Optimized Hyper Heuristic Scheduling using Adaptive Load Balancing Algorithm in Cloud Computing

Md Oqail Ahmad
Department of Computer Science, Aligarh Muslim University, Aligarh, India
Email: oqail.jmu@gmail.co;

Rafiqul Zaman Khan
Department of Computer Science, Aligarh Muslim University, Aligarh, India
Email: rzk32@yahoo.co.in;

***Abstract*—** Effective scheduling is important for the inclusive performance of cloud computing system. The main problem of scheduling is how to distribute the entire tasks to an appropriate virtual machine (VM). The scheduling of the tasks to suitable VM for computing of the requests in cloud computing is originate to be an NP-complete issue. An optimized algorithm is essential to schedule the tasks on the VMs to decrease the makespan time, degree of imbalance and total processing cost. In this paper, we proposed a new optimized Hyper-Heuristics Scheduling using Adaptive Load Balancing Algorithm named as HHS-ALBA, for lookup optimized scheduling solution for cloud computing. The proposed HHS-ALBA algorithm uses the idea of load balancing by the solution identified with the hyper heuristic algorithm to optimize the solution. The results show that the proposed HHS-ALBA algorithm could outperform in term of makespan, degree of imbalance factor, total processing cost and total processing time, and compared with existing HHSA and traditional algorithms. Moreover, the proposed HHS-ALBA algorithm is comparatively and statically more effective than the heuristics algorithms like ACO, PSO GA and traditional algorithms FCFS and SJF. Performance evaluation was done with CloudSim Simulator.

***Index Terms*— Hyper heuristic, Adaptive load balancing, HHSA, Cloud computing, Low-Level Heuristic, Performance parameters.**

I. INTRODUCTION

Cloud computing is technology, that offers several online assistance like application, space and infrastructure on demand pay-per-you-go concept [1]. These resources are allocated and de-allocated as per application demand while considering resource availability and performance. The requests are coming day-by-day at cloud servers, and service provider is not aware about this upcoming workload [2, 3]. Scheduling plays a vital role in efficiently managing that upcoming load to the users request and running status of VMs at datacenter. It is one of the most famous exercise that performs in the cloud circumstances, to grow the effectiveness of the load to get most extreme benefit [2].

Most of the rule based scheduling algorithm is normal scheduling algorithm considerably applied on today's appropriate cloud computing system. These algorithms are common and not difficult to apply, but these are difficult in managing the large size or difficult scheduling issues [2, 4-6] due to unusual far from optimal results. So, to solve this issue to get better the performance of scheduling algorithm [7-10] an optimized hyper heuristic scheduling using adaptive load balancing algorithm is used to find superior scheduling results for cloud computing system. The most decisive concept of Hyper-Heuristic [4] is to opt precise algorithms for a precise problem on the basis of bunch of existing algorithms as well as their execution to some amount [11,12]. The important collection of the projected algorithm is to possess the value of low-level heuristic algorithms [15] such as, Generic Algorithm (GA), Ant Colony Optimization (ACO) [13, 16] and Particle Swarm Optimization (PSO) [14] by integrating them into single algorithm. Mixing of heuristic algorithms delivers improved results than as compared to rule based scheduling algorithm [17].

The existing Hyper-Heuristics Scheduling (HHS) algorithm is only scheduling the tasks and optimizing the solution based on the operators. But the concept of load is taken into consideration in this technique which will more optimize the solution and reduce the makespan time. The concept of Adaptive Load Balancing (ALB) algorithm is merged with the hyper heuristic concept i.e. the solutions generated from the proposed algorithm HHS-ALBA are more optimized and improved. For this implementation, each and every low-level scheduling algorithm also contain the adaptive load balancing in their solution which will further utilized in the hyper technique to make the solutions more optimized with the functionality of improvement operator of hyper heuristic algorithm. The main idea of our proposed methodology are as follows:

1. The proposed HHS-ALBA algorithm uses the concept of load balancing by the solution identified with the hyper heuristic algorithm to optimize the solution.
2. Load balancing technique approves the system load balancing by computing the capacity on each VM and then relocating the task according to load on each machine and the deadlines of the tasks.
3. The concept of adaptive load balancing is merged with the hyper heuristic concept i.e. the solutions generated from the proposed algorithm are more optimized and improved.
4. The proposed algorithm, called optimized Hyper-Heuristics Scheduling using Adaptive Load Balancing Algorithm (HHS-ALBA).
5. Performance evaluation results proved that the proposed HHS-ALBA algorithm could outperforms in term of makespan, degree of imbalance , total processing cost and total processing time compared with existing HHSA, the heuristics algorithms are ACO, PSO, GA and traditional algorithms FCFS and SJF. Performance evaluation done with CloudSim Simulator.

The rest of this paper is structured as follows: Section II a short review of the research works based on heuristics scheduling in cloud computing. Section III describes about proposed approach of Hyper Heuristic Algorithm with Adaptive Load Balancing based on deadline, and also parameters estimation and method description; in Section IV experimental evaluations and results are carried out by the comparison of existing scheduling algorithms, and Section V deals with conclusions and future direction of proposed HHS-ALBA algorithm.

II. RELATED WORK

This section, we describe some of the related work in the area of hyper-heuristic scheduling in cloud computing.

G. Kaur and S. Kaur [2] have been proposed an enhanced Hyper-Heuristic Scheduling Approach by Load balancing and RASA for cloud computing systems. This approach schedule tasks and resources to optimize the processing time and processing cost by the use of two detection operator. The concept of load balancing and RASA is useful for effective scheduling, utilization of resource to improve the efficiency of cloud computing system. The researcher develop the simulation platform and analyze the proposed approach in the CloudSim environment. The experimental outcome shows that the projected approach achieves better performance compared to individual heuristics approach in term of to reduce makespan time.

C. W. Tsai et al. [4] have projected a novel hyper-heuristic scheduling algorithm (HHSA) to reduce makespan time for cloud computing. This algorithm used two detection operator; (i) diversity detection and (ii) implementation operator are engaged to figure out which low-level heuristic is used to find improved candidate result. To assess the efficiency of HHSA are developed and implemented on CloudSim as well as Hadoop. The implementation results shows HHSA can slightly reduce the makespan time compared to other scheduling algorithm.

M. D. P. Vigneshwaran et al. [12] have introduced a new hyper-heuristic scheduling algorithm based on the results of existing heuristics algorithms. The proposed algorithm deliver the best solution for cloud computing by reducing the makespan time. The proposed heuristics used two detection operators, they are improvement detection and diversity detection operator to find time to employ the LLH algorithms. The algorithm compared with other heuristics algorithm, it converges faster by using two detection operators. The efficiency of proposed algorithm is estimated on CloudSim as well as real time environment.

S. Mohanty et al. [19] proposed meta-heuristic load balancing algorithm using Particle Swarm Optimization (PSO) names as

Multi PSO. The proposed Multi PSO algorithm aim to optimize the task overhead time and resource utilization. To evaluate the performance of this algorithm used CloudSim Simulator and the comparison are made with FCFS, RR, Min-Min, GA and PSO algorithms to different measure like the makespan time and the resource utilization. The experimental result shows the Multi PSO outperforms better than scheduling algorithm in term of makespan time and average resource utilization.

S. R. Kumar et al. [20] proposed an Adaptive Earliest Deadline First (AEDF) algorithm based on the traditional Earliest Deadline First (EDF) algorithm. This algorithm anticipated for load balancing issue in demand to attain effective utilization of cloud resources as well as backing Green Computing technology. The AEDF algorithm compared with conventional load balancing algorithm, the proposed algorithm outperform in term of throughput, effective scheduling of tasks and minimized cost of dynamic server use to backing Green computing model.

A. Jain and A. Upadhyay [21] have proposed the improved heuristic scheduling algorithm to reduce the makespan time and maximizing the cloud resources utilization and also find out he superior task scheduling solutions for cloud computing with help of two meta-heuristics algorithms and three traditional algorithm on varying datasets. The implementation outcomes shows that proposed IHHSA algorithm provide superior than meta-heuristic algorithm and traditional algorithm. The aim of proposed algorithm is to control over the assets of entire the low level heuristics algorithm by executing one and only one LLH algorithms at a time.

S. H. H. Madni et al. [22] have discussed some important rule-based heuristic algorithms like Min-min, Max-min, Suffrage, FCFS [10], MCT and MET are measured for the efficiency evaluation and scrutiny scheduling of task in cloud computing. These rule based algorithm are developed and implemented to schedule independent tasks in homogeneous and heterogeneous environments to optimize the makespan time, throughput, cost and degree of imbalance factor by compared with other existing rule-based heuristics algorithm. The experimental results are performed with the support of CloudSim simulator. The results shows MET always outperform better than other heuristics algorithm.

N. Xoxa et al. [23] have discussed two batch scheduling algorithm like FCFS and SJF, it also show how to increase utilization of resource and overall efficiency of the system. In this paper, the authors also present the simulation code of FCFS and SJF, than executing the code of these scheduling algorithm, it shows SJF is more competitive than FCFS.

S. Sindhu et al. [24] have discussed many scheduling algorithm like Genetic Algorithm (GA), Particle Swarm Optimization(PSO), Min-Min and Max-Min, that are efficient in resource utilization of cloud computing.

From the brief review of literature, we appreciate that the whole thing are well organized, such as scheduling in cloud computing. There are many works in the literature about the hyper-heuristics algorithm that have been developed and implemented; however, most of them are applied in scheduling in cloud computing without load balancing. Therefore, this paper uses the idea of adaptive load balancing by the solution identified with the hyper-heuristic algorithm to optimize the solution.

III. PROPOSED APPROACH

In this research, we proposed optimized Hyper Heuristic scheduling using Adaptive load balancing algorithm based on deadline in cloud computing. In this section, the existing HHSA and ALB algorithm are discussed. Further, the proposed HHS-ALBA algorithms had been illustrated.

A. Hyper Heuristic Scheduling Algorithm

Hyper-heuristic [4] is a technique of integrating more than two heuristic algorithms with rule-based algorithms to solve the complex scheduling problems are like NP-complete [25]. Combining of heuristic algorithms presents better solutions than a tentative heuristic algorithm. As shown in Fig. 1, one of the heuristic algorithm in the pool of heuristic algorithms of candidate heuristics {H1, H2 … Hn} where n is the population size. The heuristic Hi will be selected randomly from the candidate pool by the Low-Level Heuristic (LLH) selection operator, which is referred to as Hi in Fig. 1. For this kind of heuristics, the selection and acceptance LLH operators that can be used to determine which heuristic algorithm is to be selected [18, 26]. Further, the selected Hi could be continuously used or not are. LLH selection operator are used to determine the Hi and the LLH acceptance operator used to determine the timing to select a new Hi. Whereas, Hi repeat a solution for a definite number of repetition either the Hi decreases into limited optimal.

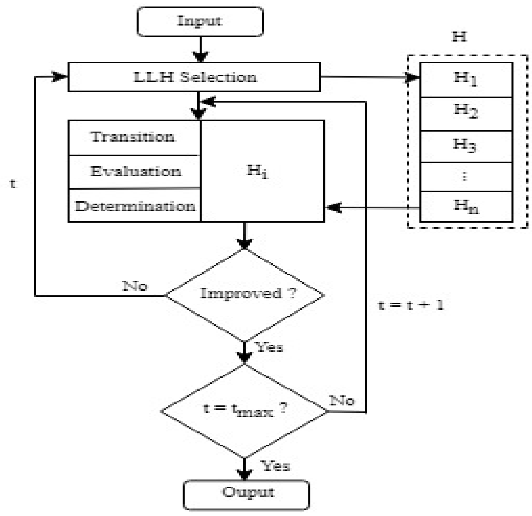

Fig.1., Worlflow of HHSA Algorithm [2]

B. Adaptive Load Balancing based on deadline

S. Ramkumar et al. [20] have introduced Adaptive Earliest Deadline First (AEDF) algorithm based on conventional Earliest Deadline First (EDF) algorithm to explain load balancing difficulty in cloud computing. The proposed AEDF algorithm is based totally on threshold value to gain the dominant scheduling and utilization of resource. The AEDF algorithm provides optimal result for load balancing in cloud environment. Based on condition of the minimum utilization threshold value, tasks follow traditional EDF scheduling to perform the tasks. When utilization of the task value more than the threshold value at that time tasks are assigned in AEDF list based on limitation of compatible tasks. The experimental evaluation suggests AEDF accomplishes effective scheduling and utilization of resources for cloud load balancing issues.

C. Scheduling Problems and Parameter Estimation

The main objective of proposed methodology is to assign the task to appropriate VM of cloud services. Scheduling play significant role in guiding task within the cloud. Scheduling process first analyses, how much of resources are having to need to complete the task and which task to be allocated to which computing component. The abstract architecture scheduling of task is shown in Fig. 2. Here, the cloud user sends the task to task manager. Task manger receives the tasks from users through user interface and then gathers and manages all accepted users demand. Then the task is sent to task scheduler. The process of scheduling happens by consuming the reserved task based on the efficiency of VMs. The information from the resource manager and load balancer is gathered by the task scheduler and then takes a decision to assign each task to the VM based on HHS-ALBA algorithm.

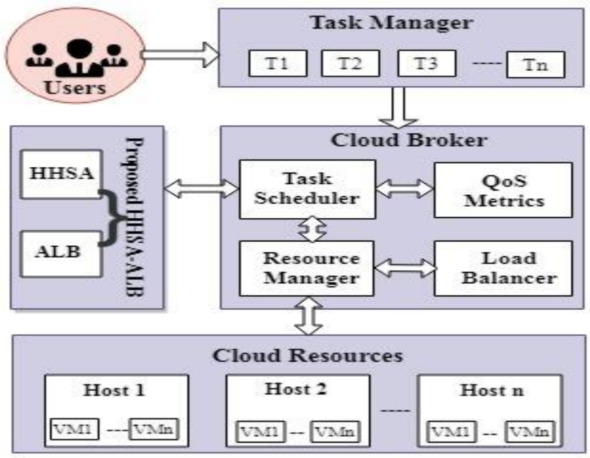

Fig.2. Abstract architecture of Task Scheduling

The main intent of the scheduler is difficult to assign the entire task based on available resources while optimizing the QoS metrics. To finding an optimal solution of this research problem, a set of independent tasks, T = {T1, T2 ….Tn} from different users has to be assigned to set of heterogeneous available resources R = {R1, R2… Rm} where n is the number of tasks and m is the number of resources. This can be done in such way that several performance parameters will be fulfilled. The equations and performance parameters have been defined below: [19, 22, 27-29].

Table 1.
LIST OF NOTATIONS WITH DESCRIPTION

Notations	Description
T	Set of task submitted in cloud i.e $T_i=\{T_1,T_2,\ldots,T_n\}$
R	Set available VM cloud Resources $R=\{R_1, R_2,\ldots, R_m\}$
n	Number of tasks arrived at any time interval
m	Number of available VM resources
T_{len}	Sum of cloudlet sizes in CloudList
Cloudlet/Task	CloudSim simulator job
MI	Size of task in MI(Million Instructions)
VMs	Virtual Machines
C_{vm}	The capacity of virtual machine
S_{time}	Submission time of the task
LLH	Low Level Heuristic
H_i	Heuristic algorithm
F1	Improvement detection operator
Sum_{Ri}	Sum of resource Ri per unit time
CR_i	cost of resource i (R_i)
RT	Response time
DT_i	Deadline of task T_i
AEDF	Adaptive Earliest Deadline First
EDF	Earliest Deadline First
MT	Makespan Time
FT	Finishing Time of task T_i
EC	Economic Cost
UT_i	Unit Time resource i
DI	Degree of imbalance factor
NT	New time
T_{mx}	The maximum T_i
T_{mn}	The minimum T_i
T_{ag}	the average T_i of VMs
TH	Threshold value of task

Table 1. gives the explanations of the used symbolizations. Equations used in the proposed work are as follows:

C_{vm} : The capacity of virtual machine is calculated with help of Eq. (1) as follows:

$$C_{vm} = \text{VmMips} * \text{NumberOfPes} \tag{1}$$

ET: The execution time a task at virtual machine can be calculated by suing Eq. (2):

$$ET_i = T_{len} \,/\, C_{vm} \tag{2}$$

where T_{len} length of the task and C_{vm} capacity of virtual machine.

DT_i: Each task has its own deadline time which is given by Eq. (3):

$$DT_i = (ET_i) * K \tag{3}$$

Where K is deadline parameter. K = 1,2,3…,n.

FT: The finishing time of a task T_i at virtual machine can be calculated by using Eq. (4)

$$FT = S_{time} + RT \tag{4}$$

Where RT is response time and S_{time} submission time of the task T_i

RT : The response time of the task can be formulated with help of Eq. (5)

$RT = FT - S_{time}$ (5)

Here FT represent the finishing time and S_{time}represent submission time of the task T_i.

UT : The utilization time of the task T_i is calculated with help of Eq. (6)

$UT_i = ET_i / DT_i$ (6)

Makespan Time (MT)

It specifies the total finishing time of every tasks. Reducing the makespan is the most well-known optimization standards for task scheduling. Which is computed by:

$MT = max\{FT_i | j = 1,2,3..,m\}$ (7)

where FT_i indicates the finishing time as defined in Eq. (4) of specific cloudlet i.

Economic Cost (EC)

EC signifies the entire of clearance to cloud provider as contrasting to the resource utilization. It is computed as follows:

$EC = Sum_{Ri}(CR_i * UT_i) \forall i \in N, i = 1, 2, 3... m$ (8)

where CR_i signifies the rate of resource i (R_i) per unit time as well as UT_i as defined in Eq. (6) means utilization time of (R_i).

Degree of Imbalance (DI)

The DI is small value of the load of the system that measures the load sharing between the VMs about to their execution ability. It is computed as follows:

$DI = (T_{mx} - T_{mn})/T_{ag}$ (9)

where T_{mx} and T_{mn} are the maximum and minimum T_i between all VMs, T_{ag} is the average T_i of VMs.

D. Proposed HHS-ALBA Algorithm

We proposed and implemented an optimized Hyper Heuristic Scheduling using Adaptive Load Balancing (HHS-ALBA) algorithm in cloud computing. The main intent of the proposed algorithm is to improve the scheduling result in terms of Makespan Time (MT), Degree of Imbalance (DI), and Economic Cost (EC). The existing Hyper-Heuristic Scheduling Algorithm (HHSA) is the mixing of two or more heuristics algorithms. This algorithm combine's hyper heuristic algorithm are GA, PSO and ACO and traditional algorithms FCFS and SJF to commence the perfect scheduling results with reduced calculation time. The HHSA uses improvement operator to determine once to modification the LLH algorithm and a perturbation operator to optimize the results achieved by each LLH algorithm. Additional, optimize the scheduling outcomes in terms of Makespan Time i.e., if the solution is not improved until some iterations reached.

Algorithm 1: Hyper Heuristic Algorithm

Input: Set of independent tasks, T = $\{T_1, T_2, \ldots. T_n\}$ from different users has to be scheduled to set of heterogeneous available resources R = $\{R_1, R_2 \ldots R_m\}$ that is scheduling problem.

Output: Optimize the Makespan, Total processing cost, total processing time, and Degree of Imbalance (7), (8) and (9).

1. Set up the parameters; $\emptyset_{max}$ is number of iterations the chosen LLH is to run and $\emptyset_{ni}$ is the number of iterations LLH is performed.
2. Input: the scheduling problem.
3. Set the population of results $P = P_1, P_2, P_3 \ldots \ldots \ldots \ldots . P_z$
4. Randomly choose a heuristic algorithm H_i from candidate pool H, the algorithm combine's algorithm are GA, PSO and ACO, and traditional algorithms FCFS and SJF
5. While the termination criterion is not fulfilled.
6. Update the population of results P by applying proper algorithm H_i.
7. Improvement Detection operator i.e. F1 = Improved Detection (P).
   a) Check if the solutions of the chosen H_i is enhanced
   b) If the solutions are not enhanced till $\emptyset_{ni}$ iterations then return "false" and choose alternative algorithm from candidate pool (H).
8. Perturbation operator i.e. P= Perturb (P) reanalyse the results.
9. End While.
10. End.

Algorithm 2: Hyper Heuristic Algorithm with Adaptive Load Balancing based on deadline

Evaluate the load for the best result produced by the selected low-level algorithm then apply adaptive load balancing on that result. Then that result will be used by Hyper Heuristic algorithm.

Input: For all tasks T_i optimized using low level scheduling algorithms.

Output: Optimize the Makespan, Economic Cost and Degree of Imbalance (7), (8) and (9).

1. Find the ET_i as defined in Eq. (2) and the DT_i as defined in Eq. (3)
   Calculate UT_i as defined in Eq. (6)
   If (T_i (UT_i)) < TH
   Insert T_i into EDF List
   Else
   Insert T_i into AEDF List
   End If
2. End For
3. For all submitted tasks in the EDF List; T_i
   Schedule Task using EDF
4. End For
5. For all submitted tasks in the AEDF List; T_i
6. Schedule T_i with Min (ET_i)
   Calculate NT = ET_i + ST
   If $(T_{(i+1)}ET_{(i+1)}) + NT \leq DT_{(i+1)}$
   Schedule task T_i in AEDF List
   Else
   Schedule $T_{(i+1)}$
7. End For

Then the improvement operator check that the results are not improving now then perturbation operator will select another algorithm from the pool and take the best optimized solution from the algorithm running previously as an initial population which will again optimized with the algorithm selected from the pool by perturbation operator. The same process is repeated until maximum iteration reached. The existing algorithm is only scheduling the tasks and optimizing the solution based on the operators. But the concept of load is taken into consideration in this technique which will more optimize the solution and reduce the makespan time.

The proposed HHS-ALBA algorithm uses the concept of load balancing by the solution identified with the hyper heuristic algorithm to further optimize the solution. Load balancing technique ensures the system load balancing by computing the load on each VM and then relocating the task according to load on each machine and the deadlines of the tasks. The idea of adaptive load balancing is merged with the hyper heuristic concept i.e. the solutions generated from the proposed algorithm are more optimized and improved. For this implementation, each and every low-level scheduling algorithm also contain the adaptive load balancing in their solution which will further utilized in the hyper technique to make the solutions more optimized with the functionality of improvement operator of hyper heuristic algorithm.

IV. RRSULT AND DISCUSSION

In this section, evaluate the performance of the proposed HHS-ALBA algorithm based on deadline. Initially, we discuss experimental setup and analyses the computational results of proposed algorithm, and compare with existing HHSA, the heuristics algorithms are ACO, PSO GA; and traditional algorithms FCFS and SJF.

A. Simulation Setup

Simulations are conducted to estimate the performance results using Java (Jdk1.7) with CloudSim [30] Simulator. Simulation measure and test the results of the developed HHS-ALB algorithm at different performance metrics such as makespan, degree of imbalance, total processing cost and total processing time. The experiments were performed on PC with Windows 7 Ultimate (64- bits), core i7 processor, 3.40 GHz with memory of 16 GB. CloudSim toolkit enable us to model basic component of Cloud like data centers, hosts, VMs, broker and policies for resource utilization. All the experiments are performed by using 50 VMs, 15 host machines within a data center.

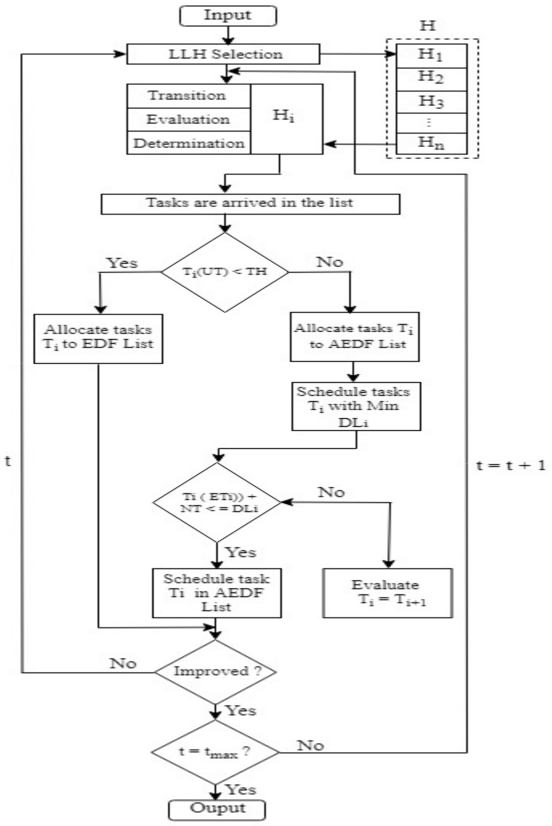

Fig.3. Workflow diagram of HHS-ALBA algorithm

B. Experimental results

The basic concept of the proposed methodology is to merge the concept of two important algorithms HHSA and ALB; this hybrid approach ensures the system load by calculating load and computational capabilities on each VM based on deadline better from existing scheduling algorithms. An experimental study is undertaken to evaluate the behavior of the proposed HHS-ALBA with existing HHSA, the heuristics algorithms are ACO, PSO, GA and traditional algorithms FCFS and SJF. The proposed algorithm is evaluated in term of makespan time, degree of imbalance, total processing time and total processing cost. The proposed strategy is evaluated using different scenarios approximate more than 50 times on different number of tasks range from 100 to 500 with random length cloudlet are assigned to fixed number of VMs (50 VMs). The maximum number of iteration (100) of LLH for each task range from 100 to 500. Configuration details of the CloudSim simulator are given in Table 2.

Table 2.

SIMULATION ENVIRONMENT PARAMETERS

Parameters	Values
Simulator version	CloudSim3.0.2
Number of datacentre	1-15
Number of host in datacentre	30
VmScheduler	TimeShared
Number of VMs	50 heterogeneous VMs
Computing power	100-4000 MIPS
Number of PE per VM	1-4
VM Ram	512 MB
CloudletScheduler	TimeShared
cloudlets/task's range	100-500
Length of cloudlet	1000-6000
Iteration of LLH	100

As performance analysis graph is depicted in Fig. 4. The makespan results by changing number of cloudlet and fixed number of VMs. The proposed HHS-ALBA algorithm showing much less than compared with existing HHSA, the heuristics algorithms are ACO, PSO, GA and traditional algorithms FCFS and SJF. GA and HHSA are showing better performance than ACO, PSO, FCFS and SJF. It is clear that HHS-ALBA makespan time reducing high performance than other existing algorithm. As shown in Fig. 4, the proposed HHS-ALBA algorithm get improvement of 57 % from HHSA , 89% from SJF, 90% from FCFS, 83% from GA, 96% from PSO and 94% from ACO on makespan for small number of tasks as 100 and number of VMs i.e. 50.

The case when the number of tasks is increased to 500, HHS-ALBA achieved gain of 9% from HHSA, 71% from SJF, 83% from

FCFS, 52% from GA, 94% from PSO and 84% from ACO. HHS-ALBA outperformance on the makespan time as compared to other strategies with load balancing.

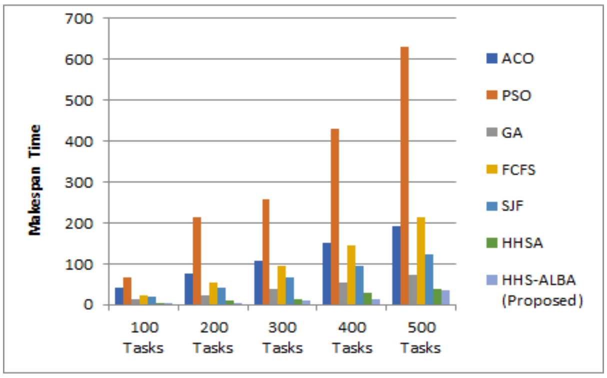

Fig.4.Comparison analysis of makespan time of tasks (100 to 500) and VMs (50)

As shown in Fig. 5, HHS-ALBA exhibits the best performance on the degree of imbalance as compared to other strategies with load balancing. The proposed HHS-ALBA algorithm get improvement of 17 % from HHSA , 18% from SJF, 19% from FCFS, 16% from GA, 19% from PSO and 17% from ACO on degree of imbalance for small number of tasks as 100 and number of VMs i.e. 50.

The case when the number of tasks is increased to 500, HHS-ALBA achieved gain of 0.6% from HHSA, 2% from SJF, 2% from FCFS, 1% from GA, 1% from PSO and 2% from ACO.

As shown in Fig. 6, HHS-ALBA exhibits the best performance on the total processing cost as compared to other strategies with load balancing.

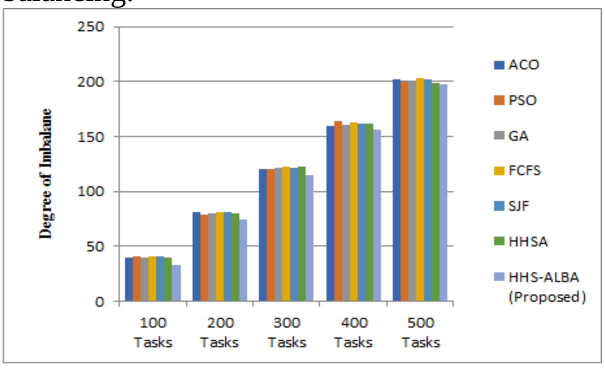

Fig.5. Comparison analysis of degree of imbalance of tasks (100 to 500) and VMs (50)

The proposed HHS-ALBA algorithm get good improvement of 68 % from HHSA , 73% from SJF, 73% from FCFS, 67% from GA, 74% from PSO and 73% from ACO on total processing cost for small number of tasks as 100 and number of VMs i.e. 50.

The case when the number of tasks is increased to 500 and number of VMs i.e. 50, HHS-ALBA achieved gain of 10% from HHSA , 25% from SJF, 25% from FCFS, 11% from GA, 24% from PSO and 25% from ACO.

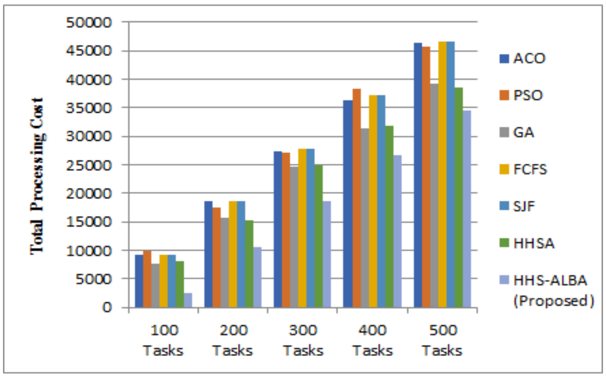

Fig.6. Comparison analysis of total processing cost of tasks (100 to 500) and VMs (50)

As shown in Fig. 7, the proposed HHS-ALBA algorithm get improvement of 2% from HHSA , 70% from SJF, 75% from FCFS, 45% from GA, 16% from PSO and 71% from ACO on total processing time for small number of tasks as 100 and number of VMs i.e 50.

The case when the number of tasks is increased to 500, HHS-ALBA achieved gain of 21% from HHSA, 67% from SJF, 82% from FCFS, 59% from GA, 54% from PSO and 68% from ACO. HHS-ALBA outperforms on the total processing time as compared to other strategies with load balancing.

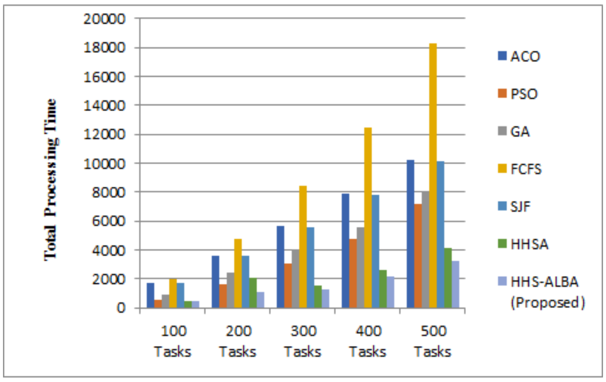

Fig.7. Comparison analysis of total processing time of tasks (100 to 500) and VMs (50)

V. Conclusion

In this paper, an optimized hyper-heuristics scheduling using adaptive load balancing algorithm (HHS-ALBA) based on deadline in cloud computing has been developed and implemented. The concept of adaptive load balancing is merged with the hyper heuristic concept i.e. the solutions generated from the proposed algorithm are more optimized and improved. The performance of HHS-ALBA algorithm is evaluated and tested at CloudSim simulator and compared with existing HHSA, the heuristics algorithms are ACO, PSO, GA and traditional algorithms FCFS and SJF. This paper compares the makespan, degree of imbalance, total processing cost and total processing time of each algorithm applicable for cloud computing. The simulation analysis was conducted to distinguish the effect by changing the number of cloudlets from 100 to 500 while fixed number of VMs (50). The simulation result shows that the performance of HHS-ALBA is best on all performance metrics.

As a future work, the proposed HHS-ALBA algorithm can be implemented and designed for solving workflow scheduling issue in cloud computing and real world environment.

References

[1] R. A. Haidri, C.P. Kattib and P. C. Saxena, "Capacity based deadline aware dynamic load balancing (CPDALB) model in cloud computing environment", *International Journal of Computers and Application*,. 2019.

[2] G. Kaur and S. Kaur, "Improved Hyper-Heuristic Scheduling with Load-Balancing and RASA for Cloud Computing Systems", *International Journal of Grid and Distributed Computing*, 2016.

[3] M. O. Ahmad and R. Z. Khan, "The Cloud Computing: A Systematic Review", *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, 2015.

[4] C. W. Tsai, W. C. Huang, M .H. Chiang, et al., "A Hyper-Heuristic Scheduling Algorithm for Cloud", *IEEE Transactions on Cloud Computing*, 2014.

[5] L. F. Bittencourt, A. Goldman and M. Edmundo, "Scheduling in distributed systems: A cloud computing perspective, Computer Science Review", 2018.

[6] S. Yin, P. Ke and L. Tao, "An improved genetic algorithm for task scheduling in cloud computing", *13th IEEE Conference on Industrial Electronics and Applications (ICIEA); Wuhan*, 2018. doi: 10.1109/ICIEA.2018.8397773.

[7] R. N. Calheiros and R. Buyya, "Cost-Effective Provisioning and Scheduling of Deadline-Constrained Applications in Hybrid Clouds" *In: Wang X.S., Cruz I., Delis A., Huang G. (eds) Web Information Systems Engineering – WISE.. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, , 2012.

[8] Y. Zhao, R. Calheiros and G, Gange, "SLA-Based Profit Optimization Resource Scheduling for Big Data Analytics-as-a-Service Platforms in Cloud Computing Environments", *in IEEE Transactions on Cloud Computing*,; doi: 10.1109/TCC.2018.2889956.

[9] M. A. Rodriguez and, R. Buyya, "A taxonomy and survey on scheduling algorithms for scientist workflows in IaaS cloud computing environments" *Concurrency and Computation: Practice and Experience*, 2016.

[10] F. Zhang, J. Cao, K. Li and S. U. Khan, "Multi-objective scheduling of many tasks in cloud platforms", *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2013.09.006. 2014

[11] R. Z Khan and M.O. Ahmad, "A survey on load balancing algorithms in cloud

computing" *International Journal Autonomic Computing*, 2017.

[12] M. D. Vigneshwaran and A. P. Umamakeswari "An Efficient Hyper-Heuristic Scheduling Algorithm For Cloud", *International Journal of Pure and Applied Mathematics*, 2017.

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization", *Proceedings of ICNN'95 International Conference on Neural Networks*, Perth, WA, Australia.1942-1948, 1995.

[14] M. Dorigo, and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling\nsalesman problem", *IEEE Transactions on Evolutionary Computation, 1997.* http://doi.org/10.1109/4235.585892.

[15] R. Aron, I. Chana and A. Abraham (2015). "A hyper-heuristic approach for resource provisioning-based scheduling in grid environment", *Journal of Super Computing*, 2015. DOI 10.1007/s11227-014-1373-9.

[16] Cr. Mateosa, E. Pacinib and C. G. Garinob, "An ACO-inspired Algorithm for Minimizing Weighted Flowtime in Cloud-based Parameter Sweep Experiments", *Advances in Engineering Software*, 2013.

[17] R. Priyanka, P. Priyadharsini and M. Nakkeeran, "Fault Tolerant Based Hyper-Heuristic Algorithm For Task Scheduling In Cloud", *ARPN Journal of Engineering and Applied Sciences*, 2015.

[18] E. N. Alkhanak and S. P. Lee, "A hyper-heuristic cost optimisation approach for Scientific Workflow Scheduling in cloud computing", *Future Generation Computer Systems*, 2018. https://doi.org/10.1016/j.future.2018.03.055.

[19] S. Mohanty, P. K. Patra, Ray, M. et al., "A Novel Meta-Heuristic Approach for Load Balancing in Cloud Computing", *International Journal of Knowledge-Based Organizations*, 2018.

[20] S. RamKumar, V. Vaithiyanathan and V. Lavanya, "Towards Efficient Load Balancing and Green it Mechanisms in Cloud Environment", *World Appl. Sci. J., 29 (Data Mining and Soft Computing Techniques*, 2014.

[21] Jain, A. and Upadhyay, A. Cloud Scheduling Using Improved Hyper Heuristic Framework. International Conference on Advanced Computing Networking and Informatics, Advances in Intelligent Systems and Computing; pp. 127-133.

[22] S. H. H. Madni, M. S. Abd Latiff, M. Abdullahi, et al., "Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment", *PLoS ONE*. 2017.

[23] N. Xoxa, M. Zotaj, I. Tafa et al., Simulation of First Come First Served (FCFS) and Shortest Job First (SJF) Algorithms", *IJCSN - International Journal of Computer Science and Network.*

[24] S. Sindhu, "Task Scheduling In Cloud Computing. International Journal of Advanced Research in Computer Engineering & Technology", 2015.

[25] E. K. Burke, M. Gendreau, M. Hyde, et al., "Hyper-heuristics. A survey of the state of the art", Journal of the Operational Research Society, 2013.

[26] A. Pinjari and B. R. Mandre, "Cloud-Scheduling Algorithm using Hyper heuristics Approach: Study and Implementation", *International Journal of Computer Systems*, 2015.

[27] F. Ebadifard and S. M. Babamir, "A PSO-based task scheduling algorithm improved using a load balancing technique for the cloud computing environment", *Concurrency and Computation: Practice and Experience*, 2018.

[28] M. A. Alworafi, A. Dhari, S. A El-Booz, et al. "An Enhanced Task Scheduling in Cloud Computing Based on Hybrid Approach", *In: Nagabhushan P., Guru D., Shekar B., Kumar Y. (eds) Data Analytics and Learning. Lecture Notes in Networks and Systems*, 2019.

[29] M. O. Ahmad and R. Z. Khan, “Load Balancing Tools and Techniques in Cloud Computing: A Systematic Review”, *In: Bhatia S., Mishra K., Tiwari S., Singh V. (eds) Advances in Computer and Computational Sciences. Advances in Intelligent Systems and Computing*. 554. Springer,Singapore, 2017.

[30] N. Rodrigo, R. N. Calheiros, A. Beloglazov, et al., “CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms”, *Software: Practice and Experience (SPE)*, 2011.

.