ISSN 2348 – 8034
Impact Factor- 4.022

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

ANT COLONY OPTIMIZATION BASED WORKFLOW SCHEDULING IN CLOUD COMPUTING

Deepika Saxena*[1] and Dr. R.K. Chauhan[2]
*[1]Assisstant Professor , D.N. College, Department Of Computer Science , Kurukshetra University Kurukshetra
[2]Professor, Department Of Computer Science And Applications, Kurukshetra University Kurukshetra

ABSTRACT

In the present scenario of Information and Technology, Cloud Computing has become buzzword. Here, dynamically scalable services and distributed virtualized resources are provided over the internet on pay-as-per use basis. Instantaneously, there are huge numbers of users accessing services of cloud and various tasks need to be handled in the cloud computing environment, the high effective task scheduling algorithm is one of the crucial problems that the cloud computing is required to solve. Cloud task scheduling is an NP-hard optimization problem and many different meta-heuristic algorithms have been proposed to solve it. A good task scheduler should adapt its scheduling strategy dynamically according to the changing environment and the types of tasks. Aiming to the model structure of cloud computing, in this article we have introduced modified Ant Colony Optimization algorithm (ACO) to combine with optimized task scheduling algorithm which is dynamic and adapt according to the availability of resources. This paper relates advanced heuristic and combinatorial optimization problem solving technique i.e. Ant Colony Optimization (ACO) which outperforms over other evolutionary algorithm and optimization technique. In proposed algorithm, group of tasks are represented as workflow are scheduled by ants based on heuristic function to the virtual machine. This means all the available tasks are efficiently scheduled to the very best of its optimization. We recompile the cloudsim and simulate the proposed algorithm and results of this algorithm are compared with sequential task scheduling. The experimental results indicates that proposed algorithm has high performance in terms of least execution time that considers heterogeneous resources and elasticity of clouds that can be dynamically acquired on pay-per-use basis. This algorithm is not only beneficial to user and service provider, but also provides better efficiency by applying load-balancing feature i.e. benefit at system level.

Keywords: *Ant colony optimization, Cloud computing, pheromone, task scheduling.*

I. INTRODUCTION

Computational world is very huge and complex. Cloud computing has undertaken almost entire space of IT development digital computation. Specifically, cloud is a collection of resources (hardware and software) distributed at worldwide datacenters. The term ``Cloud`` means a cluster of interconnected distributed datacenters which are expanded worldwide for example, Google, Amazon etc. At each datacenter, there are several computer systems acting as servers and providing enormous variety of services to the customers. Briefly describing, cloud computing is a model for enabling convenient, on-demand network access to shared pool of computing resources for example, networks, servers, storage, applications and several other services that can be rapidly provisioned and released with minimal management effort or cloud service provider interaction. The basic idea of cloud computing is based on a very fundamental principal of reusability of IT capabilities. The ultimate goal of this Cloud System is to instantly provide resources whenever customer requires it on their laptops, PCs, mobile phones etc wherever possible. There are many servers available at various datacenters forming clusters of cloud which are provided by service providers throughout the world. We use to pay as per our demand for using these resources. There are many popular issues for research in cloud computing like virtualization, data security, license management, scalable storage management, mobile cloud, availability of services, task scheduling. But, scheduling of job is always a prime topic of research in cloud computing. This becomes a very challenging issue to schedule several requests simultaneously at datacenter satisfying all the customers, workload balancing, reduction of energy consumption, faster execution of requests at minimum cost etc. There is always a lot more to explore in task scheduling problem which is NP problem. There are heterogeneous resources available at various datacenters. Our work considers basic services of cloud providers such as dynamic resource allocation of unlimited heterogeneous resources, its elasticity and flexibility along with VM performance variation. So, traditional scheduling algorithms like FCFS, shortest job first, round-robin and priority etc, are not recommendable. Also, Customer put Quality of Service constraints like minimum cost and task completion in minimum time etc. Along with it, CSP (Cloud Service Provider) requires maximum returns on investment. At system level, maximum resource utilization and load balancing is required. In cloud computing, Broker acts as an intermediator between user and CSP. Broker exists at system level. Broker decides where to map job or task submitted by user to the resource provided by CSP. So, while designing any new scheduling algorithm all the changes are performed at DCB (Data Center Broker). Likewise, many researchers have provided many scheduling algorithms, which are working well in one or the other way. With the development of cloud computing, there have been many heuristic algorithms to solve the problem of task scheduling in cloud computing, such as genetic algorithm, particle swarm optimization algorithm and immune evolutionary algorithm. These algorithms usually have poor convergence performance and get fall into premature state soon. Some researchers have used the ant colony optimization algorithm to solve the task scheduling problem, but the study found that the ant colony optimization algorithm is easily trapped into local optimal state and the algorithm has high randomness [25]. Besides most current algorithms ignore the calculated performance of each different virtual machine in cloud data center and the computation amount of tasks submitted by users also have the difference. In that case, it is quiet easy to cause the load imbalance of virtual machine leased by users, accordingly the overall performance of the cloud data center will be under the influence. Aiming at the shortage of the existing scheduling algorithms, this paper proposes a modified ACO cloudlet scheduling algorithm which includes all the essential features like reduced execution time, cost optimization load balancing with improved throughput. In addition, considering that most researches are focusing on scheduling independent tasks and ignore the workflow model with priority constraints users may submit, we are using DAG (Directed Acyclic Graph) to research workflow scheduling. Through considering the temporal and causal constraint for each task to choose the best resources and coordinate the execution of various tasks to obtain the final results. This paper uses CloudSim toolkit as the simulation platform and compares the proposed algorithm with the FIFO scheduling policy, the Min-Max algorithm, Min-Min task scheduling and PSO(Particle Swarm Optimization) based task scheduling in cloud system. The results obtained are showing that proposed algorithm outperforms the existing algorithms in all respect. The rest of the paper is divided as follows: section 2 covers the literature review, section 3 discusses proposed scheduling scheme, section 4 contains proposed algorithm, section 5 shows experimental data and results and finally section 6 concludes this paper.

II. LITREATURE REVIEW

Thousands of user share cloud resources by submitting their computing task to the cloud system simultaneously. Scheduling these huge number of task is a challenge to cloud computing environment. Optimal resource allocation or task scheduling in the cloud should decide optimal number of systems required in the cloud so that the total cost and total execution time is minimized. Cloud service scheduling is categorized at user level and system level [6]. At user level scheduling deals with problems raised by service provision between providers and customers [17, 21]. The system level scheduling handles resource management within data centers [8, 11, 16]. A novel approach of heuristic-based request scheduling at each server, in each of the geographically distributed data centers, to globally minimize the penalty charged to the cloud computing system is proposed in [1]. A new fault tolerant scheduling algorithm MaxRe is proposed in [23]. This algorithm incorporates the reliability analysis into the active replication schema and exploits a dynamic number of replicas for different tasks. Scheduling based genetic algorithm is proposed in [12, 14, 22]. This algorithms optimizes the energy consumption, carbon dioxide emissions and the generated profit of a geographically distributed cloud computing infrastructure. The QoS Min-Min scheduling algorithm is proposed in [10]. An optimized algorithm for VM placement in cloud computing scheduling based on multi-objective ant colony system algorithm in cloud computing is proposed in [8]. Scheduling in grid environment based ACO algorithms are proposed in [13, 14, 19]. The existing scheduling techniques in clouds, consider parameter or various parameters like performance, makespan, cost, scalability, throughput, resource utilization, fault tolerance, migration time or associated overhead. In this paper, cloud task scheduling based ACO approach has been presented for allocation of incoming jobs to VMs considering in our account only makespan to help in utilizing the available resources optimally, minimize the resource consumption and achieve a high user satisfaction.

The basic principles of cloud computing is that to break down the tasks reported by massive users into smaller tasks via the network, by using multiple computers connected in the network to search, compute and combine the results and then send them back to the users. This paper has applied first PSO and then ACO optimization technique to schedule the task at cloud system [27]. Average turnaround time and average cost of overall task scheduling is minimized, as turnaround time and cost of each job is minimized individually. As a result, number of tasks increases, which improves the performance [29]. This paper has focused on grouping task, prioritization and greedy resource allocation. Criteria for calculating cost of every task must not be same, as some tasks are simple, some tasks are complex. Different task have different CPU requirement, memory requirement etc. So, activity based costing is better way of calculating cost of each task, which measures cost of objects and performance of activities and computes cost more accurately [30].Task execution cost can be reduced and user required QoS is improved using load balancing at resource level scheduling [31]. ERUA algorithm [32] satisfy user and cloud service provider through dynamic resource management where utilization ratio must fall under 1, leading to better resource utilization. This paper focus at resource level scheduling. Processing each job individually increases communication cost and time. Because of this communication overhead overall performance of task scheduling increases. But, job-grouping technique groups the small scaled user jobs in job groups which reduces overhead communication time [33]. There are different levels of elasticity structures offered by different cases of data flow structures, operator characteristics and other parameters etc for data flow schedule optimization on cloud [34]. Priority based dynamic resources allocation to tasks scheduling algorithm, which considers multiple SLA objective of job, by preempting best-effort job in cloud environment, is described in [36]. In paper [37], it has been discussed that hierarchical scheduler exploit the multi-core architecture for effective scheduling. They have used diversity of task priority at local and global level for proper load balancing across heterogeneous processors. TDP algorithm is there, where 'T' stands for task selection, 'D' for deadline and 'P' means priority in terms of cost, which selects task according to its constraints and requirements, finally scheduling is done using single priority queue [38].

III. PROPOSED MODEL CONCEPTS

Ant Colony Optimization (ACO) algorithm is a new kind of simulated evolutionary heuristic optimization algorithm and it has been successfully applied to several NP-hard combinatorial optimization problems . The ACO algorithm was proposed by Italian scholar M. Dorigo according to food-seeking behavior of ants in 1996. Here, the ant is a simple computational agent which iteratively constructs a solution for the problem to solve.The ants release some pheromone on the way they move, when these ants reach a crossing they never walked, they will choose a path randomly and release some pheromone proportional to the length of path. The follower ants follows the trail of the other ants to the food source by sensing the pheromone on the path. As this process continues, most of the ant more likely to choose the shortest path with a huge amount of pheromones, on this occasion a mechanism of positive feedback forms, this mechanism ensures that the good information can be preserved and ants can find an optimal way finally[23,24]. In this case, there are more and more pheromones on this path. As the time on, the amount of information on other paths is gradually reduced, eventually the path most ants moves is the optimal path. It is clear that an ACO algorithm can be applied to any combinatorial problem when there is possible to define

1. Problem representation which allows ants to incrementally build/ modify solutions.
2. The heuristic desirability η of edges.
3. A constraint satisfaction method which forces the construction of feasible solutions.
4. A pheromone updating rule which specifies how to modify pheromone trail τ on the edges of the graph.
5. A probabilistic transition rule of the heuristic desirability and of pheromone trail.

In this section, cloud task scheduling based on modified ACO algorithm will be proposed. Our proposed algorithm model applies following concepts:

1. Problem Representation

The problem is represented as a graph $G=(N, E)$ where the set of nodes N represents the VMs and workflows and the set of edges *E represents* mapping of the tasks (or set of tasks i.e. workflow) to selected VMs as shown in Figure . All ants are placed at the starting VMs randomly. During an iteration ants build solutions to the cloud scheduling problem by moving from one VM to another for next task until they complete a tour (all tasks have been allocated). Here figure 1 is showing Workflows in the form of DAGs as assigned to various VMs available at Datacenter .

ISSN 2348 – 8034
Impact Factor- 4.022

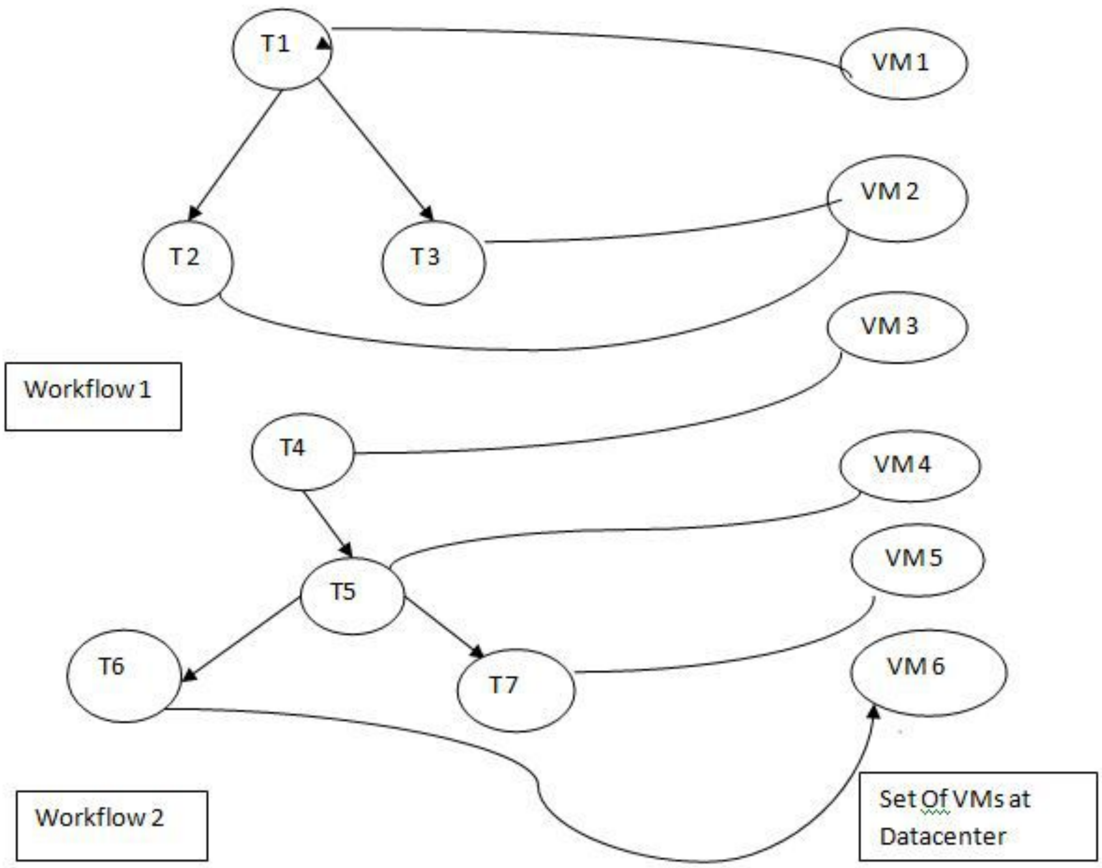

Figure 1. Showing Workflows in the form of DAGs as assigned to various VMs available at Datacenter

2. Heuristic Desirability

The heuristic used here is the inverse of expected execution time of the task *i* on *VM j*.

3. Constraint Satisfaction

The constraint satisfaction method is implemented as a simple, short-term memory of the visited VM, in order to, avoid visiting a VM more than once in one ACO procedure and minimize time of the assigned couplings (task and VM).

4. Pheromone Updating Rule

It is represented by Equations 3, 4, 5, 6 and 7. Pheromone evaporates on all edges and new pheromone is deposited by all ants on visited edges; its value is proportional to the quality of the solution built by the ants.

5. Probabilistic Distribution Rule

The formula for defining the probability distribution at each move makes use of a set $tabu_k$ which indicates set of infeasible moves for ant k . The probabilistic distribution rule, called random proportional, is shown in Equation 1.
The pseudo code of the proposed ACO algorithm and scheduling based ACO algorithm are shown in Algorithms 1 and 2 respectively. Here, The main operations of the ACO procedure are initializing pheromone, choosing VM for next task and pheromone updating as follows:

ISSN 2348 – 8034
Impact Factor- 4.022

Algorithm 1: Modified ACO algorithm
Input: *List of Cloudlet (Tasks) and List of VMs*
Output: *The best solution for tasks allocation on VMs*
Steps:
1. Initialize:
Set Current_iteration_t=1.
Set Current_optimal_solution=null.
Set Initial value τij(t)=c for each path between tasks and VMs.
2. Place m ants on the starting VMs randomly.
3. For k:=1 to m do
Place the starting VM of the k-th ant in $tabu_k$.
Do ants_trip while all ants don't end their trips
Every ant chooses the VM for the next task according to Equation 1.
Insert the selected VM to $tabu_k$.
End Do
4. For k:=1 to m do
Compute the length Lk of the tour described by the k-th ant according to Equation 4.
Update the current_optimal_solution with the best founded solution.
5. For every edge (i, j), apply the local pheromone according to Equation 5.
6. Apply global pheromone update according to Equation 7.
7. Increment Current_iteration_t by one.
8. If (Current_iteration_t < tmax)
Empty all tabu lists.
Goto step 2
Else
Print current_optimal_solution.
End If
9. Return

Algorithm 2: ACO algorithm based Cloudlet Scheduling
Input: *Incoming Cloudlets and VMs List*
Output: *Print "scheduling completed and waiting for more Cloudlets"*
Steps:
1. Set Cloudlet List=null and temp_List_of_Cloudlet=null
2. Put any incoming Cloudlets in Cloudlet List in order of their arriving time.
3. Do ACO_P while Cloudlet List not empty or there are more incoming Cloudlets
Set n= size of VMs list
If (size of Cloudlet List greater than n)
Transfer the first arrived n Cloudlets from Cloudlet List and put them on temp_List_of_Cloudlet
Else
Transfer all Cloudlets.from Cloudlet List and put them on temp_List_of_Cloudlet
End If
Execute ACO procedure with input temp_List_of_Cloudlet and n
End Do
4. Print "scheduling completed and waiting for more Cloudlets"
5. Stop

 ISSN 2348 – 8034
Impact Factor- 4.022

Initializing Pheromone

The amount of virtual pheromone trail **τij(t)** is presented by the edge that connects task **i** to **VMj**. The initial amount of pheromone on edges is assumed to be a small positive constant **τ0** (homogeneous distribution of pheromone at time **t=0**.

VM Choosing Rule for Next Task

During an iteration of the ACO algorithm each ant **k,k=1, ..., m (m** is the number of the ants), builds a tour executing **n** (**n** is number of tasks) steps in which a probabilistic transition rule is applied. The **k**-ant chooses **VMj** for next task **i** with a probability that is computed by Equation 1

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} * [\eta_{ij}]^{\beta}}{\sum_{s \in allowed_k} [\tau_{is}(t)]^{\alpha} * [\eta_{is}]^{\beta}} & if\ j \in allowed_k \\ 0 & otherwise \end{cases} \quad (1)$$

Where, **τij(t)** shows the pheromone concentration at the *t* time on the path between task **i** and **VMj,** allowed **k={0,1,…,n-1}-tabu k** express the allowed VMs for ant **k** in next step and **tabu k** records the traversed VM by ant **k**, and **ηij=1/dij** is the visibility for the **t** moment, calculated with heuristic algorithm and **dij** which expresses the expected execution time and transfer time of the task **i** on **VMj** can be computed with Equation 2.

$$d_{ij} = \frac{TL\_Task_i}{Pe\_num_j * Pe\_mips_j} + \frac{InputFileSize}{VM\_bw_j} \quad (2)$$

Where, ***TL_Task*** $_i$ is the total length of the task that has been submitted to **VM**$_j$, Pe_num$_j$ is the number of **VM**$_j$ processors, Pe_mips$_j$ is the MIPS of each processor of **VM**$_j$**, InputFileSize** is the length of the task before execution and **VM_bw**$_j$ is the communication bandwidth ability of the **VM**$_j$. Finally, the two parameters **α** and **β**in Equation 1 are used to control the relative weight of the pheromone trail and the visibility information respectively

Pheromone Updating Equation

After the completion of a tour, each ant **k** lays a quantity of pheromone _ **τij k(t)** computed by Equation 3 on each edge **(i, j)** that it has used.

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L^k(t)} & if\ (i,j) \in T^k(t) \\ 0 & if\ (i,j) \notin T^k(t) \end{cases} \quad (3)$$

Where, **Tk(t)** is the tour done by ant **k** at iteration **t, L**^{k} **(t)** is its length (the expected makespan of this tour) that is computed by Equation 4 and **Q** is a adaptive parameter.

$$L^k(t) = \arg\max_{j \in J} \{sum_{i \in IJ}(d_{ij})\} \quad (4)$$

Where, **ij** is the set of tasks that assigned to the **VM**$_j$**.** After each iteration pheromone updating which is applied to all edges is refreshed by Equation 5

$$\tau_{ij}(t) = (1-p)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (5)$$

Where, is the trail decay, **0< < 1 and** $_{ij}$**(t)** is computed by Equation 6.

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \qquad (6)$$

When all ants complete a traverse, an elitist is an ant which reinforces pheromone on the edges belonging to the best tour found from the beginning of the trial **(T+)**, by a quantity **Q/L+,** where **L+** is the length of the best tour **(T+)**. This reinforcement is called global pheromone update and computed by Equation 7.

$$\tau_{ij}(t) = \tau_{ij}(t) + \frac{Q}{L^{+}} \quad if\ (i,j) \in T^{+} \qquad (7)$$

The above seven equations are mentioned in paper [45].

IV. EXPERIMENTAL RESULTS & ANALYSIS

The CloudSim toolkit is used demonstrate the simulation. CloudSim (2.1.1) is used for the verification of proposed algorithm. This paper uses CloudSim to simulate a cloud data center and overrides the DatacenterBroker class and the Cloudlet class to realize the simulation of these algorithms above. Besides, this paper designs set of workflow model to test the validity of proposed algorithm. The parameters (**α**, ***β***, ***p***, ***tmax***, ***m*** the number of ants and **Q**) considered here are those that affect directly or indirectly the computation of the algorithm. We tested several values for each parameter while all the others were held constant on 20, 40, 60, 80 and 100 tasks. The default value of the parameters was **α=1, β=1, ρ=0.5, Q=100, tmax=150** and **m=8.**

The comparison of the results obtained from proposed algorithm is done with the FCFS scheduling policy, the Min-Max algorithm, Min-Min task scheduling and PSO(Particle Swarm Optimization) based task scheduling in cloud system. Below, Table 1 shows configuration of VMs and Table 2 is showing performance of proposed methodology over existing algorithms.

This work was supported by the Korea Ministry of Environment (MOE) as ‘‘the Environmental Health Action Program.’’ This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2016011249).

Table 1. Configuration Of VMs

VM	**RAM**	**Processing Power(MIPS)**	**Processing Element**
VM1	5024	22000	1
VM2	1048	11000	1
VM3	3308	22000	1
VM4	4604	32000	1
VM5	8028	55000	1
VM6	4000	41000	1

Comparision Of Makespan(with respect to Time)

The experimental results shows the remarkable improvement in performance over FCFS, Max-Min, Min-Min and PSO algorithms. Table 2 is presenting performance of proposed methodology over existing algorithm. Figure 2 is showing the graph of comparision of makespan with respect to time. The results are showing that proposed algorithm outperforms the existing algorithm.

Table 2. Performance of proposed methodology over existing algorithms

No. of task set	FCFS	Min-Max	Min-Min	Particle Swarm Optimization(PSO)	Proposed Algorithm based on Ant Colony Optimization(ACO)
20	603.6427	366.2881	359.4416	280.6669	202.6671
40	2202.9658	1188.6538	1146.1112	660.5355	481.7792
60	6664.0266	3766.8554	2881.6667	1044.6681	891.5592
80	7241.3959	5044.6669	3331.1161	1846.4401	1458.9923
100	8563.3427	6887.4331	5842.556	3566.6664	2177.9954

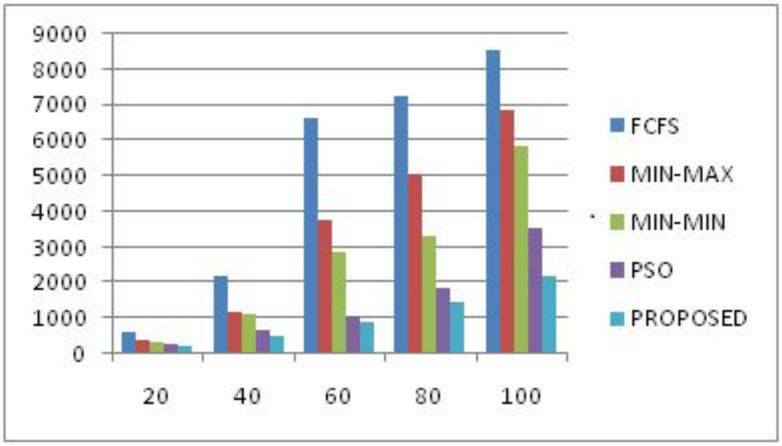

Figure2. Bar Graph showing comparision of proposed ACO scheduling algorithm with existing scheduling techniques.

V. CONCLUSION & FUTURE WORK

Scheduling is a highly critical problem in cloud computing system, because a cloud provider has to serve many users in cloud computing system simultaneously. So scheduling is the major issue in establishing cloud computing systems. In this paper we have discussed about the problems of cloudlet scheduling in computational cloud, where user submits the jobs (requires small processing requirement) and we have tried to find a solution for that problems. Our proposed ACO scheduling algorithm reduced the total processing time of the tasks and also reduced the communication overheads. This algorithm takes time utilization and resource utilization and load balancing into consideration and hence results in high signification. Our proposed ACO algorithm allocated the resources efficiently and optimum solution is obtained.

In the future, there are several points that deserve to be further investigated. First, this paper only presents workflow model with single structure and correlation to verify the advantages of the proposed algorithm, so we will propose different types of workflow models in the future to verify the reliability of the proposed work. Second, this paper only considers minimizing the execution time of tasks and ignores the cost problem existing in reality, so we will realize high efficiency and low cost at the same time

REFERENCES

1. *Boloor K., Chirkova R., Salo T., and Viniotis Y., "Heuristic-Based Request Scheduling Subject to a Percentile Response Time SLA in a Distributed Cloud," in Proceedings of the IEEE International Conference on Global Telecommunications Conference, Florida, USA,pp. 1-6, 2010.*
2. *Bonabeau E., Dorigo M., and theraulaz G.,Swarm Intelligence: From Natural to ArtificialIntelligence, Oxford University Press, NewYork, USA, 1999.*
3. *Buyya R., Ranjan R., and Calheiros N.,"Modeling and Simulation of Scalable CloudComputing Environments and the CloudSimToolkit: Challenges and Opportunities," in Proceedings of the 7th High PerformanceComputing and Simulation Conference, Leipzig,Germany, pp. 1-11, 2009.*
4. *Dorigo M. and Blum C., "Ant Colony Optimization Theory: A Survey," in TheoreticalComputer*

Science, vol. 344, no. 2, pp. 243-278,2005.

5. *Dorigo M., Birattari M., and Stutzel T., "AntColony Optimization," IEEE Computational Intelligence Magazine, vol. 1, no. 4, pp. 28-39,2006.*
6. *Fangzhe C., Ren J., and Viswanathan R.,"Optimal Resource Allocation in Clouds," in Proceedings of the 3rd International Conference on Cloud Computing, Florida, USA, pp. 418- 425, 2010.*
7. *Gao K., Wang Q., and Xi L., "Reduct Algorithm Based Execution Times Prediction inKnowledge Discovery Cloud Computing Environment," the International Arab Journal of Information Technology, vol. 11, no. 3, pp. 268-275, 2014.*
8. *Gao Y., Guan H., Qi Z., Hou Y., and Liu L., "AMulti-Objective Ant Colony System Algorithmfor Virtual Machine Placement in CloudComputing," Journal of Computer and SystemSciences, vol. 79, no. 8, pp. 1230-1242, 2013.*
9. *Ghalem B., Tayeb F., and Zaoui W.,"Approaches to Improve the Resources Management in the Simulator Cloudsim," in Proceedings of the Conference on Interaction and Confidence Building Measures in Asia,Lecture Notes in Computer Science, Istanbul,Turkey, pp. 189-196, 2010.*
10. *Hsu C. and Chen T., "Adaptive Scheduling Based on Quality of Service in Heterogeneous Environments," in Proceedings of the IEEE International Conference on Multimedia and Ubiquitous Engineering, California, USA, pp. 1-6, 2010.*
11. *Ijaz S., Munir E., Anwar W., and Nasir W.,"Efficient Scheduling Strategy for Task Graphs in Heterogeneous Computing Environment," the International Arab Journal of Information Technology, vol 10, no. 5, pp. 486-492, 2013.*
12. *Kessaci Y., Melab N., and Talbi E., "A Pareto-Based GA for Scheduling HPC Applications on Distributed Cloud Infrastructures," in Proceedings of the IEEE International Conference on High Performance Computing and Simulation, Istanbul, Turkey, pp. 456-462, 2011.*
13. *Lorpunmanee S., Sap M., Abdul A., and Chompoo C., "An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment,"in Proceedings of World Academy of Science, English and Technology, 2007.*
14. *Mahamud K. and Nasir H., "Ant Colony Algorithm for Job Scheduling in Grid Computing," in Proceedings of the 4th Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation,Kota Kinabalu, Malaysia, pp. 40-45, 2010.*
15. *Nemhauser G. and Wolsey A., Integer and Combinatorial Optimization, John Wiley and Sons, New York, USA, 1988.*
16. *Paul M. and Sanyal G., "Survey and Analysis of Optimal Scheduling Strategies in Cloud Environment," in Proceedings of the IEEE International Conference on Information and Communication Technologies, Georgia, USA, pp.789-792, 2012.*
17. *Qiyi H. and Tinglei H., "An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing," in Proceedings of the IEEE International Conference on Intelligent Computing and Integrated Systems, Guilin, China, pp. 673-675, 2010.*
18. *Reeves C., Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publishing, Oxford, England, 1993.*
19. *Singh M., "GRAAA: Grid Resource Allocation Based on Ant Algorithm," the Journal of Advances in Information Technology, vol. 1, no.3, pp.133-135, 2010.*
20. *Weiss A., "Computing in the Clouds," Net Worker on Cloud computing: PC functions move onto the web, vol. 11, no. 4, pp. 16-25, 2007.*
21. *Xu M., Cui L., Wang H., and Bi Y., "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing," in Proceedings of the IEEE International Conference on Parallel and Distributed Processing with Applications, Chendu and JiuZhai Valley, China, pp. 629-634, 2009.*
22. *Zhao C., Zhang S., Liu Q., Xie J., and Hu J., "Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing," in Proceedings of the IEEE International Conference on Wireless Communications, Networking and Mobile Computing, China, pp.1-4, 2009.*
23. *Zhao L., Ren Y., Xiang Y., and Sakurai K., "Fault-Tolerant Scheduling with Dynamic Number of Replicas in Heterogeneous Systems," in Proceedings of the IEEE International Conference on High Performance Computing and Communications, Washington, USA, pp.434-441, 2010.*
24. *Zhu K., Song H., Liu L., Gao J., and Cheng G.,"Hybrid Genetic Algorithm for Cloud Computing Applications," in Proceedings of the IEEE International Conference on Asia-Pacific Services*

ISSN 2348 – 8034
Impact Factor- 4.022

Computing Conference, Jeju, Korea, pp.182-187, 2011.

25. *Shengjun Xue, Mengying Li, Xiaolong Xu, and Jingyi Chen, "An ACO-LB Algorithm for Task Scheduling in the Cloud Environment", Journal Of Software, Vol. 9, No. 2, February 2014, Pp.466-473.*
26. *Z. Lu, C Esteve, J Chirivella and P Gagliardo, "A Game Based Assistive Tool for Rehabilitation of Dysphonic Patients". 3rd International Workshop on Virtual and Augmented Assistive Technology (VAAT) at IEEE Virtual Reality 2015 (VR2015), Arles, France, IEEE, (2015).*
27. *XiaoLi He1, Yu Song2 and Ralf Volker Binsack, " The Intelligent Task Scheduling Algorithm in Cloud Computing with Multistage Optimization", International Journal of Grid and Distributed Computing Vol. 9, No. 4 (2016), pp.313-324 http://dx.doi.org/10.14257/ijgdc.2016.9.4.28*
28. *Yogita Chawla and Mansi Bhonsle, "Dynamically optimized costbased task scheduling in Cloud Computing", International Journal,Volume 2, Issue 3 May-June 2013.*
29. *Monika Chaudhary and Sateesh Kumar Peddoju, "A DynamicOptimization Algorithm for Task Scheduling in Cloud Environment" International Journal of Engeenering Research and Application, Vol 2, Issue 3, May-June 2012*
30. *Qi Cao, Zhi-bo Wei and Wen-Mao Gong, "An optimization algorithm for task scheduling based on activity based costing in cloud computing ",Wuhan University ,China 978-1-4244-2902-8/09/$25.00 2009 IEEE.*
31. *Hong Sun, Shi-ping Chen, Chen Jin and Kai Guo, "Research and simulation of task scheduling algorithm in cloud computing"University of Shanghai, China,July 25,2013.*
32. *Ram Kumar Sharma and Nagesh ,Sharma, "A Dynamic optimization algorithm for task scheduling in cloud computing with resource utilization,"IJSET@2013 volume no.2,Issue no.pp:1062-1068 1oct 2013.*
33. *Nithiapidary Muthuvelu, Junyang Liu, Nay Lin Soe,Srikumar Venugopal, Anthony Sulistio and Rajkumar Buyya, "A Dynamicjob grouping-based scheduling for deploying applications with fine-grained tasks on global grids" The Uviversity of Melbourne,Australia ICT building, 111 barry street, carlton VIC 3053,2005.*
34. *Herald Kllapi, Eva Sitaridi and Manolis M. Tsangaris, "Schedule optimization for data processing flows on the Cloud" ,University of Athens ,copyright 2011.*
35. *Vandana Choudhary, Saurabh Kacker, Tanupriya Choudhray and Vasudha Vashisht, "An Approach to improve task scheduling in adecentralized cloud computing environment", IJCTA| jan-feb2012.*
36. *Chandrashekhar S.Pawar and Rajnikant B. Wagh, "Priority based dynamic resource allocation in Cloud computing", Shirpur, India,2011.*
37. *S.V Parikh et al., "Double Level Priority Based Task Scheduling with Energy Awareness in Cloud Computing", Journal of Information, Knowledge and Research in Information Technology, ISSN: 0975 – 6698| NOV 12 TO OCT 13 | VOLUME – 02, ISSUE– 02*
38. *Dr.V.Vaithiyanathan et al., "An Efficient TPD Scheduling Algorithm for Cloud Environment", International Journal of Engineering and Technology (IJET), ISSN : 0975-4024, Vol 5 No 3 Jun-Jul 2013.*
39. *Rep. UCB/EECS, vol. 28, 2009.] J. Geelan, "Twenty-one experts define cloud computing," Cloud Computing Journal, vol. 2009, pp.1-5, 2009.*
40. *S. Singh and K. Kant, "Greedy grid scheduling algorithm in dynamic job submission environment," in Emerging Trends in Electrical and Computer Technology (ICETECT), 2011International Conference on, 2011, pp. 933-936.*
41. *R. N. Calheiros, et al., "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services," Arxiv preprint arXiv:0903.2525, 2009.*
42. *C T Lin*1 et al., "Comparative Based Analysis of Scheduling Algorithms for Resource Management in Cloud Computing", *1Department of Mechanical Engineering University of California,*
43. *Davis, California, Vol.-1, Issue-1, July 2013.*
44. *Maria A. Rodriguez and Rajkumar Buyya, "Deadline basedresource provisioning and scheduling algorithm for scientificworkflows on cloud" 2168-7161 (c) 2013 IEEE.*
45. *Medhat Tawfeek, Ashraf El-Sisi, Arabi Keshk and Fawzy Torkey, " Cloud Task Scheduling Based on Ant ColonyOptimization" The International Arab Journal of Information Technology, Vol. 12, No. 2, March 2015*