International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 7, Issue 4
April 2018

Performance Analysis of Min-Min, Max-Min and Artificial Bee Colony Load Balancing Algorithms in Cloud Computing.

Neha Thakkar[1], Dr. Rajender Nath[2]
[1]M.Tech Scholar, Professor[2]
[1,2]Department of Computer science and Applications,
Kurukshetra University, Kurukshetra, India

Abstract: *Load Balancing is an integrated aspect of Cloud Computing. In order to achieve optimal machine utilizations, tasks are transferred from overloaded machine to under-loaded machines. The load balancing algorithms are categorized into two categories static and dynamic. Min-Min and Max-Min algorithms are referred as the static algorithms, and Artificial Bee Colony as the part of (honey bee) referred as the dynamic load balancing algorithms. In this Paper, the implemented Min-Min, Max-Min and Artificial Bee Colonyload balancing algorithms are described and these algorithm are compared in term of their makespan, average resource utilizations'. The result of analysis indicated that the artificial bee colony dynamic load balancing algorithm performed better than the static min-min and max-min load balancing algorithms*

Keywords: Cloud Computing, Load Balance, Min-Min Algorithm, Artificial Bee Colony Algorithm, Max-Min Algorithms Make-span, Resource-utilizations.

I. INTRODUCTION

Cloud computing is one of the distributed computing paradigm which mainly focuses on providing everything as a service to the customers and it provides computational as well as storage resources to users. The processing units in cloud environments are called virtual machines (VMs).Scheduling of the customers tasks within the available resources is a challenging task. More than one task is assigned to one or more VMs that run the tasks simultaneously. This kind of environments should make sure that the loads are well balanced in all virtual machines.

Load balancing is the task of distribution of application tasks to different processors in an efficient manner to minimize program execution time. The implementation of load balancing can make cloud computing more effective and it also improves user satisfaction. Load balancing distributes workloads across multiple computing resources such as computers, a computer cluster, network links, central processing units or disk drives. The main aim of load balance to reduce the response time and optimize the resource use. A load balancing algorithm attempts to improve the response time of user's submitted applications by ensuring maximal utilization of available resources. The load balancing ensures that all the resources do the equal work in the cloud computing[1]. To achieve optimal machine utilization, tasks from overloaded virtual machines are transferred to the neighbor Virtual machine whose load value is below threshold.[2] Cloud computing provides much more effective computing by centralized memory, processing, storage and bandwidth. It should make sure that the tasks are not loaded heavily on one VM and also ensure that some VMs do not remains idle and/or under loaded [3]. The main objective of load balancing methods is to speed up the execution of applications on resources whose workload varies at run time in an unpredictable manner Figure 1 represent the load balancer.

The altering the idea of Domain Name system(DNS) record was the actual began of load balancing functionality. Before load balancing to achieve the goal of scalability and high availability of resource, the commercial users adopt DNS only[4].

This paper is divided into the following sections: section 2 describes the Min-Min, Max-Min and Artificial Bee colony load balancing algorithms, Section 3 provides experimental results and finally

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 7, Issue 4
April 2018

Section 4 gives the conclusion and future enhancement possibilities.

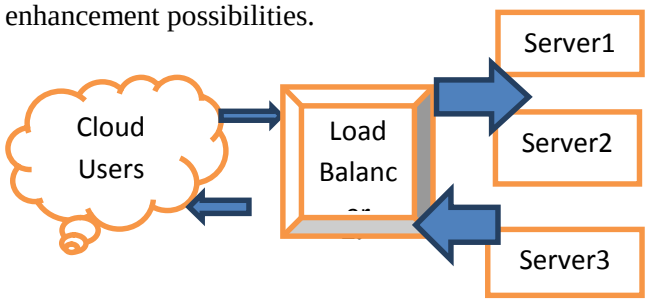

Fig 1. Load Balancer

II. LOAD BALANCING ALGORITHMS

Load balancing algorithms are broadly classified into two categories: static and dynamic load balancing algorithms.

Static algorithms: Static algorithms are those which divide the traffic equivalently between servers. There are the many different types of algorithm like min-min, max-min.

Dynamic algorithm: Dynamic algorithms are which search for the lightest server in the network and then designated appropriate weights on it. There are the many different types of algorithm like Particle Swarm Optimization(PSO), Ant Colony Optimizations(ACO), Honey bee Foraging Behavior.

In load balancing there are various type of strategies or algorithm[5] have been adopted such as Artificial Bee colony(ABC),Random Sampling etc. After reviewed the many static and dynamic algorithm we choose the three popularly used algorithms as Min-Min, Max-Min and Ant Bee colony (ABC),have been Presented.

A) Min-Min Algorithm

The Min-Min algorithm is simple and still basis of present cloud scheduling algorithm[6]. It is based on the concept of assigning a task having minimum completion time(MCT) first for execution on the resource, which has the minimum completion time(fastest resource).this algorithm is divide into two steps:

In first steps, expected completion time of each task in the metatask is calculated on each resources.

In second steps, the task with minimum expected completion time is selected and assigned to the corresponding resource, and then the selected task is removed from the metatask.This process repeats until all the tasks in metatask get mapped[7].

It starts with a set S of all unmapped tasks. Then the resource R which has the minimum completion time for all tasks is found. Next, the task T with the minimum size is selected and assigned to the corresponding resource R (hence the name Min-Min). Last, the task T is removed from set S and the same procedure is repeated by Min-Min until all tasks are assigned (i.e., set S is empty). assuming we have a set of n tasks (T1, T2, T3 … Tn) need to be scheduled onto m available resources (R1, R2, R3 … Rm). We denotes the Expected Completion Time for task i ($1 \le i \le n$) on resources j ($1 \le j \le m$) as Ctij that is calculated as in (1),

Where rtj represents the Ready Time of resource Rj and Etij represents the Execution Time of task Ti on resource Rj.

Ctij = Etij + rtj

1. For all submitted tasks in the set; T
2. For all resources; Rj
3. Ctij=Etij+rtj; End For; End For;
4. Do while tasks set is not empty
5. Find task Tk that cost minimum execution time.
6. Assign Tk to the resource Rj which gives minimum expected complete time
7. Remove Tk from the tasks set
8. Update ready time rtj for select Rj
9. Update Cij for all Ti
10. End Do

Min-min algorithm

Min-Min algorithm is a simple algorithm but it gives the fast result when the size of task in metatask is small as compared to large size task, on the other hand ,if large size tasks,it gives a poor resource utilization and large makespan because size tasks have to wait for the completion of smaller tasks.

B) Max-Min Algorithm

Max-Min is a resource allocation and scheduling algorithm used in cloud and in grid computing environments to minimize makespan, cost and maximize profit and resource utilization. This is done by selecting a task in the job list with the highest completion time on a resource that can execute it within a shorter period of time.

The working of max-min algorithm appears same as min-min algorithm, instead of assigning minimum executing jobs giving priority to task having maximum execution time. Once the completion of lengthy jobs, task with minimum execution time will be assigned to the processor or else the short

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 7, Issue 4
April 2018

unit jobs have to wait upto the completion of all lengthy tasks. This algorithm also maintains time sequences about the task engaged and update the execution time periodically to the load balancer as well as processor.

The concern of this algorithm is to give priority to tasks with maximum completion time [8], [9] by executing them first before assigning other tasks that has minimum completion time [8]. However, the disadvantage of Max-Min algorithm is that, the execution of task with maximum completion time first might increase the total response time of the system [9] thus making it inefficient.

Previous researchers on scheduling techniques in cloud computing have endorsed Max-Min as the best method of scheduling resources in cloud and grid computing. These researchers have contributed tremendously by making Max-Min an efficient task scheduling algorithm. The efficiency of this algorithm has made cloud computing acceptable in both educational institutions and industries as a preferred platform for data storing and information dissemination.

It starts with a set S of all unmapped tasks. Then the resource R which has the minimum completion time for all tasks is found. Next, the task T with the maximum size is selected and assigned to the corresponding resource R (hence the name Max-Min). Last, the task T is removed from set S and the same procedure is repeated by Max-Min until all tasks are assigned (i.e., set S is empty). assuming we have a set of n tasks (T1, T2, T3 … Tn) need to be scheduled onto m available resources (R1, R2, R3 … Rm). We denotes the Expected Completion Time for task i ($1 \le i \le n$) on resources j ($1 \le j \le m$) as Ctij that is calculated as in (1), Where rtj represents the Ready Time of resource Rj and Etij represents the Execution Time of task Ti on resource Rj.

$$Ctij = Etij + rtj$$

1. For all submitted tasks in the set; T
2. For all resources; Rj
3. Ctij=Etij+rtj; End For; End For;
4. Do while tasks set is not empty
5. Find task Tk that cost maximum execution time.
6. Assign Tk to the resource Rj which gives minimum expected complete time
7. Remove Tk from the tasks set
8. Update ready time rtj for select Rj
9. Update Cij for all Ti
10. End Do

Max-min algorithm

Max-min algorithm perform better than Min-min algorithm in such situation when the number of short task is more than the longer tasks.[11] For example, if there is only one long task, the Max-min algorithm executes many short tasks concurrently with the long one. In order to avoid the main drawbacks of the Max-min and Min-min, the two schedulers can be executed alternatively to each other for assigning tasks to appropriate Resources, for eliminating each other drawback. Such method, called Resource Awareness Scheduling Algorithm (RASA) which was a new task scheduling algorithm[12].

C) Artificial Bee Colony Algorithm

The artificial bee colony algorithm (ABC) is an optimization algorithm based on the intelligent foraging behavior honey bee swarm and was proposed by Kraboga in 2005[13]. The algorithm is completely inspired by nautral foraging behavior of honey bee.

In the ABC model, [14]the colony consists of three groups of bees: employed bees, onlookers and scouts. It is assumed that there is only one artificial employed bee for each food source. In other words, the number of employed bees in the colony is equal to the number of food sources around the hive. Employed bees go to their food source and come back to hive and dance on this area. The employed bee whose food source has been abandoned becomes a scout and starts to search for finding a new food source. Onlookers watch the dances of employed bees and choose food sources depending on dances.

Employed bees be the food offer and provide the neighborhood of the provision in its memory. once sharing the information among the dance area, used bees attend food provide visited by its previous cycle and choose new food offer by victimization the information among the neighborhood. Then spectator prefers a food provide looking forward to nectar information provided by used bees. Onlooker bees get the information regarding food sources from used bees in hive and select one altogether the sources. Spectator bee is anticipating a dance to determine on a food provide. Waggle/tremble/Vibration dances are performed by the bees to relinquish a plan regarding quality and quantity of food and its distance from bee hive.

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 7, Issue 4
April 2018

Scout bee disbursed random search. once the nectar provide is abandoned by the bees, a current food provide is indiscriminately determined by a scout bee.. The main steps of the algorithm are given below

1) Initial food sources are produced for all employed bees.

2) REPEAT

- Each employed bee goes to a food source in her memory and determines a neighbor source, then evaluates its nectar amount and dances in the hive
- Each onlooker watches the dance of employed bees and chooses one of their sources depending on the dances, and then goes to that source. After choosing a neighbor around that, she evaluates its nectar amount.
- Abandoned food sources are determined and are replaced with thenew food sources discovered by scouts.
- The best food source found so far is registered.

3) UNTIL (requirements are met)

In ABC, a population based algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees is equal to the number of solutions in the population. At the first step, a randomly distributed initial population (food source positions) is generated. After initialization, the population is subjected to repeat the cycles of the search processes of the employed, onlooker, and scout bees, respectively. An employed bee produces a modification on the source position in her memory and discovers a new food source position. Provided that the nectar amount of the new one is higher than that of the previous source, the bee memorizes the new source position and forgets the old one. Otherwise she keeps the position of the one in her memory. After all employed bees complete the search process, they share the position information of the sources with the onlookers on the dance area. Each onlooker evaluates the nectar information taken from all employed bees and then chooses a food source depending on the nectar amounts of sources. As in the case of the employed bee, she produces a modification on the source position in her memory and checks its nectar amount. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. The sources abandoned are determined and new sources are randomly produced to be replaced with the abandoned ones by artificial scouts.

III EMPIRICAL COMPARATIVE ANALYSIS

Simulation of three load balancing algorithm:- Min-Min, Max-Min and Artifical bee colony(ABC) algorithms has been done by using Matlab as a simulation tool. In order to illustrate algorithms, assume we have ten tasks submitted by different users for scheduling on five available resources. Table I, represents the processing speed and service level of each resource while Table II, represents the task size and the user group of each task. Data given in Table I and Table II are used to calculate the expected completion time and execution time of the tasks on each of the resources.

TABLE I.
RESOURCES SPECIFICATION

Resources	Processing Speed(MB/Sec)	Service Level
R1	10	VIP
R2	8	Ordinary
R3	9	Ordinary
R4	5	Ordinary
R5	3	Ordinary

TABLE II. TASKS SPECIFICATION

Tasks	Processing Speed(MB/Sec)	Service Level
T1	95	VIP
T2	24	VIP
T3	61	Ordinary
T4	49	Ordinary
T5	89	Ordinary
T6	76	Ordinary
T7	46	Ordinary
T8	3	Ordinary
T9	82	Ordinary
T10	45	Ordinary

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 7, Issue 4
April 2018

Min-Min Algorithm Completion Time:

Table III demonstrates the complete time of each task basis on the execution time of the task on each resource. On next step of the algorithm iteration, data in Table III will be updated until all tasks are allocated.

TABLE III. COMPLETION TIME (EXPECTED COMPLETE TIME) OF TASKS ON EACH OF THE RESOURCES : MIN-MIN SCHEDULING ALGORITHM

Task/RES	(VIP)R1	R2	R3	R4	R5
T1	20.4	~~27.87~~	~~28.55~~	~~34.2~~	~~31.66~~
T2	~~2.7~~	~~3~~	2.66	~~4.8~~	~~8~~
T3	10.9	~~13.375~~	~~14.88~~	~~12.2~~	~~20.33~~
T4	~~9.7~~	~~11.875~~	8.11	~~9.8~~	~~16.33~~
T5	~~19.8~~	~~27.125~~	18	~~33~~	~~29.66~~
T6	~~18.5~~	~~15.25~~	~~16.55~~	15.2	~~25.33~~
T7	~~9.4~~	5.75	~~7.77~~	~~9.2~~	~~15.33~~
T8	0.30	~~0.375~~	~~0.33~~	~~0.6~~	~~1~~
T9	~~19.1~~	16	~~17.22~~	~~31.6~~	~~27.33~~
T10	4.8	~~5.625~~	~~7.66~~	~~9.0~~	~~15~~

Max-Min Algorithm Completion Time:

Table IV demonstrates the complete time of each task basis on the execution time of the task on each resource. On next step of the algorithm iteration, data in Table IV will be updated until all tasks are allocated.

TABLE IV. COMPLETION TIME (EXPECTED COMPLETE TIME) OF TASKS ON EACH OF THE RESOURCES : MAX-MIN SCHEDULING ALGORITHM

Task/RES	(VIP)R1	R2	R3	R4	R5
T1	9.5	~~11.87~~	~~10.5~~	~~19.0~~	~~31.6~~
T2	18	~~18.7~~	~~18~~	~~20~~	~~23.33~~
T3	15.6	~~17.8~~	~~16.6~~	~~27.4~~	~~20.3~~
T4	~~20.5~~	~~16.3~~	15.33	~~25~~	~~16.33~~
T5	~~18.4~~	~~11.12~~	9.88	~~17.8~~	~~29.66~~
T6	~~17.1~~	~~19.75~~	~~18.3~~	15.2	~~25.33~~
T7	~~20.2~~	~~16~~	~~20.44~~	~~24.4~~	15.33
T8	~~18.3~~	~~16.25~~	15.66	~~15.8~~	~~16.33~~
T9	~~17.7~~	10.25	~~19~~	~~16.4~~	~~27.33~~
T10	~~20.1~~	15.87	~~20.33~~	~~24.2~~	~~30.33~~

Artificial Bee Colony Algorithm Completion Time:

In ABC Algorithm, random solutions are generated for the maximum 100 iteration, and the ready time is calculated for each resources. Table V demonstrate the ready time of each resource

RT =

R1(VIP)	R2	R3	R4	R5
17.10	16	15.33	17	16

.The experiment is carried out to compare the performance of the algorithm on the Makespan, Average Resource Utilization Ratio, Average VIP Tasks' Completion Time, and Average Ordinary Tasks' Completion Time.

1) **Makespan:**Makespan is a measure of the throughput of the heterogeneous cloud system. It can be calculated as the following relation:

Makespan=max (rtj)

Where rtj denotes the ready time of each resource after scheduled. The less the makespan of a scheduling algorithm the better it works.

2) **Average Resource Utilization Ratio (ARUR):** ARUR is calculated through the following relation:

ARUR= mean (rtj)/Makespan*100%

Where ARUR is in the range 0 to 1. The best and most efficient load balancing level is achieved if ARUR equals 1. So, scheduling algorithm will have better performance if ARUR is close to 1.

3) **Average VIP Task Completion Time (AVIPCT)** is calculated by

AVIPCT= mean (CTj-VIP)

The less the AVIPCT of a scheduling algorithm produced, the better it works.

4) **Average Ordinary Task Completion Time (AORDCT)** is calculated by

AORDCT= mean (CTj-ORD)

The less the AORDCT of a scheduling algorithm produced, the better it works. In order to meet user priority demand, it will result in the tradeoff between AVIPCT and AORDCT.

5) **MetaTask(MT):**It is a set of tasks , which will be mapped on aviable resources in cloud computing[8].It can be shown as:

$MT=\{t_1,t_2,t_3,\ldots\ldots\ldots,t_n)$

After implementing the Abc, min-min,max-min load balancing algorithm with 10 tasks and 5 virtual machines(vm) their makespan and resource

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 7, Issue 4
April 2018

utilization is calculated. The result Ofmakespan is shown with the help of bar graph in Fig2.

Approach	Makespan	AvgRU	VM Counts
Min-Min	20.4	50.06	[4 2 3 1 0]
ABC	17.8	92.38	[2 3 2 2 1]
Max-Min	18	78.13	[3 2 3 1 1]

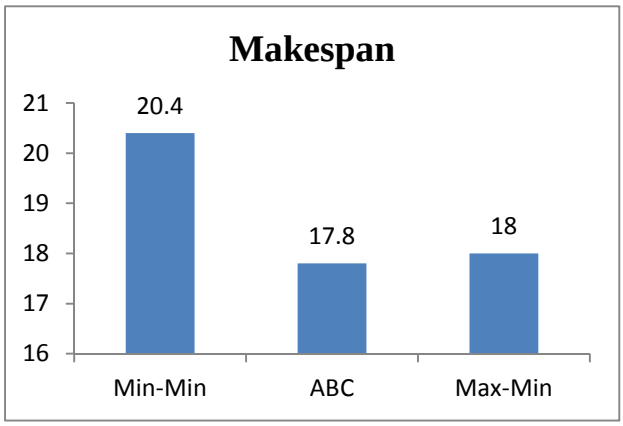

Fig 2: Performance Analysis Using Makespan.

The average resource utilization is better in artificial bee colony as compare to the min-min and max-min load balancing algorithm, the result is shown in fig 3.

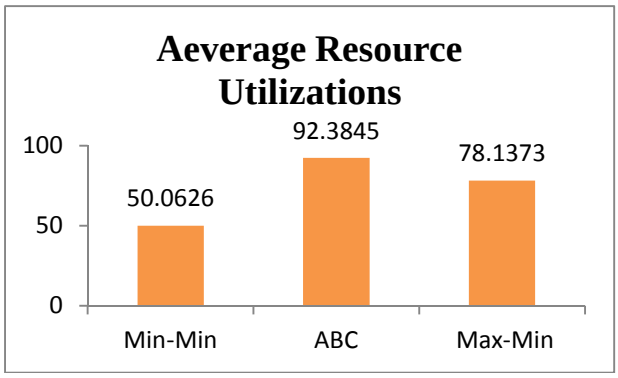

Fig 3: Performance Analysis Using Average Resource Utilizations.

In this paper we calculated the resource utilization time or ready time of each resource.

RT	R1	R2	R3	R4	R5
Min-Min	20.4	16	18	15.2	0
ABC	16.7	17.5	13.88	17.8	16.33
Max-Min	18	15.87	15.66	15.2	15.33

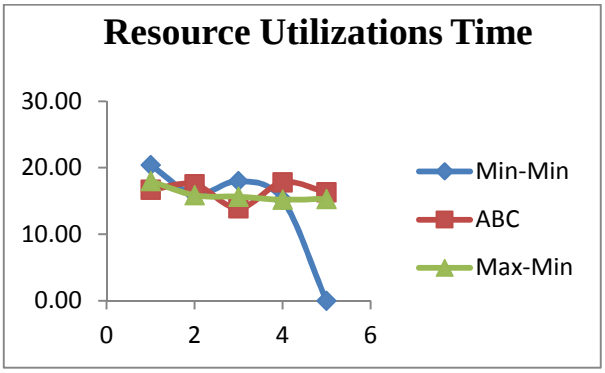

Fig 4: Resource Utilizations Time.

After implementing these three algorithms of load balancing we concluded that the Artifical bee colony algorithms is better than the min-min and max-min load balancing algorithms. It was always said that dynamic load balancing algorithm is better than the static load balancing algorithm. But, in static load balancing algorithm Max-Min algorithm is performed better than the Min-Min load balancing algorithm,because in this algorithm the small size tasks are concurrently executed with large size task,thus reducing the makespan and providing better resource utilizations.

IV. CONCLUSION AND FUTURE SCOPE

Load balancing is the main challenge in cloud computing. Finding an efficient load balancing algorithm has always been the prominent field for research. In this Paper, simulation based performance analysis of threewidely used algorithms- Min-Min, Max-Min and Artificial Bee Colony - have been done on the basis of parameters like makespan, average resource utilizations and resource utilization time. It has been found from the experimental analysis that Artificial Bee Colony algorithm perform better in term of all the three parameters than the other two algorithms. It has further been observed that in Min-Min and Max-Min algorithms the resources are not optimally utilized while in Artificial Bee Colony algorithm resources are better utilized. Experimental results have shown that in Min-Min and Max-Min algorithms the makespan are poor as compared to artificial bee colony algorithm.The resource utilization time is also better in case of artificial bee colony.Experimental results shown that the Max-Min outperform the Min-Min when number of large sized tasks is more than the short length task. But

when short length tasks are more than the long length tasks, Min-Min outperform the Max-Min algorithm. In future there is a wide scope of improvement in artificial bee colony load balancing algorithm where the ArtificialBee Colony and load balancing module willbe made hybrid for making the result betterin term of makespan and average resource utilizations and resource utilization time.

REFERENCE

[1]. Tejinder Sharma, Vijay KumarBanga, "Efficient and Enhanced Algorithm in Cloud Computing",IJSCE,ISSN:2231-2301,2013

[2]. Joshua Samuel Raj, Hridya K.S,V. Vasudevan, "Augmenting Hierarchical Load Balancing with Intelligence in Grid Environment", International Journal of Grid and Distributed Computing Vol. 5,No. 2, pp. 9- 8, 2012

[3]. Guopu.Zha, Sam Kwong , "Gbest-guided Artificial bee Colony algorithm for numerical function optimization", Elsevier 31663173,2010

[4] G. Siva Shanmugam"Effort Of Load Balancer to achieve Green Cloud Computing ", International Journal of Multimedia and ubiquitous Engineering,VOL.11 No.3,2016.

[5] Y.Zhang, H.Kameda& S.L. Hung "Comparison of dynamic and static load-balancing strategies in heterogeneous distributed system" IEEE proceedings in Computer and Digital Techniques,144(2), 1997.

[6] Yu, Xiaogao, and Xiaopeng Yu. "A new grid computation-based Min-Min algorithm", Fuzzy Systems and Knowledge Discovery, FSKD'09 Sixth International Conference on, Volume 1, Pages 43 – 45, IEEE, 2009.

[7] Nehasharma,Dr.Sanjaytyagi ,"Compartive analysis of min-min and max-min on the basis of makespan parameters", International Journal Of Advanced Research in Computer Science,8(3),March-April2017,1038-1041[8] Elzeki, O., Rashad, M., &Elsoud, M."Overview of scheduling tasks in distributed computing systems", International Journal of Soft Computing and Engineering (IJSCE),2(3),470-475,2012.

[9] Santhosh, B., &Manjaiah, D. (2014),"An improved task scheduling algorithm based on max-min for cloud computing",International conference on Advances in computer & communications Engineering(ACCE-2014),vol 2,issue2,may2014.[10]Brar, S. S., &Rao, S. "Optimizing workflow scheduling using max-min algorithm in cloud environment", International Journal of Computer Applications, 124(4), 2015.[11]Elzeki, O. M., M. Z. Reshad, and M. A. Elsoud. "Improved Max-Min Algorithm in Cloud Computing", International Journal of Computer Applications, Volume 50, Issue12,Pages22-27,2012.[12]SaeedParsa, Reza Entezari-Maleki,"RASA: A New Grid Task Scheduling Algorithm", International Journal of Digital Content Technology and its Applications Volume 3, No 4,dec ,2013..[13]. D.Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report TR06,Computer Engineering Department,Erciyes University, Turkey,2005.[14].Karaboga, D.; Basturk, B. "On the performance of artificial bee colony (ABC) algorithm". Appl. Soft Comput. J. 2008, 8, 687±697.