Random Walk based ACO Load Balancing Algorithm for Cloud Computing Environment

Nishu Rana[1], Dr. Pardeep Kumar[2]
[1]M.Tech Scholar, Associate Professor[2]
[1,2]Department of Computer science and Application, Kurukshetra University, Kurukshetra, India

Abstract

In a few years, Cloud computing has got unstoppable growth. As the Cloud Computing is developed day by day the Cloud providers requires optimization of various services to achieve a high level of security, availability and responsiveness. The virtual machines are migrated lively to produce efficient result to realize load balancing as well as to optimize utilization of resources. Now a day, the most challenging for service providers is to maintain reliability and elasticity and lesser the Makespan (MS) and better the resource utilization (RU). That is the reason Cloud Service providers requires a dynamic load balancing algorithm. Dynamic Load Balancing (DLB) algorithms are those that deceases the Makespan (MS) while increases the resource utilization. For such problems, Metaheuristic Optimization Approaches have been successfully proved to produce near-optimal solutions with fair time. In order to improve the cloud computing utilization, Random Walk Ant Colony Optimization (RWACO) is proposed. The proposed RWACO algorithm improves pheromone factors, Makespan as well as for better Resource Utilization (RU) characterized by the existing algorithm. Results of simulation are conducted in the CloudSim and these results indicate that RWACO is superior to the conventional ACO.

Keywords

Cloud Computing, Load Balancer, Ant Colony Optimization, Levy Flight's, Make-span, Resource-utilizations.

I.INTRODUCTION

In cloud computing environment one of the most challenging is the Load Balancing. It is a useful method in which workload is distributed from one to other side servers, networks or other computing resources as shown in fig 1.

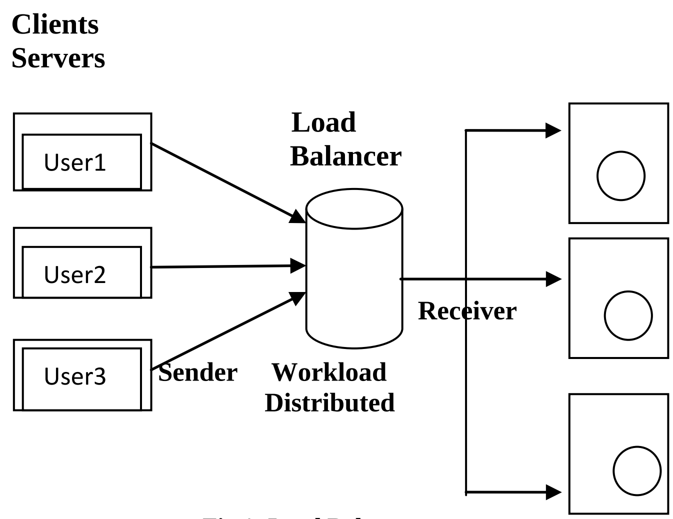

Fig 1: Load Balancer

Load balancing making sure that none of the existing resources is idle while others are being utilized. Load distribution is balanced by moving the load from the heavily source nodes to the lightly destination node [1]. With such a distribution workload can be avoided to some extent. Load balancing is procedure to optimize the resources uses and reduces response time [2]. While balancing the load in cloud computing several issues are exits like one is security, other is the speed of services and reliability etc. That's why, for developing a successful load balancing algorithm, one must try to take care of various factors such as to judging the load correctly performances monitoring as well as system stability and last is the node selection etc.
Load balancing algorithm is of two types**:-**

- **Static Algorithms:** Those algorithms in which network traffic is divided among the servers or nodes. Static algorithm do not considered the work performance of the last task and depend on current state system.

- **Dynamic Algorithm:** Those algorithms in which the server which is either idle or having least load in the whole network or system is searched and preferred for assigning load. Here current state of the system is used to make decisions to manage the load.

The rest of the paper is presents as: Section (II) presents Related Work on Ant Colony Optimization algorithm. Section (III) presents Problem Formulation. In Section (IV), Random Walk Ant Colony Optimization (RWACO) algorithm is described. Section (V) describes the Simulation Based Analysis. Finally the Conclusion and Future Scope is given in Section (VI).

II.RELATED WORK

As Load Balancing is important and complicated inside the Cloud Computing, various researchers make efforts to achieve various techniques for load balancing optimization and workload distribution. Wei-Tao et. al, in [3] presented VM migration strategy based on metaheuristic which is called Ant Colony Optimization. In ACO-VMM, local migration agents autonomously monitor the resource utilization and launching the migration. Results show that ACO-VMM outperforms the existing migration strategies by achieving load balance of whole system, as well as reducing the number of migrations and maintaining the required performance levels.

Krishna H. Hingrajiya et. al, in [4] presented an approach for solving traveling salesman problem based on improved ant colony algorithm. The main contribution of this paper is a study of the avoidance of stagnation behavior and premature convergence by using distribution strategy of initial ants and dynamic heuristic parameter updating based on entropy. Then, the local search solution is provided. The experimental results as well as the performance comparison showed that the proposed system reaches the better search performance over ACO algorithms. The proposed system is better in terms of convergence speed and the ability to finding better solution. Yang Xianfeng et. al. in [5] proposed to employ Genetic Algorithm (GA) for ACO initialization. ACO could arrive at local optimal point, and the convergence speed is typically low. Along this line, they introduce the idea of Simulated Annealing (SA) to avoid local optimal and accelerate the convergence. Lastly, their experiments show that improved ACO achieves good performance in load balancing.

Ashish Gupta and Ritu Garg in [6] proposed a multi-objective scheduling algorithm considering the optimization of makespan and load balancing has been proposed. The algorithm uses the ACO approach to obtain the local optimal solutions, finally non domination sorting is applied to obtain the Pareto set of solutions representing the trade-off between makespan time and the load balancing in cloud. Thereby, meeting the demands of users in terms of execution performance and increases the utilization of resources. Using CloudSim toolkit, the proposed scheduling mechanism has been simulated and the results specify that the proposed LB-ACO algorithm for task scheduling outperforms in comparison to NSGA-II.

Awatif Ragmani et. al, [7] proposed an improved architecture for load balancing by offering to share the functions of the main controller representing the entry point to Cloud into two parts. Firstly, the main controller will keep the function of partitioning user's tasks through different regional load balancers. Secondly, the auxiliary controller will insure the updating state of the system through various agents. Through this approach, their aim to optimize the response times of services in the Cloud. Following this work, the plan to test the solutions proposed firstly, on the CloudSim simulator and secondly in a real environment and finally to improve the proposed algorithm.

Navtej Singh Ghumman and Rajwinder Kaur in [8] proposed a hybrid improved max min and ant approach which is performed by cloudSim. This helps in better load balancing. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load between all nodes of a cloud system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. This study is concentrated only on tasks and resources. They considered the load of planet lab, but still take limited number of cloudlets because it effect on system speed.

Yongjun Sun, in [9] proposed with aiming some features of the wireless sensor network the improved ant colony routing algorithm is proposed. By improving the heuristic function and giving overall consideration to transmission distance of node communication, transmission direction, residual energy and distance from the Sink node, a new route updating rule was introduced, which leads to a relatively high average residual energy level and a high minimal value of the residual energy. The simulation results indicate that by comparing with EEABR, Leach-Ant and OARA, the method proposed in the paper obviously minimizes the average energy consumption and extends the life cycle of the wireless sensor network.

Peng Yinghui in [10] proposed vocational skills certification examination in different professions, different types and different levels has developed vocational skills certification exam management system. Using Ant colony optimization (ACO) algorithm intelligently make skills identification papers has solved the multi-objective problem; so determine vocational skills certification examination test of the different professions, different types, different levels of difficulty with normal distribution, meet different groups requirements of vocational skills certification examination. Through specific simulation experiments and found that the improved ant colony algorithm can automatically generate moderate difficulty paper covering range of knowledge, student test scores was good using this papers.

N. A. Rahmat in [11] proposed Differential Evolution Ant Colony Optimization (DEACO) to optimize Economic Load Dispatch in power system. Implementation of the IEEE Reliability Test System (RTS) demonstrated that this technique is feasible to crack the economic problem. Comparative studies with respect to ACO and the traditional ELD techniques designate that the proposed DEACO outperformed these two techniques. The results are very promising and proved that the DEACO is efficient and faster than ACO. Further studies are required to enhance the capability of the DEACO engine.

III. PROBLEM FORMULATION

For an appropriate load balancing algorithm, includes all Jobs (called Tasks) to Cloud based Resource (called Virtual Machines) Main objective function is to reduce Makespan time ratio of scheduling algorithm and increasing the Resource Utilization (RU) taking Levy's Flight (LF). Here problem is depending on multi-criteria optimization function which minimizes the makespan and maximizes the resource utilization. Levy Flight's is a very good solution to such type of problem. Levy Flight's said to be as Random Walk (RW).The Random Walk are those in which the step-lengths have a probability distribution that is heavy tailed. By using Levy flights (LF) random search performed can be done more effectively as compared to an easy random walk [12]. ACO is an example of easy random walk. The basic equation of Levy flight (LF) is following as:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \alpha \oplus levy(\lambda)$$

Where the "t" defines: Current Generation, and the "Θ" defines the multiplication of one or more, and alpha > 0 defines the size of steps. Levy flights (LF) in Equation (1) are changed by the Levy Flights distribution that is following as:

$$levy(\lambda) \sim g^{-\lambda}, \quad (1 < \lambda \leq 3)$$

(2)

The Levy flights having mainly two steps: firstly

(i) Random direction chosen accordingly uniform distribution (ii) After that, the chosen Levy distribution is generated by sequences steps.

IV. RANDOM WALK ANT COLONY OPTIMIZATION (RWACO)

A. Levy Flight's

Animals foraging path was considering as a random or sub-random manner in nature, Most of the studies have shown that the behavior of the levy flights (LF), typically the characteristics of levy flights. In simple words, it is random walk whose step-length is shown from the levy flights distribution. It can be define by the basic formula of power-law:

Levy flights (s) ~|s| and (-1-beta) where 0<beta<2.

In this (RWACO) carries the steps of Ant Colony Optimization (ACO) which is defined above and can carry result of past Load Balancing following new task scheduling.

This is helpful in the Cloud Computing Environment. As compare to the Ant Colony Optimization (ACO) algorithm, the RWACO algorithm carries the basic ACO algorithm to lesser the Makespan (MS) time ratio of scheduling tasks, and better Resource Utilizations (RU) as well as taking the loading of various Virtual Machines (VMs). For improving the performance of ACO, a possible method is to change the pheromone volatilization coefficient ρ using Levy distribution while ρ is a constant in ACO, so proposed a new Random Walk ACO algorithm based on Levy Flight (RWACO) in which each movement of each ant individual obey the levy distribution. In RWACO, the pheromone equation (3) is introduced and gets the following equation for updating the location of each ant individual:

$$\tau ij \ (t+1) = (1 - \rho i,\ j) * \tau ij \ (t) + \Delta \tau ij \ (t)$$

(3)

In equation (3)

$$\rho_{i,j} \sim levy(\beta)\ ,\ levy(\beta) \sim \frac{u}{|v|^{\frac{1}{\beta}}}(\tau_{ij}(t+1) - \tau_{ij}(t)) \text{ and } u \sim N(0,\sigma_u^2)\ ,\ v \sim N(0,1)\ ,\ \sigma_u =$$

$$\frac{\mathcal{T}(1+\beta)\sin(\beta\pi/2)}{\mathcal{T}\left(\frac{1+\beta}{2}\right)\beta 2^{(\beta-1)/2}}^{1/\beta}$$

(4)

The Flowchart as well as pseudo code of the Random Walk Ant Colony Optimization (RWACO) is discussed step wise. With changing in the pheromone factor equation and by adding Levy Flight's equation decreases the makespan and increases the resource utilization which is the aim of this paper. Now if we combined the above analysis, the step of RWACO is following as figure 4.

B. Flow chart of RWACO

The flow chart of Random Walk based Ant Colony Optimization (RWACO) are described below.

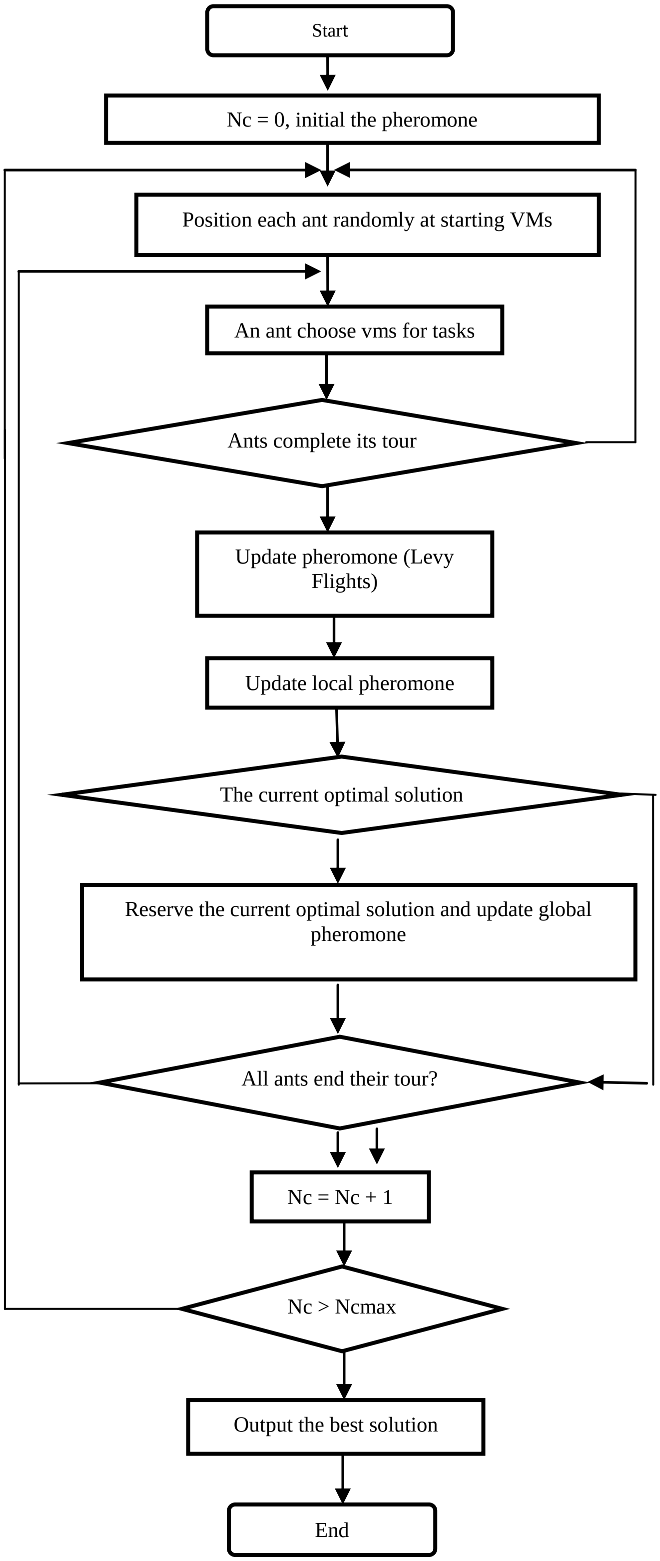

Fig 2: FLOWCHART OF MACO

(II) Programming steps of MACO

The programming steps of the proposed RWACO algorithm in searching for the minimum makespan path can be described as follow:

Procedure RWACO

Fig 3: Programming Steps Of Rwaco

Begin
Initialize the pheromone
Position each Ants randomly at starting VMs
Choose all Ants for next task VM
Ants complete its tour, update pheromone with levy flights (acc. to eq. 3 and 4)
Update local pheromone
Ants end their trip
Nc = Nc + 1, reserve current optimal solution
Nc > Nmax, if criteria satisfy
End index no.
Update best solution
End

A basic step of Random Walk is given here in which firstly initialize the pheromone value of all virtual machines (VMs). After that have to place all ants at the starting virtual machine (VM) randomly. Then, have to choose virtual machines for next tasks. Update the pheromone with levy flights (equation 3 and 4) as all ants complete their tour. Then, update local pheromone. Then the Makespan (MS) of all ants is calculated and Update presently optimal solution. Now if a criterion satisfied then end the iterations. Lastly, update the global pheromone with the best solution.

IV. EXPERIMENTAL RESULTS

Simulation based analysis on two load balancing algorithms that is Ant Colony Optimization (ACO) and Random Walk Ant Colony Optimization (RWACO) have been done by using CloudSim toolkit. It is a tool which allows researchers for check out their performance problem in Cloud Computing Environment and it is cost free. CloudSim basically is a simulation based toolkit; it does not belonging to a running software.

From above experiments Table 1, Table 2 and Table 3 shows that Random Walk Ant Colony Optimization (RWACO) performs better as compared to the Exiting algorithm i.e. Ant Colony Optimization (ACO) with their graph representation. The experiments have done by comparing the performance of the algorithm on the basis of makespan, Average Resource Utilization (ARU) and LBL. There are 5 numbers of tasks which varies and 5 virtual machines that are fixed, similarly on the other hand there are 5 numbers of tasks which are fixed and 5 virtual machines that vary on which makespan, avgRU and LBL are performed in Table 1, Table 2. In Table 3 represents averages of both the algorithms with their corresponding bar graphs.

- **MAKESPAN (MS): -** MS is the finishing time in the last task. calculated by the formula:
  Ms = maximum (ready time (Rti))

Where rti: ready time of scheduled resources.
Less Makespan (MS) scheduling algorithm = Better work performs.

- **AVERAGE RESOURCE UTILIZATION (ARU): -** calculated with the formula relation:
  ARU= Mean (ready time (Rti))/MS * 100

Higher ARU = Better work performance.

Now, when tasks are random and Vms are fixed (means taking VMs=5)
So the results of makespan and resource utilization, lbl are shown as follow of both the algorithms i.e. Ant Colony Optimization (ACO) and Random Walk Ant Colony Optimization (RWACO).

Table 1: Makespan

No. of tasks	ACO-Makespan	RWACO-Makespan
100	46.79	43.01
300	142.35	137.3
500	227.51	215.07
700	312.7	311.22
900	433.05	420.02
1000	458.05	445.54

After implementing the Random Walk Ant Colony Optimization (RWACO) and Ant Colony Optimization (ACO) by performing 6 iterations their makespan, resource utilization as well as LBL are calculated. The result of makespan is shown with the help of bar graph in fig 4.

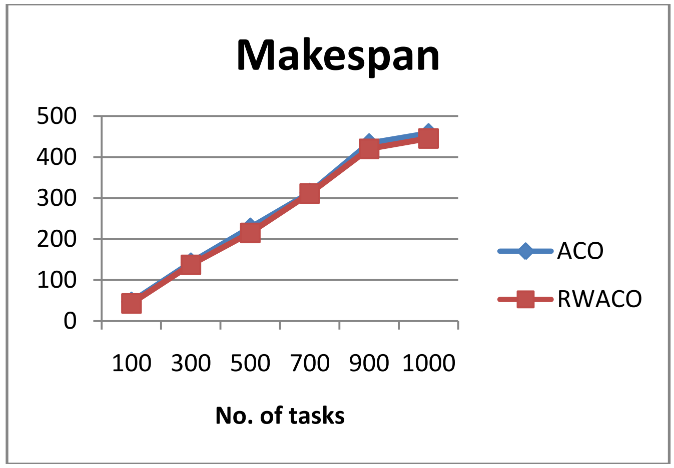

Fig 4: Performance Analysis Using Makespan

Table 2: Resource Utilizations

No. of Tasks	ACO-Resource Utilizations	RWACO-Resource Utilization
100	93.25	93.36
300	98.27	98.65
500	98.09	99.1
700	95.3	97.68
900	95.36	97.58
1000	99.71	98.08

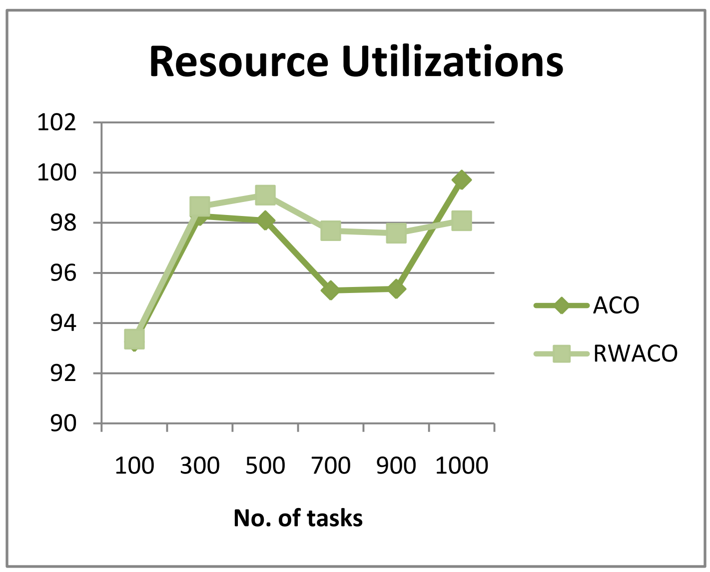

Fig 5: Performance Analysis Using Resource Utilization

Table 3: LBL

No. of Tasks	ACO-LBL	RWACO-LBL
100	95.01	95.86
300	98.83	99.08
500	98.85	99.18
700	99.41	98.32
900	95.59	97.62
1000	99.1	98.45

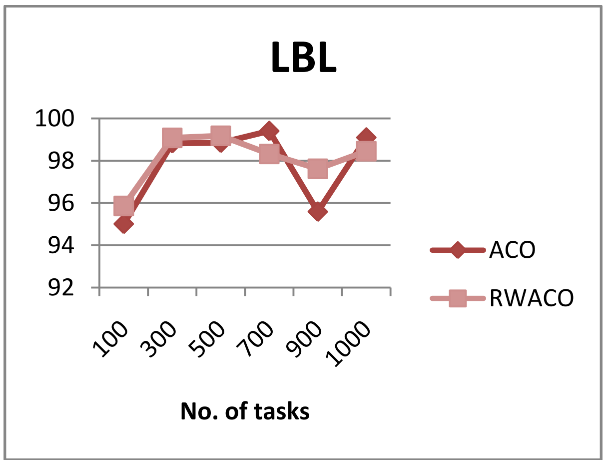

Fig 6: Performance Analysis Using LBL

Now, considered 5 tasks which are fixed whereas VMs are vary. After implementing both the algorithms on basis of multiple objectives their Makespan, Resource Utilization and LBL are as follow along with their bar graph representation.

Table 4: Makespan

No. of VMs	ACO-Makespan	RWACO-Makespan
3	52.2	45.82
5	49.87	40.32
7	30.09	24.32
9	32.21	25.93
11	28.21	21.87

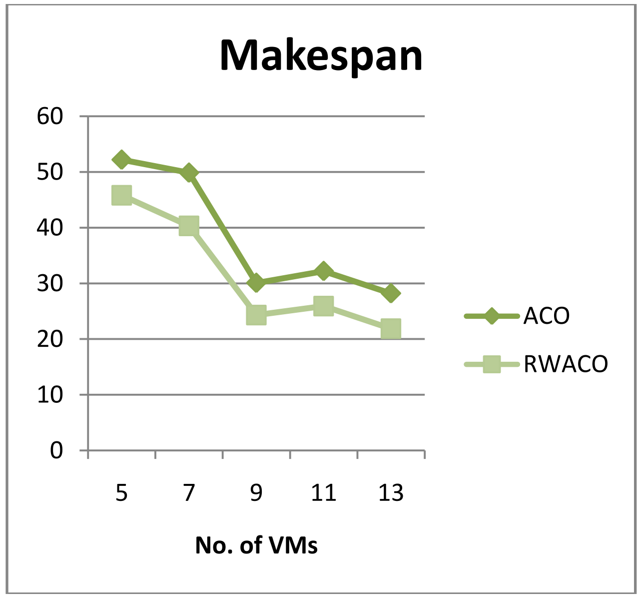

Fig 7: Performance Analysis Using Makespan

Table 5: Resouce Utilization

No. of VMs	ACO-Resource Utilization	RWACO-Resource Utilization
3	90.63	95.08
5	75.76	85.8
7	92.06	92.28
9	91.32	91.91
11	85.16	90.78

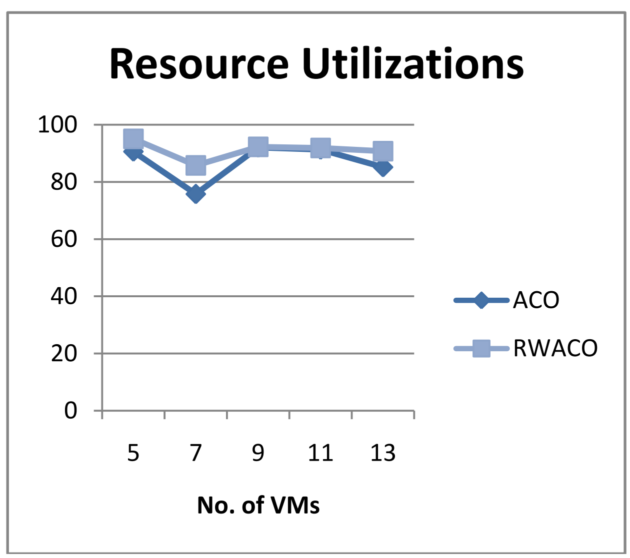

Fig 8: Performance Analysis Using Resource Utilization

Table 4: LBL

No. of VMs	ACO-LBL	RWACO-LBL
3	94.61	96.23
5	94.76	95.45
7	93.06	93.64
9	93.32	94.18
11	93.16	94.3

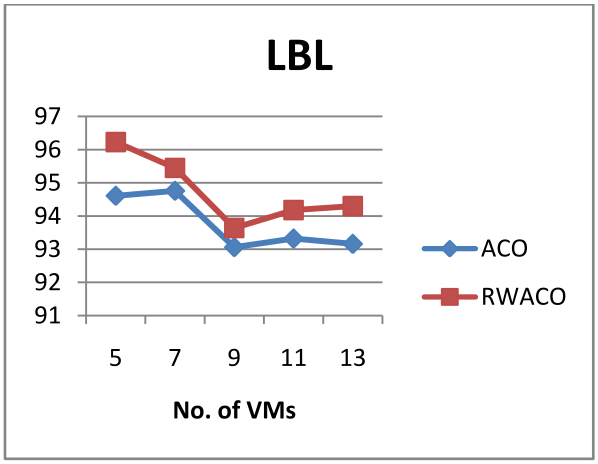

Fig 9: Performance Analysis Using LBL

Now, here represents the Average tables of both the algorithms on the basis of three parameters i.e. Makespan, Resource Utilization and load balancer along with their representative bar graphs.
Average on the basis of when tasks vary whereas VMs are fixed, then Makespan, Resource Utilization and LBL are as follow:

Table 5: Averages

Approaches	Makespan	Resource Utilization	LBL
ACO	270.135	96.66333	97.79833
RWACO	262.0267	97.40833	98.085

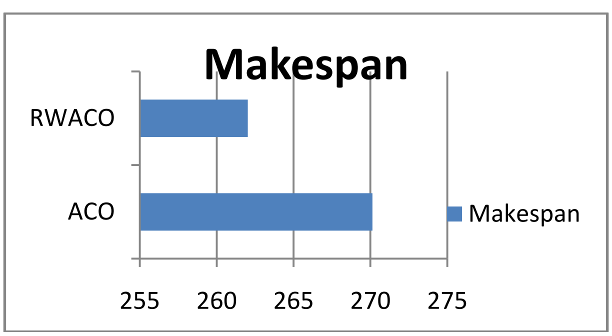

Fig 10: Average Makespan

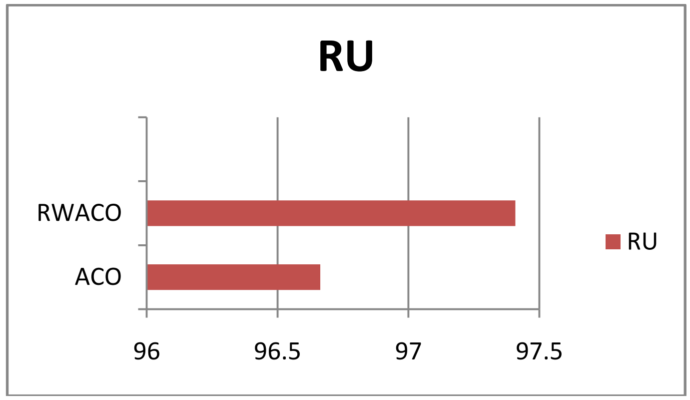

Fig 11: Average Resource Utilization

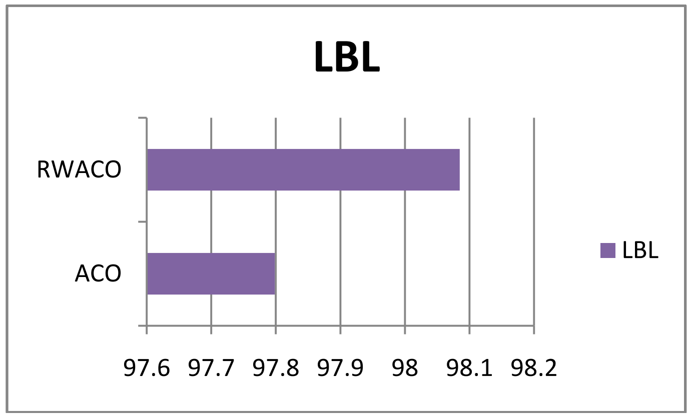

Fig 12: Average LBL

Average on the basis of when tasks are fixed, whereas VMs vary, then Makespan, Resource Utilization and LBL are as follow

Table 6: Averages

Approaches	Makespan	Resource Utilization	LBL
ACO	270.135	96.66333	97.79833
RWACO	262.0267	97.40833	98.085

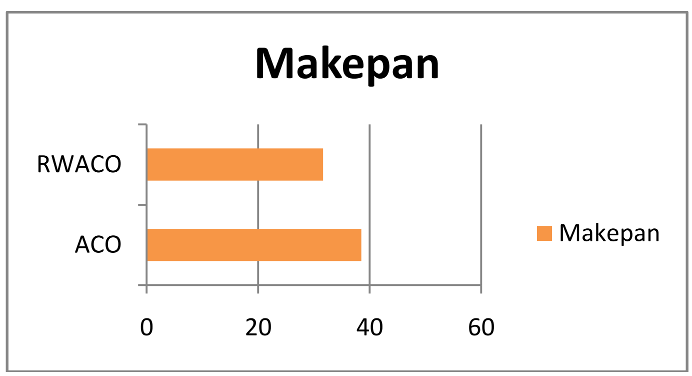

Fig 13: Average Makespan

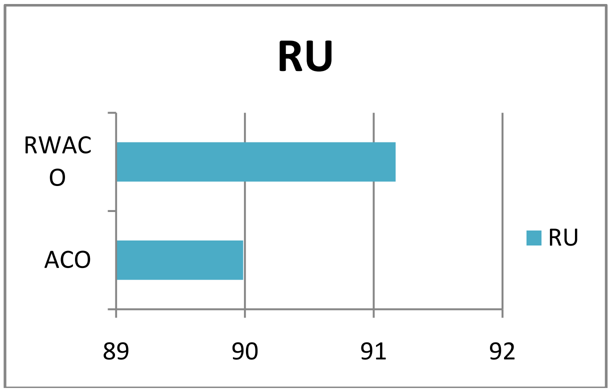

Fig 14: Average RU

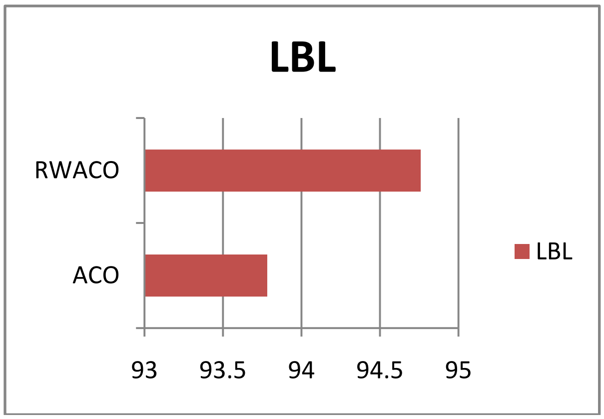

Fig 15: Average LBL

V. CONCLUSION

Load balancing is the main challenge in cloud computing. Finding an efficient load balancing algorithm has always been the prominent field for research. In order to improve performance of ACO algorithm the concept of Levy flight has been added to ACO algorithm for getting the new update of the individual equations. Although ACO's results are good for the task allocations, but RWACO outperforms ACO in all of the cases i.e. irrespective of number of tasks. The execution time for Random Walk Ant Colony Optimization (RWACO) load balancing algorithm is lesser as compared to ACO which will improve the makespan and resource utilization.

In future further implementation of RWACO on various environments can be done for checking its practical feasibility.

REFERENCES

[1] Nitika, Shaveta and Gaurav Raj; "Comparative Analysis of Load Balancing Algorithms in CloudComputing", International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 3, May2012.

[2] V.Krishna Reddy, Srikanth Reddy, "A Survey of Various Task Scheduling Algorithms in Cloud Computing", i-manager's Journal on Computer Science (JCOM), Volume I, Issue 1, 2013.

[3] Wei-Tao Wen et. al," An ACO-Based Scheduling Strategy on Load Balancing in Cloud Computing Environment"364-368 © 2015 IEEE.

[4] Krishna H. Hingrajiya, Ravindra Kumar Gupta, Gajendra Singh Chandel," An Ant Colony Optimization Algorithm for Solving Travelling Salesman Problem" International Journal of Scientific and Research Publications, Volume 2, Issue 8, August 2012.

[5] Yang Xianfeng1 and Li HongTao2" Load Balancing of Virtual Machines in Cloud Computing Environment Using Improved Ant Colony Algorithm" International Journal of Grid Distribution Computing Vol. 8, No.6, (2015), pp.19-30 http://dx.doi.org/10.14257/ijgdc.2015.8.6.03.

[6] Ashish Gupta,Ritu Garg,"Load Balancing Based Task Scheduling with ACO in Cloud Computing " International conference on computer and applications (ICCA),174-179,2017 IEEE.

[7] Awatif Ragmani, Amina EI Omri, Noreddine Abghour, Khalid Moussaid, Mohammed Rid," A Performed Load Balancing Algorithm for Public Cloud Computing Using Ant Colony Optimization" 978-1-4673-8894-8/16/©2016 IEEE.

[8] Navtej Singh Ghumman and Rajwinder Kaur," Dynamic Combination of Improved Max-Min and Ant Colony Algorithm for Load Balancing in Cloud System" 6th ICCCNT - 2015 July 13 - 15, 2015 IEEE.

[9] Yongjun Sun, Wenxin Dong and Yahuan Chen," An Improved Routing Algorithm Based on Ant Colony Optimization in Wireless Sensor Networks" 2016 IEEE.

[10] Peng Yinghui," Ant Colony Optimization Algorithm In Occupational Skill Testing Management System" 978-1-4673-9613-4/16 ©2016 IEEE.

[11] N.A.Rahmat," Differential Evolution Ant Colony Optimization (DEACO) Technique In Solving Economic Load Dispatch Problem" International Power Engineering and Optimization Conferemce (IPEOC) 2012 IEEE.

[12] Andres Iglesias et al, "Cuckoo Search Algorithm with Levy Flights for Global-Support Parametric Surface Approximation in Reserve Engineering 10, 58; doi: 10.3390, 2018"

[13] M.Vazuez and L.D Whitley, "A hybrid genetic algorithm for the quadratic assignment problem,"in GECCO, 2000, pp.135-142