Context Aware Load Balancing Strategy for Batch of Tasks in Cloud and Fog Computing Environment

Hiba Shakeel
Department of Computer Science and Engineering,Institute of Technology and Management, Aligarh, India
Sushil Kumar Sharma
Department of Computer Science and Engineering,Institute of Technology and Management, Aligarh, India

Abstract.

As the demand of load balancing (LB) in cloud and fog computing environment is growing, so the need for improved overall system performance. LB problem is verified to be NP-complete problem. The emerging use of heterogeneity is to process large number of intensive jobs in cloud and fog computing.In this paper, a Context Aware Load Balancing Strategy (CLBS) algorithm is proposed. The proposed LB strategy model has an intelligent scheduler like adaptive nature and this approach is used to minimize execution time of parallel jobs. CLBS works in two phases*viz.* the selection and allocation phase. In selection phase, CLBS chooses strategy for minimization of imbalanced load and makespan using fuzzy rule-based technique, while the allocation phase is carried out using minimum completion time. For the performance comparison, CLBS is compared with three existing state-of-artalgorithms from the literature. The simulation results of CLBS are showing better performance in terms of makespan and system utilization.

Keywords. Cloud Computing, Fog Computing, Load Balancing, Fuzzy rule-based technique, Intelligent Scheduler, Utilization.

1. INTRODUCTION

Cloud and fog are recent technologies which have shown drastic advancement in past few years. Cloud computing provides on-demand computational services to the users through internet where payment is based on the usage. The advantage of cloud is that it saves client's time and money that is wasted on local server's purchase and maintenance. Fog computing is descendant of cloud and services urgent requests that arrive from smart/IoT devices. Fog is new and has unique features like wireless connections, mobility support, location awareness, etc. With the increase in data of cloud and rigorous addition of smart devices to the network, cloud and fog face some serious issues like security, privacy and LB [1-3].

LB in cloud and fog computing is a method that allocates excess dynamic local workload to have ideally balanced computational machines [4-5, 19-22]. Balancing load ideally on all virtual machines (VMs) maximizes resource utilization and achieves better user satisfaction, ensuring that no VM is overloaded/underloaded, thus improving the system performance. It also helps in applying scalability,over-provisioning, fail-over, avoiding bottlenecks,minimizing response time and attaining Green Computing in cloud[6]. LB plays a vital role in cloud-cum-fog environment for the resource utilization of VMs that are globally distributed. IaaS cloud provides the resources for users' requests like execution of the tasks, storage of data, network access. Nevertheless, the capacity of resources can vary time to time for a specific user application. Due to this reason, the system performance of the IaaS can be affected. Thus, some resources get underloaded/overloaded ormay be idle for a long time, resulting in poor system performance [7]. The other issues related to it are limiting energy consumption and reducing carbon emission [8]. The main objectives of LB are - to keep the system stable and busy, increase the performance of the system, provide scope for future enhancement in the system and have a backup plan in case the system fails, even partially. The nature of adaptive LB is to help maximize overall system performance by confirming that user application requests are distributed and executed fairly over the set of computational machines.

When it comes to cloud and fog systems, there is a great deal of global uncertainty involved. Fuzzy logic concept can be used to reduce the effect of uncertainty in decision making process for LB algorithms [9]. For solving/running parallel and distributed job applications, the fuzzy logic technique for heterogeneous computing environments is intrinsically the best choice. All significant tasks are separated and distributed over numerous machines for parallel execution in this type of application. The chances of one of the machines being idle while another host has several jobs queued up can be very high for such type of system [10, 11].

1.1. Motivation

Cloud and fog receive diverse requests. Choosing the right LB strategy depends on the type of jobs that needs to be executed. For example, if the users have majority of the jobs with less amount of workloads for execution, Min-Min strategy will give better load balance and makespan as compared to SJFR-LJFR, LJFR-SJFR and Max-Min strategies. Similarly, if majority of jobs with great amount of workloads arrive, Max-Min strategy will give better load balance and makespan than Min-Min, SJFR-LJFR and LJFR-SJFR. Moreover, if the users have both some jobs with great amount of workloads and some with less amount of workloads, then strategy is chosen depending on type of jobs which are greater in number, for instance, if more heavy jobs with great amount of workloads arrive, then Max-Min will be better choice to balance the load with reduced makespan. This requires cloud and fog systems to use CLBS which is adaptive and depends on job specifications. Or else, one or more backend VMscan possibly become overloaded, whereas others will be underutilized.

1.2. Contribution

The contributions of the paper are summarized as follows: we are combining two recent works : ITSLB [12] and LBSM [13]. In this paper, adaptive natured scheduler is proposed named as CLBS to process an independent batch of tasks (BoT) in cloud and fog computing, having the objectives of optimizing load balance, makespan and system utilization. The proposed CLBS is a combination of both the strategies. CLBS model is developed for recognizing task pattern and best optimization strategies (i.e. Min-Min, Max-Min, SJFR-LJFR, LJFR-SJFR) to give best performance in terms of reduced tasks' finish time and load imbalance in cloud data center using fuzzy rule-based technique. In simulation results, CLBS is compared with LBSM, ITSLB and OLB. Results are reported consequently.

The rest of the paper is organized as follows: the next section is a summary of the related work that has been published in the literature. The design ideas, system overview, and problem description with parameter estimation are all covered in Section 3.

Section 4 describes the suggested Context aware Load Balancing Strategy (CLBS) algorithm and its version and Section 5 presents the simulation study and result analysis. The study concludes with final observations in Section 6. Section 7 suggests the future work.

2. RELATED WORK

An appropriate LB strategy helps in optimal resource utilization, thus reducing its consumption. Static and Dynamic LB are the two techniques of balancing load. Static LB strategy distributes work to all the resources before the actual execution of algorithm. Dynamic LB strategy is more complex and distributes work to all the resources during the execution. There are various independent static task scheduling algorithms such as Max-Min [15],Min-Min [14], Minimum Execution Time (MET) [15], Opportunistic Load Balancing (OLB) [15-16], Minimum Completion Time (MCT) [17], Longest Job in the Fastest Resource-Shortest Job in the Fastest Resource (LJFR-SJFR) [18], Independent Task Scheduling with Load Balancing (ITSLB) [12], Load Balancing Strategy with Migration Cost (LBSM) [13],etc.

MET [15] algorithm assigns tasks with minimum expected execution timeto the resourceswithout considering resource availability . Performance achieved is good as task is given to the fastest resource. It doesn't balance load well and does not fully support heterogeneous environment. The OLB [15-16]does random allocation of tasks to resources that can be available, so as to keep them busy to get maximum resource utilization. As it does not take into account the expected time to compute during task allocation, it may increase makespan.The purpose of MCT [17] is to assign task to the available resource which takes minimum expected completion time. The tasks which do not have MCT are also given resources. Thus, it combines benefits of OLB and MET with improved overall performance.In Min-Min [14], task related information is already available. It begins with set of unallocated tasks. It has two stages. The first stage finds MCT of all the tasks. While the second stage selects the task with minimum expected completion time (ECT) and a suitable resource is given to this task. This task is further removed from metatask once completed. The process loops until all the tasks complete.Min-Min is one of the simplest strategy for task assignment to the resources. The drawback is that the tasks with largest completion time will not be allocated for long time. It may not very well balance the load as few resources may remain idle and resultantly makespan may increase. Max-Min [15] starts with group of unallocated tasks. Max-Min also works in two stages. In first stage, it finds maximum completion time of task . It further calculates maximum ETC for each task. In stage two, task with maximum completion time is chosen and given suitable resource. This task is removed from metataskonce its completed. This process repeats until all the tasks are completed. Max-Min maps task with longer execution time on faster computational node, then task having shorter execution time are executed. So, overall mapping of task is better than Min-Min. Thus, Max-Min provides better LB as well as makespan.LJFR-SJFR [18]starts with group of unmapped task. It finds the MCT value from which task with least MCT called SJFR is selected. The task with maximum completion time from all the tasks is treated as LJFR.In LJFR-SJFR, alternate allocation of longest and shortest tasks is done to best resources. SJFR-LJFR allocates shortest and longest tasks to the best resources, alternatively.ITSLB [12] algorithm uses two variants namely ITSLB(Min-Max) andITSLB(Max-Max). In ITSLB (Max-Max), the load with greatest ETC value istransferred from maximum overloaded computational node (MOL) to maximum underloaded computational node (MUL). In ITSLB(Min-Max) strategy, migration of task will take place from MOL to MUL but here minimum to maximum ETC value of tasks are chosen for transfer . LBSM [13] algorithm uses two approaches for migration of tasks namely LBSM(LSTS) and LBSM(STS). In LBSM(LSTS) approach, firstly largest task is selected and transferred from MOL to MUL then smallest task is selected. They will be transferred alternatively from MOL to MUL and so on. When smallest task from MOL is transferred to MUL, it is referred to LBSM (STS). A brief comparative study has been shown in Table 1.

TABLE 1. A COMPRATIVE STUDY OF REALTED WORK

Techniques	Methodology	Advantages	Disadvantages
MET	Task is given to machine having MET	Overall low execution time of the system	Not good for LB
OLB	Task randomly given to machine expected to be available next	Keeps machine busy	Poor makespan
MCT	Task is mapped to machinewith MCT	Combines benefits of OLB and MET	Considers one task at one time
Min-Min	Task with MCT is given to machine with MET	Low makespan when majority of tasks are small. Lowers waiting time of small jobs	Poor resource utilization. Increased average waiting time of larger tasks
Max-Min	Task with maximum completion time given to machine with MET	Low makespan and good resource utilization. Lowers waiting time of large jobs	Increased average waiting time of smaller tasks
LJFR-SJFR	Alternate allocation of longest and shortest job to fastest machines	Does not let longer or shorter jobs starve	Inferior to Min-Min in makespan
ITSLB	Load is migrated based on the basis of expected time to compute usingmax ETC (Max-Max) and min to max ETC value (Min-Max)	Reduced load imbalance factor	Applied only to FCC network
LBSM	Task are migrated from overloaded to underloaded machines based on two strategies which are LSTS and STS	Minimizes load imbalance	Connection is needed for migration of task

3. BACKGROUND AND PROBLEM FORMULATION

The Context Aware Load BalancingStrategy (CLBS) is designed for independent BoT allocation on heterogeneous VMs in cloud-cum-fog environment to optimize the makespan and resource utilization. This section describes various components related to proposed strategy such as cloud and fog system overview, problem description and parameter estimation.

3.1. System Overview

The recent real time and scientific applications such as smart city, processors, chips, health monitoring systems and many more have extended the use of cloud and the emerging fog computing technology. The current trend is hosting real time and scientific applications in the cloud cum fog environment. Therefore, in this paper, we consider the scheduler model for the proposed approach in cloud-fog environment. Cloud and fog computing environment is a collection of resources, where each resource contributes to solve complex users' problems. The heterogeneous structure is made up of a number of organization, each of which has a set of resources. Our suggested system is based on a centralized architecture, with independent batch of tasks being sent to any VM andthe eventual dispatch of the independent BoT based on the appraisal of a VM's fitness for a given workload. The present state of the system is maintained by our proposed strategy using the following information:

- The time of processing, the number of VMs from various domains available.
- Clock frequency of the VMs.
- The VMs' ready time reflecting the current load status on each VM in the scheduler.

3.2. Problem Description and Parameter Estimation

Consider the following scenario: the scheduler model under consideration comprises VMs with variable speed capacities. Let the set of VMs $K = \{V_j: j = 1, 2,\ldots,n\}$ and batch of m independent task $T = \{T_i : i = 1, 2,\ldots,m\}$.The problem is finding the mapping $(\mathcal{M})$ for batch of independent tasks T assigned on the set of VMs K

$$\mathcal{M} : T \rightarrow K \qquad (1)$$

After generation of workload, the Expected Time to Compute (ETC) matrix model can be calculated in the following way:

$$ETC_{i,j} = \frac{\mathcal{WL}_i}{\mathcal{CC}_j} \qquad (2)$$

The capacity of task will be calculated in Millions of Instructions(MIs) and VM capability in MIPS. The average ETC is computed as:

$$\bar{E} = \frac{\sum_{j=1}^{n} ETC_{i,j}}{n} \qquad (3)$$

After computation of E matrix and , CLBS recognizes the pattern of workload and using fuzzy based rule determines which optimization strategy will be suitable for this particular type of independent tasks. The fuzzy rule-based technique is shown in section 4. Now, for each allocation of task, the completion time (CT_{ij}) of T_i on V_j is required. Therefore, it is computed by using equation (4) as follows:

$$CT_{ij} = RT_j + ETC_{ij} \qquad (4)$$

whereRT_j is the ready time of V_j. Because makespan (Ms) is a crucial parameter for task execution, a decrease in makespan is regarded as an enhancement in system performance. The shorter the makespan, the better the task schedule, meaning that all of the allocated tasks are completed in less time. It is possible to calculate the makespan as follows:

$$Ms = \max_{1 \le j \le n} \left(\sum_{V_k \vdash \forall T_i} CT_{ij} \right) \qquad (5)$$

For the owner's perspective, average resource usage (U_g) is an important optimization criteria for improving the performance of the IaaS system. For a given allocation, the U_g of heterogeneous VMs for an independent BoT can be computed as:

$$U_g = \frac{\sum_{j=1}^{n} CT_j}{n \times Ms} \qquad (6)$$

4. PROPOSED MODEL

This section presents the Context Aware Load Balancing Strategy (CLBS) to optimize the makespan and resource utilization in cloud-cum-fog computing environmentwhile analysing its various features. The strategy is categorized in two major phases: *selection* and *allocation phase*.The first phaseisused for determining the best optimization strategy on the basis of tasks' workload pattern while *allocation phase* is as per completion time of each task.

The various scheduling algorithms are available for this work such as Max-Min, Min-Min, SJFR-LJFRand LJFR-SJFR. The Min-Min approach gives better result when the majority of task has minimum workload. Similarly, Max-Min strategy works better when majority of task has maximum workload. The proposed CLBS works as an intelligent system using fuzzy rule-based techniques. The proposed CLBS is itself dynamic and adaptivein nature, as optimization strategy is not predetermined and rather chosen according to the workload pattern of majority of tasks arriving (i.e. on the basis of context). After generation of workload of tasks, CLBS model determines minimum workload value which is subtracted from all workload

values. After subtracting, from the resultant values, CLBS determines maximum workload value which divides all workload values. Thus, all tasks' workload values are converted in the range of 0 and 1. The range of these workloads is categorized into two parts, such as 0 to less than 0.5 called as *minimum workload value* and greater than or equal to 0.5 to less than or equal to 1 which is called *maximum workload value*. The percentage of workloads belonging in this categorizationis calculated and used in four fuzzy membership functions, such as low, moderately low, moderately high and high workload values, as depicted in Fig 1.

For computation of fuzzy membership function,equation (7) is considered. After computing these membership functions, if the percentage of minimum workload value lies between 0 to 20% that means 20% tasks are of minimum workload value and rest are of the maximum workload value. In this condition, CLBS model will use Max-Min strategy for allocation of tasks. Similarly, if minimum workload value lies in between 20% to 40%, 40% to 60%, 60% to 80% and 80% to 100% then CLBS model will use Max-Min, LJFR-SJFR, SJFR-LJFR and Min-Min strategies for task allocation, respectively.

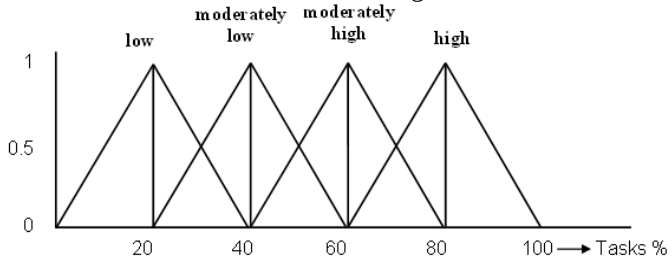

Fig. 1. Fuzzification for optimization strategy

The fuzzy rule-based technique is as follows:

$$\mu_{low} = \begin{cases} \frac{20-x}{20-0}, if\ x < 20 \\ 1,\ if x = 20 \\ \frac{x-20}{40-20}, if\ 20 < x < 40 \end{cases}$$

$$\mu_{moderatelylow} = \begin{cases} \frac{40-x}{40-20}, if\ 20 < x < 40 \\ 1,\ if x = 40 \\ \frac{x-40}{60-40}, if\ 40 < x < 60 \end{cases} \quad \mu_{moderatelyhigh} = \begin{cases} \frac{60-x}{60-40}, if\ 40 < x < 60 \\ 1,\ if x = 60 \\ \frac{x-60}{80-60}, 60 < x < 80 \end{cases} \quad \mu_{high} = \begin{cases} \frac{80-x}{80-60}, if\ 60 < x < 80 \\ 1,\ if x = 80 \\ \frac{x-80}{100-80}, 80 < x \leq 100 \end{cases}$$

(7)

Where x represents the percentage of minimumworkload value.Once CLBS model will recognize best optimization strategy by using equation (7). Then allocation will begin according to completion time of each task as per equation (4). The average ETC is the threshold for the level of load balance. If because of assigning a task, that machine exceeds the average ETC, then those tasks will be assigned to another machine, which fits best. The pseudo code of CLBS is as follows:

CLBS ( )

Input: *m*, *n*, $\mathcal{WL}_i$,$\mathcal{CC}_j$

Output: Schedule, *Ms and*$U_{\bar{s}}$

Begin

1. Generate random $\mathcal{WL}_i$
2. Determine optimization strategy as per eq. (7)
3. Compute *ETC* matrices and $\bar{\bar{E}}$ as per eq. (2) and (3)
4. Sort *ETC* in ascending order
5. ***while*(*T* != ∅)**
6. ***for*** $\forall T_i \in T$
7. ***For*** $\forall V_j \in K$
8. Compute CT_{ij} as per eq. (4)
9. ***end for***
10. ***end for***
11. ***end while***
12. Assign the task to machine $V_j$that offers least completion time
13. Repeat steps 6 -10 until the set of tasks becomes empty
14. Compute *Ms* and $U_{\bar{s}}$ as per eq. (5) and (6)

5. SIMULATION RESULTS

By observing the allocation of the independent BoT in IaaS cloud-cum-fog computing for varied input parameters, simulation results were used to evaluate the proposed study. On an Intel CORE i5 processor with 8 GB RAM, simulations were done using MATLAB 2015. For modelling of heterogeneous VM environments with processing capacity within a specific range, the parameters related to cloud-cum-fog computing were taken into account. IndependentBoT is randomly generated having a specified range of tasks *m* = 100-5000, *n* = 8-64, CC_j = 10-30 (MIPS), RT_j = 20-1000ms, and WL_i = 1000-10000 (MIs).

The experimental results are evaluated by using independent BoT varying in VMs and number of tasks using ITSLB(Min-Max), ITSLB(Max-Max), LBSM(STS), LBSM(LSTS) and OLB algorithms. The purpose of the experimental study was to compare the performance of the techniques using performance indicators like makespanand average resource utilization. Experiments are carried out by: (1) Observing *Ms* and of the BoT on VMs while keeping the number of VMs same but varying the BoT size and (2) Observing the *Ms* and of the BoT on VMs while keeping the size of BoT same but varying the number of the VMs.

5.1. Varying BoT

In this section, considered parameters makespan and resource utilization is observed while keeping the number of VMs same and varyingBoT's size on VMs.The input parameters for these experiments are number of VMs, n = 16 and m =100-5000

- Fig. 2 clearly shows that when number of task increases, makespan also increases. It is also evident that proposed CLBS outperforms other algorithms, considering the makespan. CLBS has least makespan and OLB has most. LBSM(LSTS), LBSM(STS), ITSLB(Min-Max) and ITSLB(Max-Max) has almost equal makespan. The performance gain of CLBS is 24% (approx.).
- Fig.3 depicts that CLBS has best resource utilization than peer algorithms and OLB has worst, while LBSM(LSTS), LBSM(STS), ITSLB(Min-Max) and ITSLB(Max-Max) has almost similar utilization. The performance gain of CLBS is 15% (approx.) over existing work.

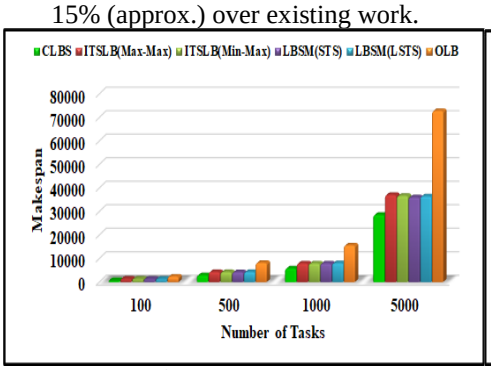

Fig. 2. Number of tasks vs. makespan

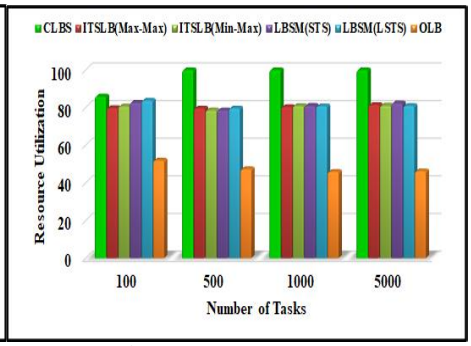

Fig. 3. Number of tasks vs. resource utilization

5.2. Varying VMs

In this section, considered parameters, makespan and resource utilization for the BoT are observed on VMs by varying number of VMs and keeping size of BoT same. The input parameters for these experiments are number of VMs, n = 8- 64 and$m$ = 5000.

- From fig. 4, it can clearly be seen that makespan reduces as the number of VMs (n) increases from 8 to 64. It is also observed that CLBS has minimum makespan and OLB has maximum, while LBSM(LSTS), LBSM(STS), ITSLB(Min-Max) and ITSLB(Max-Max) gives almost equal makespan in every single case. The performance gain of CLBS is 22% (approx.) over existing work.
- From fig. 5, it can be determined that CLBS has best resource utilization while OLB has worstfor every case of VM. Peer algorithms like LBSM(LSTS), LBSM(STS), ITSLB(Min-Max) and ITSLB(Max-Max) gives almost similar results.The performance gain of CLBS is 18% (approx.).

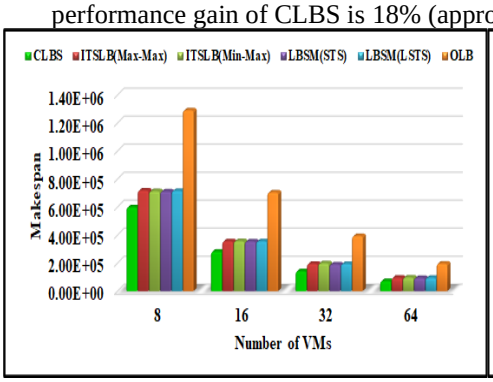

Fig. 4. Number of VMs vs. makespan

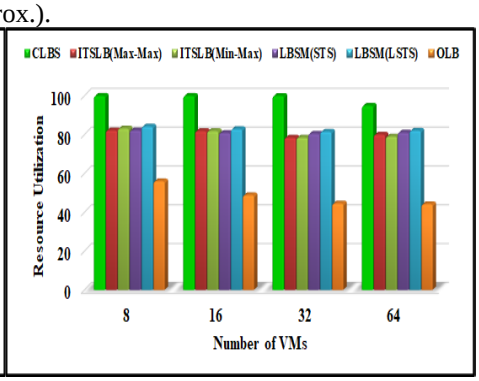

Fig. 5. Number of VMs vs. resource utilization

6. CONCLUSION

In this paper, a Context Aware Load Balancing Strategy (CLBS) isproposed to optimize the makespan and resource utilization in cloud-cum-fog environment. The CLBS is an intelligent scheduler like adaptive in nature which selects the optimization strategy using fuzzy rule-based technique. The experimental results showedthat CLBS outperformed ITSLB, LBSM and OLB strategies in terms of makespan and resource utilization.

7. FUTURE WORK

Performance of CLBS can be enhanced by taking more Quality of Serviceparameters under consideration and improving them. Strategies focused on improving multiple metrics at once can be developed to improve overall load balancing in cloud and fog computing.

8. REFERENCES

[1] Buyya, R., Vecchiola, C., &Selvi, S. T, "Mastering cloud computing: foundations and applications programming", Newnes, 2013.

[2] Masdari M, ValiKardan S, Shahi Z, Azar SI, "Towards workflow scheduling in cloud computing: a comprehensive analysis," Journal of Network and Computer Applications, *1*(66), pp.64-82, 2016.

[3] Marinos A, Briscoe G, "Community cloud computing," In IEEE international conference on cloud computing, pp. 472-484, 2009. Springer, Berlin, Heidelberg.

[4] Alam, M., & Khan, Z. A. "Issues and challenges of load balancing algorithm in cloud computing environment," Indian Journal of Science and Technology, *10*(25), pp. 1-12, 2017.

[5] Alam, M., Haidri, R. A., Mahek&Yadav, D. K. "Efficient task scheduling on virtual machine in cloud computing environment," International Journal of Pervasive Computing and Communications vol. *17*(3), pp. 271-287, 2020

[6] Kansal NJ, Chana I. Cloud load balancing techniques: A step towards green computing. IJCSI Int J ComputSci Issues. 2012;*9*(1), pp. 238–246.

[7] I. Foster, C. Kesselman, and S. Tuccke, "The Anatomy of the Grid," International Journal of Supercomputer Applications, 2001.

[8] Nidhi Jain Kansal, Inderveer Chana, "Cloud Load Balancing Techniques: A Step Towards Green Computing", IJCSI, *9*(1), January 2012

[9] Chulhye Park and Jon G.Kuhl, "A Fuzzy based distributed load balancing algorithm", Proceedings of the Second International Symposium on Autonomous Decentralized Systems (ISADS'95) IEEE, 1995.

[10] Yu-Kwong Kwok And Lap-Sun Cheung, "A New Fuzzy-Decision Based Load Balancing System For Distributed Object Computing", Journal Of Parallel And Distributed Computing, Volume *64*(2), February 2004.

[11] Kwok, Y. K., & Cheung, L. S. (2004). A new fuzzy-decision based load balancing system for distributed object computing. Journal of Parallel and Distributed Computing, *64*(2), 238-253.

[12] Khan Z, Alam M. and Haidri R. A. "Effective Load Balancing Scheduling Schemes for Heterogeneous Distributed System", International Journal of Electrical and Computer Engineering (IJECE), *7*(5), pp.2757-2765, Oct 2017.

[13] Alam M. and Shahid M. "A Load Balancing Strategy with Migration Cost for Independent Batch of Tasks (BoT) on Heterogeneous Multiprocessor Interconnection Networks", International Journal of Applied Evolutionary Computation (IJAEC), *8*(3), pp.74-92, 2017.

[14] Etminani K, Naghibzadeh M. "A min-min max-min Selective Algorihtm for Grid Task Scheduling." In Internet, 2007. ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia on 2007; pp. 1-7. IEEE.

[15] Freund RF, Gherrity M, Ambrosius S, Campbell M, Halderman M, Hensgen D, Keith E, Kidd T, Kussow M, Lima JD, Mirabile F. "Scheduling Resources in multi-user, Heterogeneous, Computing Environments with SmartNet." In Heterogeneous Computing Workshop, 1998. (HCW 98) Proceedings. 1998 Seventh 1998; pp. 184-199. IEEE.

[16] Sang A, Wang X, Madihian M, Gitlin RD. "Coordinated Load Balancing, handoff/cell-site selection, and Scheduling in multi-cell Packet Data Systems." Wireless Networks, *14*(1), pp. 103-20, 2008.

[17] Braun, T. D., Siegel, H. J., Beck, N., Bölöni, L. L., Maheswaran, M., Reuther, A. I., & Freund, R. F. et al. "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," Journal of Parallel and Distributed Computing, *61*(6), pp. 810–837, 2001.

[18] Abraham, A., Buyya, R., &Nath, B. Nature's heuristics for scheduling jobs on computational grids. In The 8th IEEE international conference on advanced computing and communications (ADCOM 2000), pp. 45-52, (2000, December).

[19] Alam, M., &Varshney, A. K. "A New Approach of Dynamic Load Balancing Scheduling Algorithm for Homogeneous Multiprocessor System," International Journal of Applied Evolutionary Computation (IJAEC), *7*(2), pp. 61-75, 2016.

[20] Haidri, R. A., Alam, M., Shahid, M., Prakash, S., &Sajid, M. "A deadline aware load balancing strategy for cloud computing," Concurrency and Computation: Practice and Experience, *34*(1), pp. e6496, 2022.

[21] Alam, M., Haidri, R. A., &Shahid, M. "Resource-aware load balancing model for batch of tasks (BoT) with best fit migration policy on heterogeneous distributed computing systems," International Journal of Pervasive Computing and Communications, *16*(2), pp. 113-141, 2022.

[22] Khan, S., Nazir, B., Khan, I. A., Shamshirband, S., &Chronopoulos, A. T. "Load balancing in grid computing: Taxonomy, trends and opportunities," Journal of Network and Computer Applications, *88*, pp. 99-111, 2017.

Biographies

HibaShakeelreceived her bachelor's and master's degree in Computer Science and Applications from Aligarh Muslim University, Aligarh, India in the year 2017 and 2020, respectively. Her research interest areas include Load balancing, Cloud Computing and Fog Computing.

Sushil Kumar Sharma received his MCA and M.Tech degree from GLA College, Mathura, India in the year 2005 and 2012, respectively. He holds 15 years of teaching experience and is Assistant Professor at Institute of Technology and Management, Aligarh, India. His research interest areas include Big Data and Cloud Computing.