the load balancer knows the configuration (such as processing capacity, memory, etc.) of each virtual machine (VM) in the cloud.
- With zero-time delay, the load balancer can get the operational status of each VM.
- The load balancer takes no time to select and assign the tasks to VM's.
- The load balancer selects the VM based on the selected algorithm for the input tasks.
- Each VM has zero boot time, so start executing the assigned task immediately.
- The incoming task size is considered in MI (million instructions) units.
- The capacities of the VM are also considered in MIPS units (million instructions per second).

The evaluation of the proposed algorithm is carried out using numerical computing software from MATLAB. During the simulation, the tasks arrive at a rate of π as a Poisson process. The random length tasks are generated using a uniform discrete distribution within the provided minimum and maximum task length limits. The similar way is used to generate task priorities and to define the execution capabilities of VM.

### 5.2. Definition of Evaluation Terms

The following measures are used to evaluate the performance of the algorithm.

**SLO Failure:** is defined as a cloud failure to serve the task within a given time limit (inverse of priority).

**SLO Failure Task Length:** defines the duration of the SLA Failure task.

**VM Reboots:** is booting VMs from sleep mode, this operation is required when the already running VMs cannot serve the current task.

**VM Reforms:** is the formation of new VM forms the available unused resources when the current task cannot be handled by the already formed (running or sleeping) VMs.

**Resource Utilization Efficiency:** shows how efficiently cloud resources are used to serve tasks and is calculated as follows:

$$Resource\ Utilization\ Efficienc = \frac{\sum_{i=1}^{N} TL_i}{\sum_{i=1}^{N}\left(\sum_{j=1}^{A_i} C_{VM}^{j}\right)} \times 100$$

Where

$TL_i$: is the load in cloud at time $i$.

$A_i$: is the number of VMs active and running at time $i$.

$C_{VM}^{j}$: Execution capacity of the $j^{th}$ VM.

$N$: is the total simulation time (discrete events of task arrival).

### 5.3. Simulation Configurations

To properly simulate the algorithm, some important parameters are required to configure these parameters and their values are listed in table 2.

Table2: The simulation parameters and their values.

| Configuration Parameter | Parameter Value |
|---|---|
| Total Execution Capacity Available | 100 MIPS |
| Sampling Window Length | 10 |
| Minimum Task Length | 100 MI |
| Minimum Task Execution Time | 1 Seconds |
| Maximum Task Execution Time | 10 Seconds |
| Threshold Sleep | 0.5 |
| Threshold Dissolve | 0.7 |
| Total Simulation Time | 100 Seconds |

### Simulation Results

The simulation results are presented in graphical forms. The results of the proposed algorithm are also compared with the two standard task scheduling algorithms called Round Robin and Random Selection.

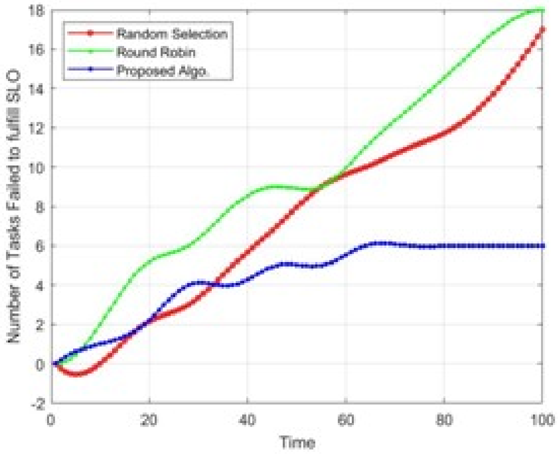

**Figure 4: Plot for number of tasks failed to receive the requested SLA by cloud due shortage of resources with respect to simulation time.**

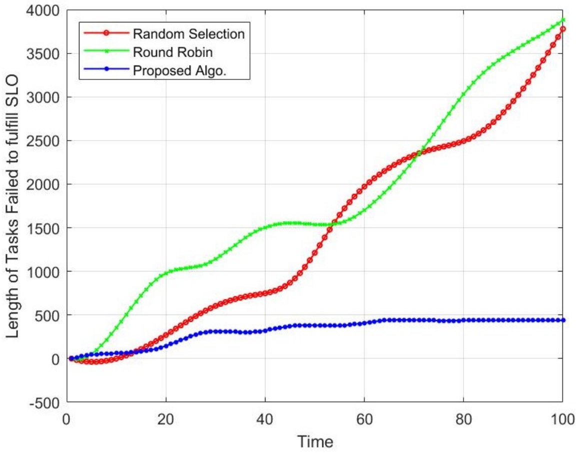

**Figure 5: Plot for total length of the tasks which failed to receive the requested SLA by cloud due shortage of resources with respect to simulation time.**

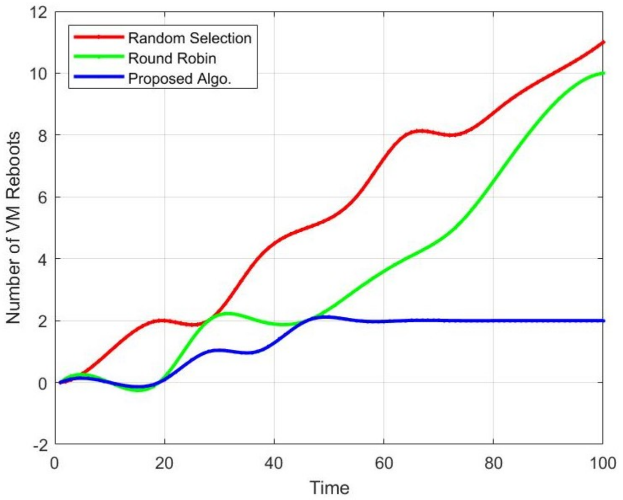

Figure 6: Plot for number of times the VM is rebooted from the sleep mode for assignment of task with respect to simulation time.

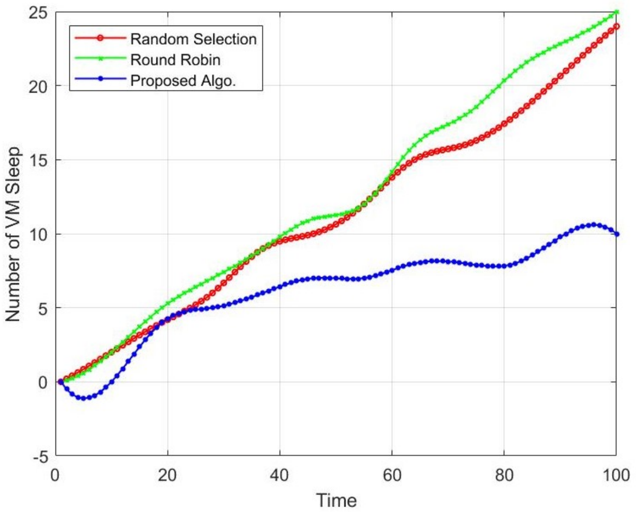

Figure 7: Plot for number of times the VM is reformed from the available resources for assignment of task with respect to simulation time.

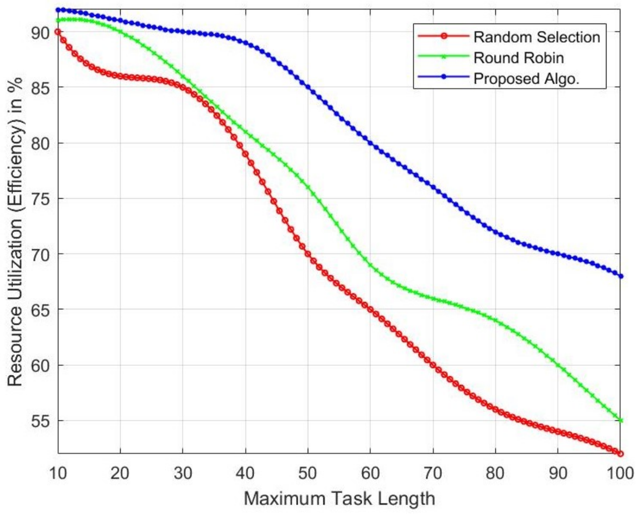

Figure 8: Plot showing the variation of the cloud resource utilization efficiency with respect to maximum task length.

## 6. CONCLUSION

We presented the fuzzy logic-based cloud systems task scheduling and resource management schemes in this paper. The simulations show that the proposed algorithm reduces the number of tasks failed to achieve SLO and the total length of the task failed to achieve SLO compared to standard algorithms by a factor of 3.0 and 8.0 (Figure 4 and 5) respectively. This shows that the algorithm rejects tasks with the lowest length when SLO is not delivered guaranteed.

A number of VM reboots and reforms reduced by a factor of 5.0 and 2.5 (figure. 6 and 7) in the proposed algorithm.

Finally, the efficiency comparison (fig. 8) shows that the proposed algorithm provides maximum efficiency. It also shows that the efficiency falls much slowly than the other algorithm, so it can be said that the algorithm provides a lot of uniform and stable performance for a wide range of loading conditions. These results show that the proposed algorithm adequately handles a cloud system for task scheduling and resource management with limited resources and SLO boundaries.

## References

[1] Gulati, A., Holler, A., Ji, M., Shanmuganathan, G., Waldspurger, C., Zhu, X.: VMware distributed resource management: design, implementation, and lessons learned. VMware Tech. J. 1(1), 45–64 (2012).

[2] John Wilkes, 2011 GAFS Omega. http://youtu.be/0ZFMlO98Jkc (2011).

[3] Kumar, S., Talwar, V., Kumar, V., Ranganathan, P., Schwan, K.: Loosely coupled coordinated management in virtualized data centers. Clust. Comput. 14(3), 259–274 (2011).

[4] Beloglazov, A., Buyya, R.: Energy efficient resource management in virtualized cloud data centers. In: Proceedings of 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid 2010), pp. 826–831. IEEE (2010).

[5] Liu, C., Mao, Y., Chen, X., Fernandez, M.F., Loo, B.T., van der Merwe, J.: TROPIC: transactional resource orchestration platform in the cloud. In: Proceedings of 2012 USENIX Annual Technical Conference (ATC 2012). USENIX (2012).

[6] Coffman Jr, E.G., Garey, M.R., Johnson, D.S.: Approximation Algorithms for NP-Hard Problems, pp. 46–93. PWS Publishing Co., Boston (1997).

[7] Panigrahy, R., Talwar, K., Uyeda, L., Wieder, U.: Heuristics for vector bin packing. Technical report, Microsoft Research (2011).

[8] Wilcox, D., McNabb, A., Seppi, K.: Solving virtual machine packing with a reordering grouping genetic algorithm. In: Proceedings of 2011 IEEE Congress on Evolutionary Computation (CEC), pp. 362–369. IEEE (2011).

[9] Jung, G., Joshi, K., Hiltunen, M., Schlichting, R., Pu, C.: Generating adaptation policies for multitier applications in consolidated server environments. In: Proceedings of 2008 International Conference on Autonomic Computing (ICAC 2008), pp. 23–32. IEEE (2008).

[10] Gupta, R., Bose, S., Sundarrajan, S., Chebiyam, M., Chakrabarti, A.: A two stage heuristic algorithm for solving the server consolidation problem with item-item and bin-item incompatibility constraints. In:

Proceedings of 2008 IEEE International Conference on Services Computing (SCC 2008), vol. 2, pp. 39–46. IEEE (2008).

[11] Li, B., Li, J., Huai, J., Wo, T., Li, Q., Zhong, L.: EnaCloud: an energy-saving application live placement approach for cloud computing environments. In: Proceedings of 2009 IEEE International Conference on Cloud Computing (CLOUD 2009), pp. 17–24. IEEE (2009).

[12] M. S. Q. Z. Nine, M. A. K. Azad, S. Abdullah, and R. M. Rahman, “Fuzzy logic based dynamic load balancing in virtualized data centers,” 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2013.

[13] A. N. Toosi and R. Buyya, "A Fuzzy Logic-Based Controller for Cost and Energy Efficient Load Balancing in Geo-distributed Data Centers," 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), Limassol, 2015, pp. 186-194.

[14] Javanmardi S., Shojafar M., Amendola D., Cordeschi N., Liu H., Abraham A. “Hybrid Job Scheduling Algorithm for Cloud Computing Environment,” IBICA 2014. Advances in Intelligent Systems and Computing, vol303. Springer, Cham.