International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 3, Issue 2, February 2014

Job Scheduling in Cloud Computing using Ant Colony Optimization

Dr.D.Maruthanayagam,
Head, Department of computer Science,
Siri PSG Arts & Science College for Women, Sankari, Salem.

T. Arun Prakasam,
Assistant Professor, Department of computer Science,
Siri PSG Arts & Science College for Women, Sankari, Salem.

Abstract: Cloud computing is a technology which uses internet and one remote server to maintain data and various applications. Cloud Computing is one of the paradigm in the field of IT. This technology uses the internet and central remote servers to maintain data and applications. There are heterogeneous environment, so, the utilization of resources are accessed and analyzed in real time manner. Allocation of resources to a large number of jobs in a cloud computing environment presents more difficulty than in network computational environments. It aims to find a suitable allocation of resources for each job. Before starting the cloud job allocation, the expected execution time for each task on each machine must be estimated and represented by an Expected Time calculation. Finally we find best completion and makespan time for job scheduling process by using proposed Ant Colony Optimization (ACO). This algorithm is evaluated using the simulated execution times for a cloud environment. Cloud Computing is dynamic in nature. So, prediction based analysis is not possible and performance monitoring of any application in cloud computing is very much important. This paper proposes a scheduling algorithm for proper utilization of the resources and to reduce congestion in cloud environment.

Keywords: Ant Colony Optimization, Cloud computing, resource utilization, job scheduling, Job schedulers.

I. INTRODUCTION

Cloud computing provides on-demand network access to a shared pool of resources. Cloud computing ensures access to virtualized IT resources that are present at the data center, and are shared by others. The data stored in Cloud are simple to use, and are paid for the usage and can be accessed over the internet. These services are provided as a service over a network, and are accessible across computing technologies, operations and business models. Cloud enables the consumers of the technology to think of computing as effectively limitless, of minimal cost, and reliable, to need not know about how it is constructed, its working, its operation, or where it is located [1].

The Cloud scheduler finds out the better resource for a particular job and submits that job to the selected systems. The cloud scheduler does not have control over the resources and also on the submitted jobs. Any machine in cloud can execute any job, but the execution time differs. As compared with the expected execution time, the actual time may vary when running the jobs in the allocated resources. So, the job allocation has been determined according to the scheduling intension and then data move operations have been initiated for necessary task to transfer relevant machines. Scheduling algorithms are used mainly to minimize resource starvation and to ensure fairness amongst the parties utilizing the resources. Scheduling deals with the problem of which resources needed to be allocated for the received task. Many scheduling algorithms are currently available. ACO is the one of most suitable algorithm for scheduling tasks in cloud computing.

In cloud computing, applications are submitted for use of cloud resources by users from their terminals. The resources include computing power, communication power and storage. An application consists of number of tasks; users want to execute these tasks in an efficient manner. There are two possibilities of submission of tasks/data on resources; in one of them, task is submitted on the resources where the input data is available and in the other, on the basis of specific criteria, resource is selected on which both task and input data are transferred and where the task is submitted on a scheduler and data on a resource identified by the scheduler [2, 3].

Show the figure 1 general step of job scheduling in cloud computing. In this figure the main component is scheduler, information repository, resources. In cloud computing a user submits tasks to schedulers. Scheduler is connecting to the information repository it's call Cloud Information Services(CIS). It has all information about to the cloud resources and it also has some virtual privet network for any other don't access to database. Job scheduling techniques (ACO) will employee within the CIS. Scheduler is receiving tasks from user then scheduler is arranging the tasks according to criteria of tasks [4].

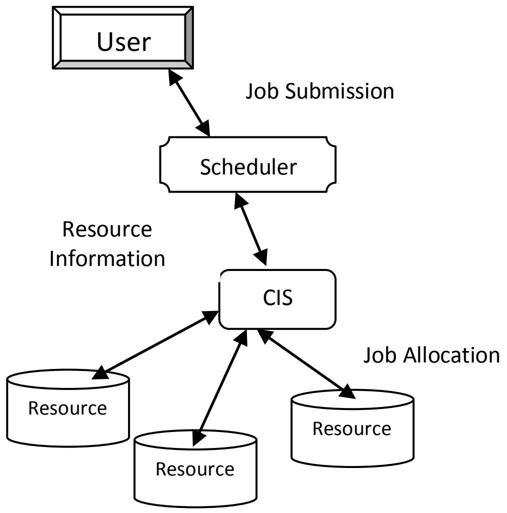

Figure 1: Scheduling Process in Cloud.

Scheduler connects to CIS and gets information about the available resources and then scheduler compute to the resources. After compute the resources, tasks are submit on that resource which have high processing capabilities. The user submits the input data or tasks to the resource via CIS and finally user gets the output from the resource through the scheduler. This is general process of scheduling [5, 6, 7].

II. RELATED WORK

Scheduling performed using several algorithms are studied in the literature recently. Marco Dorigo and Luca Maria Gambardella [8] described an artificial ant colony that was capable of solving TSP. They described about the real and artificial ants. Real ants are capable of finding the shortest path from a food source to the nest without using visual cues. Also, they are capable of adapting to changes in the environment, for example finding a new shortest path once the old one is no longer feasible due to a new obstacle. Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta, Kuwar Pratap Singh, NItin and Ravi Rastogi [9] discussed about the algorithm for load distribution of workloads among nodes of a cloud by the use of Ant Colony Optimization (ACO). This is a modified approach of ant colony optimization that has been applied from the perspective of cloud or cloud network systems with the main aim of load balancing of nodes.

Sarayut Nonsiri and Siriporn Supratid [10] discussed about the ACO that allows fast near optimal solutions to be found. It is useful in industrial environments where computational resources and time are limited. Like several search methods, being trapped in nontrivial difficulty for ACO. Nidhi Jain Kansal and Inderveer Chana [11] discussed various types of Load Balancing Techniques in Cloud Computing like Vector Dot, CARTON, Compare and Balance, Honeybee Foraging Behavior etc. Birattari, Dorigo and Stutzle [12] discussed about ant colony optimization. They also discussed about parallelization and parallel population based meta-heuristic.

III. PROPOSED WORK

Ant Colony Optimization (ACO) algorithm aims to allocate tasks in to the available resources in the cloud computing environment. The cloud scheduler's aim is to allocate the jobs to the available nodes. The best match must be found from the list of available jobs to the list of available resources. The selection is based on the prediction of the computing power of the resource. The cloud scheduler must allocate the jobs to the resources efficiently. The efficiency depends upon two criteria; one is makespan and the other is flow time. The tasks available are compared with the available resources and the best is selected. The execution times from the simulation are used for evaluating ant based algorithm in cloud environment.

The processing time matrix is used to represent the processing time expected for every task on every resource; this is done before cloud scheduling is started. The rows of the ET Matrix represent the execution time estimates of every resource for a job while column represents the execution time estimates of a specific resource of all jobs in the job pool. ET_{ij} is the expected execution time of task T_i and the resource R_j. The time to move the executables and data associates with the task T_i includes the expected execution matrix ET_{ij}.

The algorithm assumes inter-task communication; the algorithm presupposes processing times of every task of every resource where one resource is used by each task. The ET matrix is represented as N x M matrix, where N represents the independent jobs needed to be scheduled where as M represents the available resources. Each job's workload is measured by millions of instructions and the capacity of each resource is measured by MIPS. Here the algorithm, R denotes the expected time which resource R_j will become ready to execute a task after finishing the execution of all tasks assigned to it. First, the C_{ij} entries are computed using the ET_{ij} (the estimated execution time of task T_i on resource R_j) and R_j values.

$$C_{ij}=ET_j+R_j \qquad \rightarrow(1)$$

Specification of the resources allocation according to instructions and data (MIPS) completion time of the tasks on each of the resources .Tasks/Resources R1, R2 and R3 four tasks T1, T2, T3 and T4 are in the meta-task M_v and the Cloud scheduler is supposed to schedule all the tasks within M_v on three resources R1, R2 and R3.

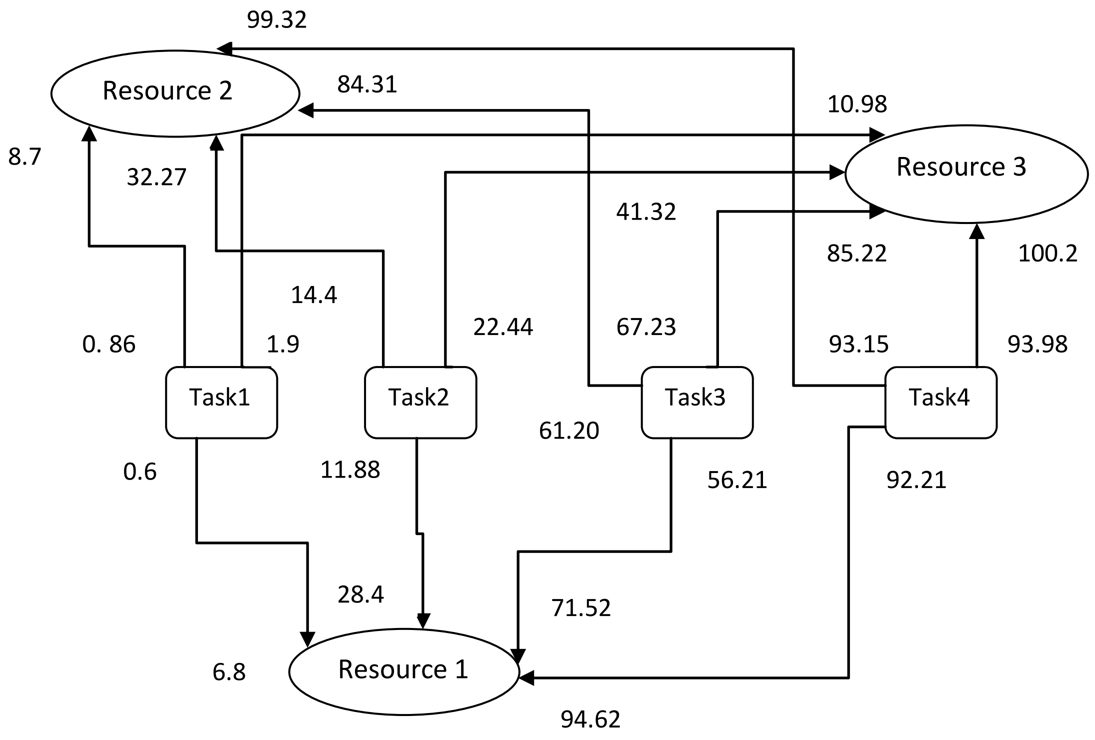

Figure 2: Timing Allotment for Tasks

Table 1 is shown the specification of the resources and tasks. Here involved of three machines and four working process than the workload and timing allotment expected are shown in figure 2.

Expected Time	Ti \ Rj	T1	T2	T3	T4
Ready Time	R1	0.68	11.88	56.21	92.21
	R2	0.86	14.48	61.20	93.15
	R3	1.98	22.44	67.23	93.98
Completi on Time	R1	6.88	28.40	71.52	94.62
	R2	8.76	32.27	84.31	99.32
	R3	10.98	41.32	85.22	100.27

Table 1: Specification of the Resources and Tasks.

Proposed -Ant Colony Algorithms (ACO)

Step 1: Collect information about the jobs (n) and resources (m) of the system in ET matrix
Compute approximate $C_{ij}=ET_j+R_j$ to ant's resource allocate end for

Step 2: Set all the initial value ρ = 0.05(pheromone evaporation value) to = 0.01(initial pheromone deposit value)
Free [0.. m-1] = 0(one dimensional matrix of size m)
k = m(number of ants= no. of tasks)

Step 3: Do until all tasks in M_v are mapped (to prepare the scheduling list) do the following steps 4 and 5 for each tasks T_i and R_j

Step 4: Select the Task (i) and Resource (j) randomly.

Step 5: Repeat the following until all jobs are executed.

```
        If the number of resources is even then
            Find the resource free times
        Calculate the heuristic information ($\eta_{ij}$)
                $\eta_{ij}$ =1/Free(j)
Calculate current pheromone trail value
        $T_{ij}$=1-$\rho$/Fk
        Where Fk = max (free (j));
Update the Pheromone Trail Matrix
        $T_{ij}$ = 1/$ET_{ij}$
Calculate the Probability matrix
        $P_{ij} = T_{ij}\, \eta_{ij} / \sum T_{ij}\, \eta_{ij}$
Update Probability matrix with various criteria,
    $Pij = \tau_{ij} . \eta_{ij} (1/ET_{ij}) / \Sigma \tau_{ij} . \eta_{ij} (1/ET_{ij})$
Running Compute Time,
    $Pij = \sum Tij.ETij + (MaxCT - \min CT)$
Makespen time with ETij
    $P_{ij} = \sum 1/ETij$ *(MaxET-MinET) *$T_{ij}$
Makespen time with CTij
    $P_{ij}=1-(\sum ETij * (MaxET - MinET))$ + (MinCT *$T_{ij}$)
```

Step 6: Find out the best feasible solution by analyzing of all the ants scheduling list.

Before starting of the algorithm, identify the number of tasks (T_i) and number of available resources (R_j) to assign the task. Makespan is used to measure the throughput of the cloud system. In general the existing heuristic mapping can be divided into two categories. One is on line mode and the other one is batch mode. In the on line mode, the scheduler is always in ready mode. Whenever a new job arrives to the scheduler, it is immediately allocated to one of the existing resources required by that job. Each job is considered only once for matching and scheduling. In the batch mode, the jobs and resources are collected and mapped at prescheduled time. In this mode, it takes better decision because the scheduler knows details of the available jobs and resources. This proposed algorithm is also a heuristic algorithm for batch mode. The results of the algorithm have four values (task, machine, starting time, expected completion time).

Calculate the heuristic information (η_{ij}):

A heuristic value, η_i associated with components, η_{ij} associated with connections between resource i and resource j. In many cases

η is the cost, or an estimate of the cost, of adding the component or connection to the solution under construction. The heuristic information η_{ij} is typically inversely proportional to the distance between resources i and j, a straightforward choice being.

$$\eta_{ij} = 1/\text{Free}(_j) \rightarrow (2)$$

The new value of free ($_j$) is the starting time plus ET_{ij}. A heuristic function is used to find out the distance of best resources.

Calculate pheromone trail value (T_{ij}):

Pheromone trail value can be used to learn an appropriate order for task assignments. While ants build a solution of the scheduling, they visit all resources and change their pheromone level which is used immediately to locally update the rule. The local update rule reduces the convergence because ants choose new resource based on high pheromone level therefore, this resource becomes less desirable for the following ants, if the pheromone trail is reduced. The highest priority machine is found which is free earlier. Here four ants are used. Each ant starts from random resource and task (they select ET_{ij} randomly j^{th} resource and i^{th} job). All the ants maintain a separate list. Whenever they select next task and resource, they are added into the list. At each iteration, the ants calculate the new pheromone level of the elements of the solutions is changed by applying following updating rule,

$$T_{ij} = 1 - \rho / F_k \rightarrow (3)$$

Update the Pheromone Trail Matrix,

$$T_{ij} = 1/ET_{ij} \rightarrow (4)$$

Where F_k = max (free (j)); At each iteration the ants calculate the minimized function 'f_k ' for $_k$th ant. Whenever they select next task and resource, they are added into the list.

Calculate the probability matrix:

Using the random proportional rule current ant in resource i choose to go to resource j with probability P_{ij}. In this algorithm two set of tasks are maintained. One is set of scheduled tasks and the other is set of arrived and unscheduled tasks. The algorithm starts automatically, whenever the set of scheduled jobs become empty. The first task to be performed and the machine in which it is performed are chosen randomly. Next, the task to be run and the machine in which it is to be run. In next section we compute the formula of probability matrix.

V.RESULTS AND DISCUSSIONS

The number of ants used is less than or equal to the number of tasks. From all the possible scheduling lists find the one having minimum makespan and uses the corresponding scheduling list. Here two kinds of ET matrices are formed, first one consists of currently scheduled jobs and the next consists of jobs which have arrived but not scheduled. The scheduling algorithm is executed periodically. At the time of execution, it finds out the list of available resources (processors) in the cloud environment, form the ET matrix and start scheduling. When all the scheduled jobs are dispatched to the corresponding resources, the scheduler starts scheduling over the unscheduled task matrix ET. This is guaranteed that the machines will be fully loaded at maximum time.

The P_{ij} 's value has been modified to include the ET_{ij} and CT_{ij} is modified to the following equation and also probability makespan time values are changed. Various Updated Probability matrix are formed for improving makespan time,

$$P_{ij} = \tau_{ij} \,.\, \eta_{ij} (1/ET_{ij}) / \Sigma\, \tau_{ij} \,.\, \eta_{ij} (1/ET_{ij}) \rightarrow (5)$$

Running Compute Time,

$$P_{ij} = \sum Tij.ETij + (MaxCT - MinCT) \rightarrow (6)$$

Makespen time with ETij

$$P_{ij} = \sum 1/ETij * (\text{MaxET-MinET}) * T_{ij} \rightarrow (7)$$

Makespen time with CTij,

$$P_{ij} = 1 - (\sum CTij * (MaxET - MinET)) + (\text{MinCT} * T_{ij}) \rightarrow (8)$$

Where,

- η_{ij} is the attractiveness of the move as computed by some heuristic information indicating a prior desirability of that move
- T_{ij} is the pheromone trail level of the move, indicating how profitable it has been in the past to make that particular move(it represents therefore a posterior

indication of the desirability of that move)

- P_{ij} is the probability to move from a state *i* to a state *j* is depending on the combination of above two values.

- MinET & MaxET –Minimum and Maximum value of ET_{ij} .
- MinCT & MaxCT –Minimum and Maximum value of CT_{ij}

The inclusion of ET_{ij} execution time of the $_i$th job by the jth machine (predicted) in the calculation of probability, that the $_j$th machine will be free, has shown a positive result in performance improvement. This improvement is in terms of the decrease in makespan time. Furthermore, instead of adding CT_{ij}, execution time of the i^{th} job by the jth machine (predicted), in the calculation of probability, the (8) formula produces better results. In addition, here probability makespan time was compute with various criteria, Formula (6), running compute time involved minimum and maximum values of CT_{ij} with Sum of $T_{ij}.ET_{ij}$. Formula (7), Makespan time in ET_{ij} involved minimum and maximum values of ET_{ij} with sum of $1/ET_{ij}$ and T_{ij}. Table 2 was shown the performance of probability values.

Table 2: Performance Improvements

Probability Makespan time P_{ij}				
Iterations	**P_{ij}**	**Computing P_{ij}**	**Makespan P_{ij} of ET_{ij}**	**Makespan Pij of CT_{ij}**
[0] [0]	0.87377273	0.98644763	1.0816979	0.87938332
[0] [1]	0.49999974	0.37401301	1.01036284	0.50797044
[0] [2]	0.33333307	0.19086036	0.80692408	0.33891651
[1] [0]	0.24999957	0.02886738	0.31049583	0.24543826
[1] [1]	0.1999997	0.02119795	0.27427265	0.20216375
[1] [2]	0.16666643	0.01264867	0.21442464	0.16945825
[2] [0]	0.14285693	0.00437975	0.12329532	0.13988579
[2] [1]	0.12499983	0.00304351	0.10497898	0.12619328
[2] [2]	0.11111097	0.00294287	0.10396651	0.11297217
[3] [0]	0.09999989	0.00250222	0.09359033	0.09781833
[3] [1]	0.09090899	0.00217643	0.08911375	0.09172445
[3] [2]	0.08333325	0.00211135	0.08836168	0.08472912

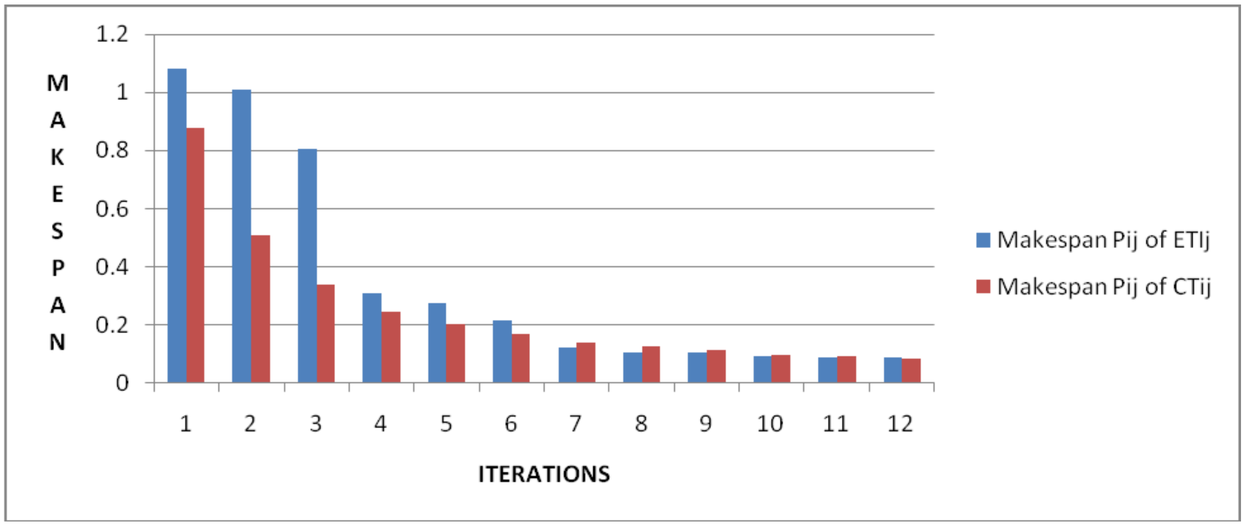

Figure 3: Makespan Comparison

The ant algorithm can be improved using some form of search algorithm. Apply this optimum technique to the output of the ant algorithm. In this method first find the problem resource those with total execution times equal to the makespan of the solution, and attempt to move or swap set of jobs from the problem processor to another resource that has the minimum makespan as compared with all other resources. This is shown in figure 3. After applying the above Proposed ACO technique, find out the problem resource and minimum makespan resource again, swap or move some of the jobs from the minimum makespan resource to problem resource. The search is performed on each problem processor and continues until there is no further improvement in the fitness value of the solution. Finally reach best resource for job scheduling in Cloud Computing.

VI. CONCLUSION

Scheduling is a critical problem in cloud computing, because a cloud provider has to serve many users in cloud computing system. So scheduling is the major issue in establishing cloud computing systems. In this paper we have discussed about those problems of job 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, processing cost and also reduced the communication overheads. This algorithm takes time utilization and resource utilization into consideration and hence results in high signification. Our proposed ACO algorithm allocated the resources efficiently as well as optimum solution is obtained.

REFERENCES

1.S.Ravichandran, Dr. E.R. Naganathan, “Dynamic Scheduling of Data Using Genetic Algorithm in cloud computing”, International Journal of Computing Algorithm, Vol 02, Issue 01, June 2013.

2. N.Muthuvelu, J.Liu, N.Soe, S.Venugopal, A.Sulistio and R.Buyya, “A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Clouds”, in Proc of Australasian Workshop on Cloud Computing and e-Research (AusCloud2005), Vol. 44,2005, pp: 41-48.

3. Monika Choudhary and Sateesh Kumar Peddoju, “A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment”, International Journal of Engineering Research and Applications

(IJERA), ISSN: 2248-9622, Vol. 2,Issue 3, May-Jun 2012, pp: 2564-2568.

4. O. M. Elzeki, M. Z. Rashad and M. A. Elsoud, "Overview of Scheduling Tasks in Distributed Computing Systems", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231 -2307, Vol.-2, Issue-3, July 2012, pp: 470-475.

5.Hai Zhong, Kun Tao and Xuejie Zhanand, "An Approach to Optimized Resource Scheduling Algorithm for Opensource Cloud Systems", ",IEEE in Fifth Annual China Cloud Conference , ISSN: 978 -0-7695-4106-8, Vol.-5, 2010,pp: 124-129.

6.Pardeep Kumar and Amandeep Verma," Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Vol 2, Issue 5, May 2012,pp:111-114.

7.Suraj Pandey, LinlinWu and Siddeswara Mayura Guru, "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments", 24th IEEE International Conference on Advanced Information Networking and Applications, ISSN: 1550-445X, Vol.-3, 2010, pp: 400-407.

8.Marco Dorigo, Luca Maria Gambardella "Ant colonies for the traveling salesman problem", TR/IRIDIA, Vol.3,Université Libre de Bruxelles,Belgium,1996.

9.Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta, Kuwar Pratap Singh, Nitin and Rastogi "Load Balancing of Nodes in Cloud Computing Using Ant Colony Optimization" IEEE, 14th International Conference on Modelling and Simulation. pp. 03-08, 2012.

10.Sarayut Nonsiri and Siriporn Supratid "Modifying Ant Colony Optimization" , IEEE Conference on Soft Computing in Industrial Application, Muroran, Japan. pp. 95-100. 2008.

11. S.L. Ho, Shiyou Yang, Guangzheng Ni, and Jose Marcio Machado "A Modified Ant Colony Optimization Algorithm Model on Tabu-Search Methods" IEEE Transaction on Magnetics, Vol. 42, pp. 1195-1198, 2006.

12. Dorigo, M., Birattari, M. and Stutzle, T. "Ant colony optimization" Iridia technical report series, Report no. TR/IRIDIA, pp. 1-23, 2006.

Authors Profile

Dr.D.Maruthanayagam received his Ph.D Degree from Manonmanium Sundaranar University,Tirunelveli in the year 2014. He has received his M.Phil, Degree from Bharathidasan University, Trichy in the year 2005. He has received his M.C.A Degree from Madras University, Chennai in the year 2000. He is working as a Head, Department of Computer Science, Siri PSG Arts & Science College for Women,Sankari,Salem,Tamilnadu,India. He has 12 years of experience in academic field. He has published 7 International Journal papers and 12 papers in National and International Conferences. His areas of interest include Grid Computing, Cloud Computing and Mobile Computing.

T.Arun Prakasam received his M.Phil(C.S) Degree from Periyar University in the year 2007. He has received his M.Sc (C.S), Degree from Periyar University, Salem in the year 2006. He is working as Assistant Professor, Department of Computer Science, Siri PSG Arts & Science College for Women, Sankari, Salem, and Tamilnadu, India. His areas of interest include Software Engineering, Distributed Computing, Gird and Cloud Computing.