International Journal of Recent Trends in Engineering & Research

ISSN (ONLINE) : 2455 – 1457 IMPACT FACTOR : 4.101

HABC: Hybrid Optimization Method Based Resource Allocation for Improving the Energy Efficiency in Green Cloud

Mrs.J.K.Jeevitha[1], Dr.G.Athisha[2]

[1]*Assistant ProfessorDepartment of Information Technology,*
PSNA College of Engineering and Technology,India
[2]*Professor & Head,Department of Electronics and Communication Engineering*
PSNA College of Engineering and Technology,India

Abstract: Cloud computing is one of the most important and plays a significant role in computing world. It makes all the cloud users happy in providing various kinds of services under SAAS, PAAS, and IAAS. Due to the increased number of customer and customer demands on services the number of services deployed in the cloud is also getting increased. Services accessed by the customers by Pay-N-Use. The resources allocated related to the user's request virtually by considering the numbers of cloud users and resources. But actual resource allocation in heterogeneous cloud environments is complicated, and the accuracy is less, so that, these problems are leads to the high level of energy consumptions. Various methods, potential experiments, and algorithms are proposed in the earlier research works to increase the efficiency of the resource allocation with QoS. This paper discourse the problem of optimum resource allocation to a user with improved Quality of Service (QoS) using HABC (Hybrid Artificial Bee Colony) method and identify the better influencing parameters using Design of Experiment (DoE) tool. Initially, the user requests and the resources are validated, then map the scheduled resources and finally choose the best and allocate the right resources using HABC algorithm. The simulation is conducted using CloudSim tool. The obtained results verified and the performance is evaluated by comparing the results to different evolutionary algorithms. The parameters such as energy, time, cost, valid request, optimum number resources, accurate mapping, and allocation considered as the QoS parameters which are calculated from the simulation and presented here.

Keywords: Cloud Computing, Resource Allocation, Resource Mapping, Request Validation, Scheduling, Quality of Service.

Nomenclature

QoS: Quality of Service
DoE: Design of Experiment
SaaS: Software as a Service
PaaS: Platform as a Service

I. INTRODUCTION

In distributed computing, several computing methods were proposed and utilized from desktop computing to cloud computing. Different things are given as a service to different people in the cloud. All the resources can be shared with any person from anywhere at any time based on pay-n-use method [1]. The virtualization concept and the scalability problem of the cloud are represented by different definitions and experiments conducted by the authors in [2] and [3] respectively. Various applications utilize cloud-based services in multiple forms like software hardware and data.

Three different users interconnected in the cloud such as service provider, customer and end users. Cloud computing provides a un-countable service to cloud customers regarding software, hardware and data centers. Major IT companies such as Google, Amazon, IBM and, Microsoft have their cloud data centers around the world to support cloud services. Due to the un-countable services and the customers, maintaining the response and allocating the resources become a mesh. The authors in [4, 5] discussed the job scheduling and the authors in [6, 7] presented the resource allocation policies for effective resource utilization and efficient service provision to the customers. The cloud subscribers determine the Resources needed by the user through the Service Level Agreement (SLA) which are provided by the cloud data centers and tend to satisfy the Quality of Service (QoS). SLA provides a two-sided contract between the cloud provider and the user; it also determines the level of performance, price, the content of the service supplied and the penalties for not providing the services to the user. Any failure to observe of the QoS leads to SLA violation, and consequently, and hence penalty must be paid by service providers [8]. The speedy growth of cloud services and their interrelated technologies, cloud infrastructures have become more problematic and more convoluted. Due to this Resource management becomes a significant issue in a current cloud environment and precisely affects the development of cloud services. Modern data centers provide a high level of performance and optimization; yet, a new concern in energy consumption has occurred.

A model of the cloud architecture with the virtual machine and physical machine sharing process illustrated in Figure-1. A scheduler and a monitor used for scheduling and monitoring the resources (VM, PM) in the cloud. There is N number of resources interconnected with the K physical machine used for computing cloud data in host computing system. These are the resource going to allocate to run applications are various time on "on-demand." In this paper, it assumed that the cloud is centralized and hosted in the datacenter where it comprises of a more number of heterogeneous servers. Different jobs assign all the servers. The virtualization procedure permits the cloud can create any number of a virtual machine for any available physical machine. Hence, assigning a task is more flexible under any server. All the servers can save their energy in an idle state where it performs the maintenance and has all the other sub-systems are ready while it waits for the task to attain. When a task comes inside, the allocated VM executes the task, and the host has to spend some more additional energy related to the number of resources required by the task, and it called as resource utilization in workload model. Cloud is distributed across in more geographical locations and mainly connected to a specific physical location. It assumed that the all the resources are homogeneously associated with their capacity and their computing capability. It can conclude by applying virtualization methods [9]. A message is transmitted from one resource to another resource when a task executed on the destination resource, which is possible in many systems. The lower and upper limit of the energy consumption of the server in cloud computing is represented as a pick-load-state and idle-state.

II. RELATED WORKS

According to the Environmental Protection Agency (EPA), the energy consumption of data center would rise from 2006 (61 billion kWh) to 2011 [12]. In 2009, data centers accounted for 2% of worldwide electricity consumption with an economic impact of US $30 billion [13]. Gartner Group forecasted data center hardware expenditure for 2012 to be as the US $106.4 billion, a 12.7% increase from 2011, whereas cloud computing revenue is predicted to jump from the US $163 billion in 2011 to the US $240 billion in 2016 [14]. IT sector contributes indirectly to carbon dioxide emission as a significant amount of carbon is used in the electricity production process, and they emit dioxide and causes greenhouse gases [15]. The IT sector was responsible for 2% of carbon dioxide worldwide in 2005, a figure that is estimated to grow by 6% per year [12]. Aggressive energy-efficiency measures for all devices inside the data center can reduce 80% of energy costs and 47 million metric tons of carbon dioxide emissions [12]. Power usage effectiveness (PUE) is the ratio of total data center energy usage to IT equipment, energy usage, which helps in measuring the data center efficiency [12]. The average PUE value of data centers in a 2005 survey was found to be 2 [16]. It estimated that the average PUE value would fall to 1.9 by 2011, although aggressive energy efficiency measures can result in typical PUE value of 1.2 [12]. The higher PUE value indicates that most of data center energy consumed in cooling measures instead of computing. An energy-efficient data center with a lower PUE index can lead to several benefits such as:

a) Reduced energy consumption, hence, lesser operational expenses;
b) Lesser greenhouse gas emissions; and
c) Lesser device costs.

Data center operators are also interested in the total cost of ownership: Which is the sum of CAPitalEXpenses (CAPEX) required in setting up the data center and the OPerational EXpenses (OPEX) needed in running the data center [12].

While considering task scheduling and resource allocation, various earlier works [17 -22] discussed the same. Some of them focused on reducing the task completion time and cost of the resource utilization. The author in [23] stated that scheduling is a systematic task in built-in supercomputers. Currently, resource allocation becomes emerging research. The author in [24] focused on Service Level Agreement for service providing for SaaS applications. In the SLA agreement, it mainly concentrated on reducing the time and cost of resource allocation. In [25-26], it said that the service providers have to allocate dedicated VMs for all the users, then it reduce the time and cost for all the responses. Similarly, various scheduling algorithms proposed for cloud computing [27-29]. The algorithms discussed in [23, 17] suggested for cloud scheduling based on their suitability. This paper also utilizes the scheduling method to reduce the time and cost and to optimize the resource allocation. Due to the speedy growth of cloud users and resources, recent research works are focused on optimization methods for resource scheduling and resource allocation. An optimal task scheduling and resource allocation is obtained by Particle Swarm Optimization (PSO) method which utilizes a fitness function [30]. The makespan reduced by increasing the fitness function and it also increases the execution capacity. In [32-33] it is proved from the experimental results that PSO can reduce the time and cost of the resource allocation. The author in [34] used Genetic Algorithm for scheduling and resource allocation and obtained better results.

From the above discussion, various earlier approaches focused on reducing the energy consumption using multiple strategies like resource allocation, task completion, proper resource utilization, SLA based resource allocation and optimized resource allocation. Hence it is identified that it is necessary for improving the energy efficiency by resource allocation.

III. ARTIFICIAL BEE COLONY

Artificial Bee Colony is one of the optimization algorithm referred from [35-37], used to select an optimal value from a mass of values. Foraging behavior of the honey bees used for optimization process. All the bees are involved in searching their foods. There are three different bees are considered here such as scout bees, onlooker bees, and employed bees. Among these bees, scout bees always search for bees in a random direction within the solution space. The employed bees found the location of the food and distribute the information to other onlooker bees. Finally, the fitness function calculated by the onlooker bees and searches the food. All the three types are bees seek their food in long distance and all directions. One they got the food the scout bees come back to original places and do "Waggle Dance". While dancing, the information about the food-source and an amount nectar to be spent for food source. At that nectar quality and the wastage of energy is evaluated. Then the onlooker bees select the best food source from the scout bees' information as the best food source.

ABC algorithm is best in searching the optimal solution, the searching time and the searching ability is little poor in ABC comparing with other optimization algorithms. Also, SA is an analogous method; it can obtain the best solution. Hence, the searching function of SA is integrated into ABC to increase the efficiency of the ABC to achieve the optimal solution efficiently and speedily in the search space. The best food source information is searched by SA randomly. By integrating two different optimization algorithms, the efficiency of the best resource selection is increased.

The entire process of **ABC** algorithm is given in the form of pseudo code, and it is presented in Figure-3 and Figure-4. There is N number of VMs considered in the cloud environment for assigning to various users. Different servers (s) are interconnected, and the incoming requests maintained by CMP (Cloud Management Policies) where it maintained the load in each **S** based on the present time. The entire scheduling process focused on the **VM** and **load**. According to the load and available **VM**, the scheduling approach works. This function reduces the time and makespan.
Let us assume, the N, VMs are

$$VM = \{VM_1, VM_2, \ldots, VM_N\}$$

$$i = \{1, 2, \ldots, N\} \; \forall \, i \in N$$

$$Task = \{task_1, task_2, \ldots, task_K\}$$

It is known that VMs are allocated parallel to the incoming requests without any collision. The complete process of VM scheduling and load balancing using ABC is illustrated in Figure-5.

Pseudo Code for Artificial Bee Colony

Initialization: Initialize SB as population P, EB, OLB, Food Sources, Fitness values;

- Generate population P (SB)
- N=0;
- repeat {
- EB search food sources calculate fitness and update fitness if the new fitness is better than the existing fitness values.
- From the EB, elect OLB to search food sources and calculate fitness values.
- Select the OLB which has best fitness values using SA(OLB)
- send SB into food sources to discover new food sources.
- N=N+1
- }until (R > N)

}

Figure-3: ABC Algorithm

Algorithm SA (OLB, EB) {

1. Let K== OLB
2. For I=1 to P
3. S ⟵ Best.OLB
4. Choose a random OLB, best.OLB=OLB
5. end i
6. return best.OLB
7. }

Figure-4: SA Algorithm

Numerical Illustration of HABC

1. **Initializing the Population**
   N number of Selected Bees (SB) = SB_1, SB_2, SB_3…. SB_N
2. **Fitness Evaluation**
   Using the following formula the fitness value of each bee is calculated as:

$$fit_{ij} = \frac{\sum_{i=1}^{n} t\_legnth_{ij}}{cap - of - VM_j\,(the\ cap\ is\ j)}$$

Where,

fit_{ij}	:	Fitness value
i	:	Population in VM_j.
VM_j	:	The capacity virtual machine with i^{th} bee.
t_legnth_{ij}	:	Length of the task which is submitted to VM_j

and the capacity is calculated according to:

$$capacity_j = pe_{num_j} \times pe_{mips_j} + vm_{bw_j}$$

Where,

$capacity_j$ denotes the capacity of VM_j

pe_{num_j} denotes the number of processors in VM_j

pe_{mips_j} denotes the millions of instruction per second over the processor in VM_j

vm_{bw_j} denotes the network bandwidth

3. **for searching the neighbors m sites are selected**
   The highest fitness value based neighbor is selected as SB among m VMs.
4. **Recruit Bees for Selected Sites**

The high amount of bees is deployed in the search space to increase the best solution. Among these bees, the fitness is computed as

$$fit_{ij} = \frac{\sum_{i=1}^{n} t\_legnth_{ij} + InputFile_{length}}{capa - of - VM_j\,(the\ capa\ \ is\ j)}$$

Where,

$InputFile_{length}$: Length calculated before execution of a task.

5. Select the best fitness bees from each set of population and assign a task to each VM_j.

Based on the absolute fitness values the best bees are selected in all the round of operation and allocate the VM using first in first out priority (best fitness, time).

IV. EXPERIMENTAL RESULT AND DISCUSSION

The proposed HABC algorithm implemented in CloudSim software and the obtained results verified. The algorithm implemented and executed so many times, with various numbers of users, data centers, tasks, and VMs. It is a simulation tool can do simulations for cloud applications based experiments. For example, a set of sample parameters used in the simulation is given in Table-1. The parameters used for optimization in HABC are given in Table-2. To execute the HABC, a set of VM, DC, PC, Laptops, and Servers assigned in the cloud, and they are all interconnected by wire or wireless mediums.

In this simulation, any user can send a request to cloud regarding any resources. This request is passed immediately to the closest server presented in the cloud. Whenever a request comes, it always directed to the server with the help of http-address. During the request and resource allocation, the HABC maps the request and the available resources using the optimization ability.

After resource allocation, the state of the resource is changed into "busy," after completion of the request tasks; the state of the resource is changed into "available/free/idle." While searching and mapping the request and the resources, the HABC looks only for the resource's' whose state is "free," the distance is decidedly less comparing with the other resources and time taken for the entire resource allocation, and task completion process is very less.

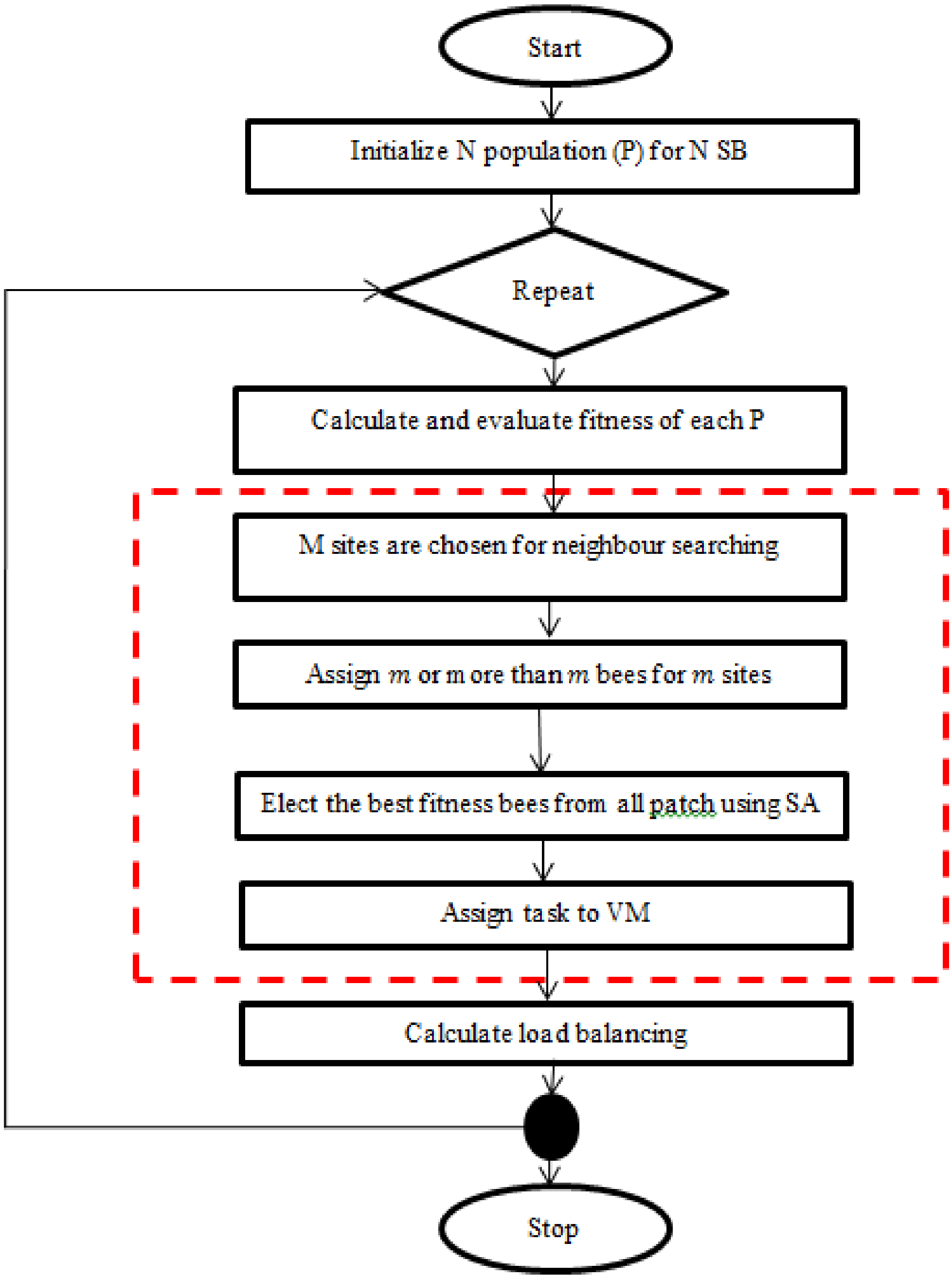

Figure-5: HABC Based Scheduling and Load Balancing

Table-1: Simulation Parameters Used in CloudSim

Device Type	Parameter	Values Assumed
Data Center	Number of Data Centers	10
	Number of Hosts	5
	Manager Type	Space Shared/Time Shared
	Number of PEs per Host	2 to 4
	Bandwidth	2000
	Host Memory	2048 to 10240
	Cost of the Datacenter	10
Virtual Machine (VM)	Number of VMs	30 to 210
	MIPS[b] of PE[c]	1000 to 2000
	Number of PE/VM	1
	VM-Memory size (MB)	512 to 2048
	Bandwidth	1000
	Manager Type	Time Shared
Task	Number of Tasks	100 to 700
	Length of Task (MI[d])	5000 to 20000
	Number of PE per requirement	1
	Manager Type	Space shared

Where,

MB denotes Megabyte

PE denotes Processing Element

MIPS denotes (Million Instructions per Second) is a measure of the processing speed of the computer

MI denotes Million Instruction.

Table-2: Parameters Used in ABC Algorithm

Symbol	Parameter	Values
n	Total number of **SB**	**1000**
m	Selected number of sites from **n** sited visited	**5**
e	Number of best sites taken from m sites	**1**
nep	Number of bees recruited from **e** sites	**800**
nsp	Number of best bees elected for other sites $(m - e)$	**200**

Initially, the number of optimized VMs selected from the total number of VM initialized is computed the results verified. The calculated results regarding optimal resource allocation are illustrated in Figure-6. The efficiency of the HABC is verified by changing the number of requests and VMs allocated. From the simulation, the obtained results regarding optimal resource allocation for the valid number of requests is shown in Figure-7. From the results, it is identified that when the deployed number of VM and requests are increased then the optimal allocation becomes faster and smooth. Obtaining available optimized resource is easy.

Resources are the VMs, will be allocated only for the valid requests sent by the authorized users. It also increases the cloud security. It is validated by verifying the user information previously stored during user registration. After the requests and the VMs are validated all the requests, and the VMs are analyzed, mapped and select the best VM for appropriate requests. This process is experimented by changing the number of VMs and requests and executed many times. In each round of operation, it is verified that how many optimized VMs are available for requests. The number of optimized VM selected is given in Figure-6. When the number of VM is increased then the optimal number of VM is also increased, and it is obtained by HABC speedily. Based on the increased number of VMs the number of optimal VMs is also increased, and it proved in the experiment. Unless otherwise, the user request of VM is valid or available they cannot match with one another.

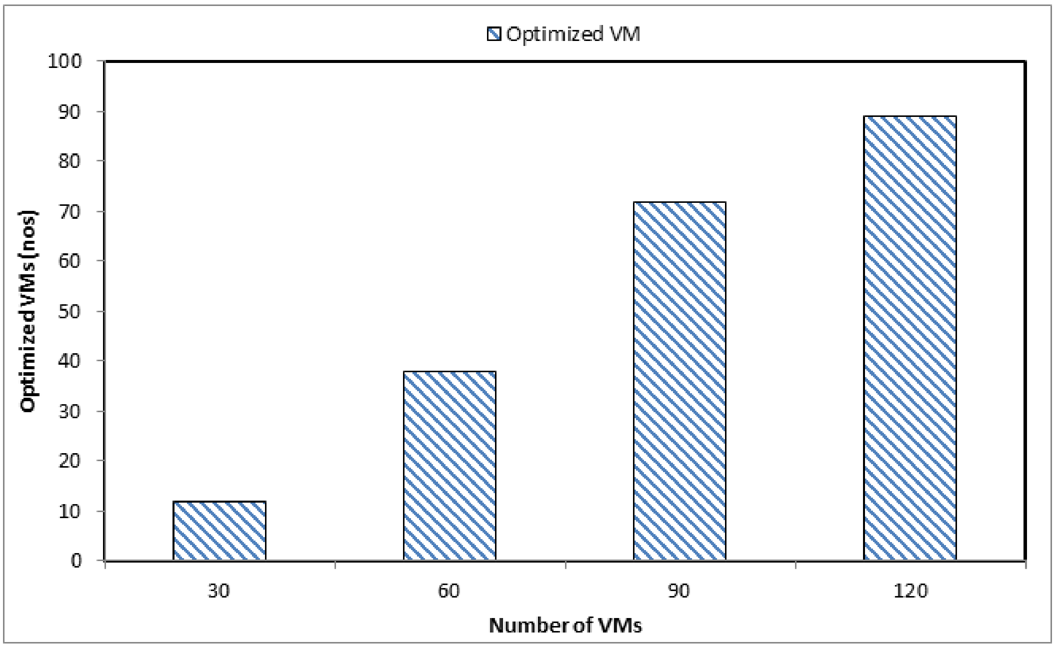

Figure-6: Number of VM versus Optimized

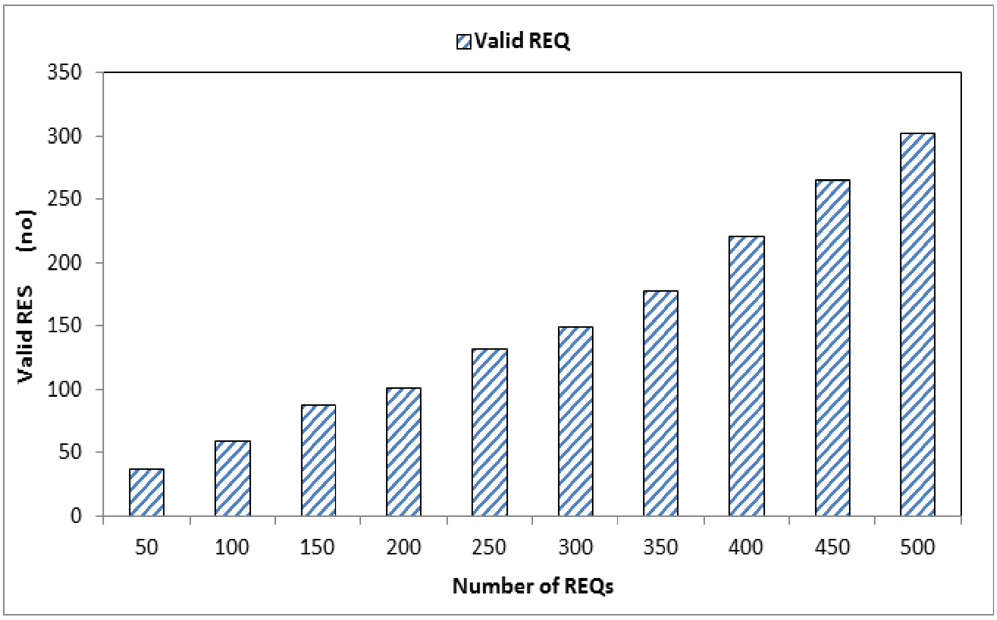

Figure-7: Number of Incoming Requests versus Valid Requests

From the various rounds of the experiment, it is noticed that the HABC is efficient regarding optimization based resource allocation. Hence it is necessary to evaluate the performance of the proposed HABC. To do that, obtained results of the proposed approach are compared with the other traditional existing methods such as FCFS (First Come First Served), SJF (Shortest Job First) and Longest Job First (LJF). In FCFS, the requests processing is done based on the queue model, in SJF the requests processing is done based on the size and distance, whereas in LJF the requests processing is done based on the big size of requests are processed first. Here all the three models are experimented along with the HABC and the obtained results are compared with one another.

According to the changed number of requests and the VMs, the average makespan is also computed and compared. The compared results among the various traditional scheduling algorithms with the HABC are given in Figure-8. From the comparison results, it is noticed that the performance of HABC is better than the other scheduling approaches.

Also, the average makespan of the scheduling approaches is calculated based on the number of VMs. Here the ABC is integrated with the existing FCFS, LJF and the average makespan is computed. Comparing with the obtained results the lesser average makespan is obtained using HABC-SJF than the other scheduling approaches. Also, it is noticed that when the number of VMs increased then the average makespan is decreased gradually and it is shown in Figure-9. From the entire performance evaluation, the ABC integrated scheduling approaches provides a better result than the other scheduling approaches. But, combining HABC with different scheduling approaches HABC based scheduling outperforms regarding average makespan.

It is experimented and verified under various parameters and compared with the earlier research works whose focus is also same to evaluate the performance of HABC. The entire performance of HABC is computed regarding total energy, energy consumed in server, mean value of the response time, number of completed tasks and number of uncompleted tasks. The obtained results are given in Figure-10. Comparing with the other existing approaches, HABC is better than the different methods.

From the results and discussion, it is identified and concluded that the proposed HABC is better and suitable for resource scheduling and resource allocation efficiently.

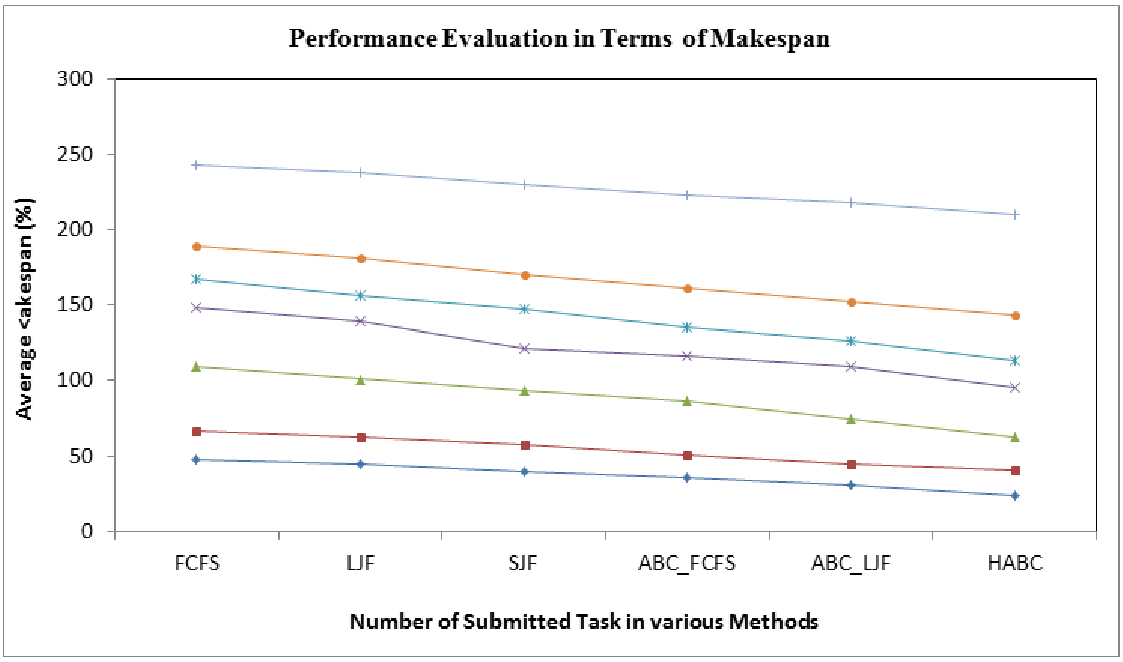

Figure-8 Comparison Of various Methods regarding Makespan versus Tasks

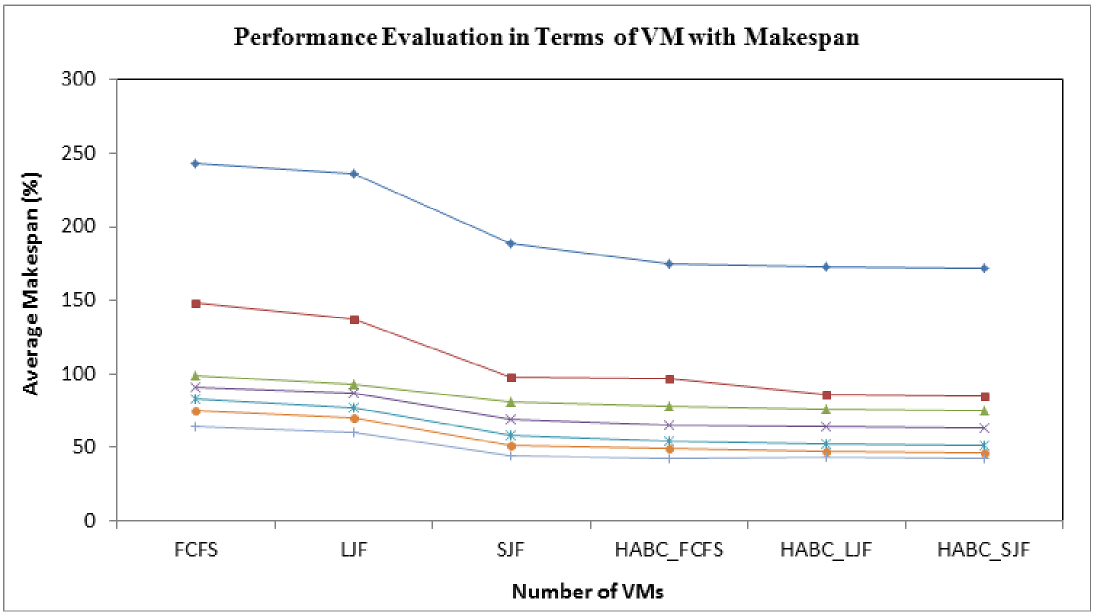

Figure-9: Comparison Of various Methods regarding Makespan versus VMs

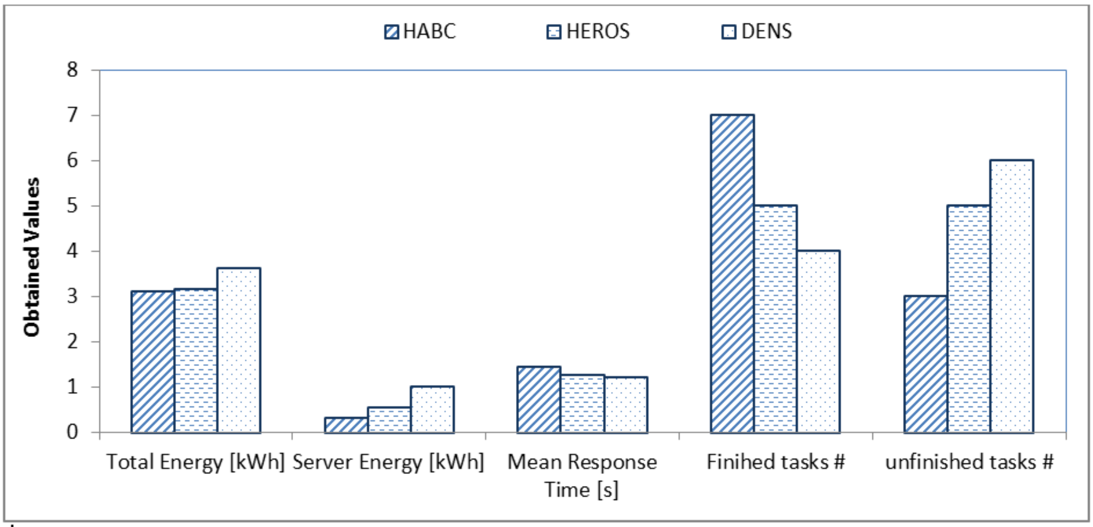

Figure-10: Performance Evaluation of HABC

V. CONCLUSION

The primary objective of this paper is to do optimum resource allocation using optimized scheduling, request validation, and VM validation. Artificial Bee Colony algorithm is used to do optimization here. The algorithm is experimented using CloudSim tool, and the results are verified. From the obtained results the incoming requests are validated, the appropriate VMs (resources) are optimized using ABC and makespan are obtained. The results illustrated that the proposed ABC algorithm is more efficient and suitable for cloud computing environment regarding reduced makespan for increased resources in a cloud environment. The performance is also evaluated by comparing among various scheduling approach, and it is concluded that ABC_LJF is proved itself is more efficient for maintaining system stability and efficiently scheduling. In future, the ABC_LJF is compared with the other optimization algorithms, and the performance is evaluated.

REFERENCES

I. M. Armbrust, A. Fox, R. Grifffth, A.D. Joseph, and R. Katz, ``Above the clouds: A Berkeley view of cloud computing," UC Berkeley Reliable Adaptive Distrib. Syst. Lab., Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, 2009.

II. Aral A., Ovatman T., Improving Resource Utilization in Cloud Environments using Application Placement Heuristics, CLOSER (2014), 527-534.

III. Mezmaz M., Melab N., A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems, Journal of Parallel and Distributed Computing, Elsevier (2011), 1497-1508.

IV. Wang W., Liang B., Li B., Multi-resource fair allocation in heterogeneous cloud computing systems, IEEE Transactions on Parallel and Distributed Systems 26(10) (2015), 2822-2835.

V. Viswanathan H., Lee E.K., Rodero I., Pompili D., Parashar M., Gamell M., Energy-aware application-centric VM allocation for HPC workloads, IEEE International Symposium on Parallel and Distributed Processing Workshops and Ph.D. Forum (IPDPSW) (2011), 890-897.

VI. Abdullah M., Othman M., Ibrahim H., Subramaniam S., Optimal workload allocation model for scheduling divisible data grid applications, Future Generation Computer Systems 26(7) (2010), 971-978.

VII. Yu J., Buyya R., Scheduling scientific workflow applications with a deadline and budget constraints using genetic algorithms, Scientific Programming 14(3-4) (2006), 217-230.

VIII. Koomey, Growth in Data Center Electricity Use 2005 to 2010. Vol. 9. El Dorado Hills, CA, USA: Analytical, 2011.

IX. Lee, Y. C. and Zomaya, A. Y. (2012). Energy efficient utilization of resources in cloud computing systems. The Journal of Supercomputing, 60(2):268-280.

X. Chedid, W., Yu, C., and Lee, B. (2005). Power analysis and optimization techniques for energy efficient computer systems. Advances in Computers, 63:129-164.

XI. Montgomery D.C., Design, and Analysis of Experiments, seven editions Wiley, New Delhi (2012). Octavio Gutierrez-Garcia.

XII. J. Koomey, Growth in Data Center Electricity Use 2005 to 2010. Oakland, CA, USA: Analytics Press, Aug. 2011.

XIII. G. Group, Forecast: Data centers, worldwide, 2010–2015, Accessed March 2013. [Online]. Available: http://www.gartner.com.

XIV. S. Greenberg, E. Mills, B. Tschudi, P. Rumsey, and B. Myatt, “Best practices for data centers: Lessons learned from benchmarking 22data centers,” in Proc. ACEEE Summer Study Energy Efficiency Build. Asilomar, Pacific Grove, CA, USA, Aug. 2006, vol. 3, pp. 76–87.

XV. G. Meijer, “Cooling energy-hungry data centers,” Science, vol. 328, no. 5976, pp. 318–319, Apr. 2010.

XVI. D. Kliazovich, P. Bouvry, and S. Khan, “DENS: Data center energy efficient network-aware scheduling,” in Proc. IEEE/ACM GreenCom, Dec. 2010, pp. 69–75.

XVII. Malathy G., Somasundaram R., Vidhya G., A survey on workflow scheduling algorithms and map reduce method in a cloud environment, International Journal of Communications and Engineering 6(6) (2012), 1-6.

XVIII. Liu M., Li T., Optimizing virtual machine consolidation performance on NUMA server architecture for cloud workloads, ACM/IEEE 41st International Symposium on Computer Architecture (2014), 325-336.

XIX. Yu J., Buyya R., Scheduling scientific workflow applications with a deadline and budget constraints using genetic algorithms, Scientific Programming 14(3-4) (2006), 217-230.

XX. Mezmaz M., Melab N., A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems, Journal of Parallel and Distributed Computing, Elsevier (2011), 1497-1508.

XXI. Gutierrez-Garcia J.O., Sim K.M., A family of heuristics for agent-based elastic cloud bag-of-tasks concurrent scheduling, Future Generation Computer Systems 29(7) (2013), 1682-1699.

XXII. Wei Wang, the Bayesian cognitive model in scheduling algorithms for data-intensive computing, Journal of Grid Computing (2012), 173-184.

XXIII. Yun-Han Lee, Improving job scheduling algorithms in a grid environment, Future Generation Computer Systems (2011), 991-998.

XXIV. Liu K., Jin H., Chen J., Liu X., Yuan D., Yang Y., A compromised time- cost scheduling algorithm in swindew-c, for instance, intensive cost-constrained workflows on a cloud computing platform, The International Journal of High-Performance Computing Applications 24(4) (2010) 445-456.

XXV. Delias P., Doulamis A., Doulamis N., Matsatsinis N., Optimizing resource conflicts in workflow management systems. IEEE Transactions on Knowledge and Data Engineering 23(3) (2011), 417-432.

XXVI. Wu L., Garg S.K., Versteeg S., Buyya R., SLA-based resource provisioning for hosted software-as-a-service applications in cloud computing environments, IEEE Transactions on services computing 7(3) (2014), 465-485.

XXVII. Abdullah M., Othman M., Ibrahim H., Subramaniam S., Optimal workload allocation model for scheduling divisible data grid applications, Future Generation Computer Systems 26(7) (2010), 971-978.

XXVIII. Amin Shokripour, Mohamed Othman, New method for scheduling heterogeneous multi-installment systems, Future Generation Computer Systems (2012), 1205-1216.

XXIX. Amid KhatibiBardsiri, Seyyed Mohsen Hashemi, A review of workflow scheduling in cloud computing environment, International Journal of Computer Science and Management Research (2012), 1-4.

XXX. Yang Z., Qin X., Li W., Yang Y., Optimized task scheduling and resource allocation in cloud computing using PSO based fitness function, Information Technology Journal 12(23) (2013).

XXXI. Xu M., Cui L., Wang H., Bi Y., A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing, IEEE International Symposium on Parallel and Distributed Processing with Applications (2009), 629-634.

XXXII. Hashemi A.B., Meybodi M.R., A note on the learning automata-based algorithms for adaptive parameter selection in PSO, Applied Soft Computing 11(1) (2011), 689-705.

XXXIII. Viswanathan H., Lee E.K., Rodero I., Pompili D., Parashar M., Gamell M., Energy-aware application-centric VM allocation for HPC workloads, IEEE International Symposium on Parallel and Distributed Processing Workshops and Ph.D. Forum (IPDPSW) (2011), 890-897.

XXXIV. Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., Vahdat, A.: Hedera: dynamic flow scheduling for data center networks. In: Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI '10), San Jose, CA, April (2010)

XXXV. Meng, X., Pappas, V., Zhang, L.: Improving the scalability of data center networks with traffic-aware virtual machine placement. In: IEEE INFOCOM, San Diego, California, March (2010)

XXXVI. Stage, A., Setzer, T.: Network-aware migration control and scheduling of differentiated virtual machine workloads. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges ofCloud Computing, InternationalConference on Software Engineering, May. IEEE Computer Society, Washington, DC (2009).

XXXVII. Rastkhadiv, F., Zamanifar, K.: Task scheduling based on load balancing using artificial bee colony in cloud computing environment. Int. J. Adv. Biotechnol. Res. 7(Special Issue-Number 5), 1058– 1069 (2016). ISSN 0976-2612, Online ISSN 2278-599X.