ISSN 2277-3061

MODIFIED ANT COLONY OPTIMIZATION FOR WORKFLOW SCHEDULING IN CLOUD ENVIRONMENT

S.J.Mohana[1], Dr.M.Saroja[2], Dr.M.Venkatachalam[3]
[1] Dept. of Computer Applications, [2,3]Dept. of Electronics
[1,2,3] Erode Arts and Science college, Erode, India.

ABSTRACT

Cloud computing is a type of parallel and distributed system consisting of a collection of interconnected and virtual computers. This technological trend has enabled the realization of a new computing model called cloud computing, in which shared resources, information,software & other devices are provided according to client requirement at specific time, are provided as general utilities that can be leased and released by users through the Internet in an on-demand fashion.Cloud workflow scheduling is an NP-hard optimization problem, and many meta-heuristic algorithms have been proposed to solve it.Allocation of resources to a large number of workflows in a cloud computing environment presents more difficulty than in network computational environments.A good task scheduler should adapt its scheduling strategy to the changing environment and the types of tasks. In this work, modified ant colony optimization for cloud task scheduling is proposed. The goal of modification is to enhance the performance of the basic ant colony optimization algorithm and optimize the task execution time in view of minimizing the makespan of a given tasks set.

Keywords

Cloud computing: workflow scheduling; ant colony optimization; modified ant colony optimization; virtual machines

1. 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. The Cloud scheduler finds out the better resource for a particular workflow and submits that flow to the selected systems. The cloud scheduler does not have control over the resources and also on the submitted workflow. Any machine in cloud can execute, but the execution time differs. As compared with the expected execution time, the actual time may vary when running the workflow in the allocated resources. So, the workflow 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. Ant Colony Optimization (ACO) is the one of most suitable algorithm for scheduling tasks in cloud computing.

ACO takes inspiration from the foraging behavior of some ant species. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. Ant colony optimization exploits a similar mechanism for solving optimization problems. In the early nineties, when the first ant colony optimization algorithm was proposed, ACO attracted the attention of more researchers and a relatively large amount of successful applications are now available. Moreover, a substantial corpus of theoretical results is becoming available that provides useful guidelines to researchers and practitioners in further applications of ACO. In ACO algorithms the ants sample the problem's solution space by repeatedly applying a stochastic decision policy until a feasible solution of the considered problem is found. The sampling is realized concurrently by a collection of differently instantiated replicas of the same ant type. Each ant "experiment" allows to adaptively modifying the local statistical knowledge on the problem structure. The algorithm is recursive in nature. However, the limitation of ACO algorithm is every ant build their own individual result set and it is later on built into a complete solution, long searching time, local optimum and the stagnation phenomenon. Moreover, the capability in a cloud computing environment is full of diversity. The ants in our proposed algorithm will continuously originate from the head node. These ants traverse the width and length of the network in such a way that they know about the location of under loaded or overloaded nodes in the network. These ants along with their traversal will be updating a pheromone table, which will keep a tab on the resources utilization by each node. Many meta-heuristic algorithms have been proposed such as ant colony optimization (ACO) algorithm which is appropriate for dynamic cloud workflow scheduling. In this research, a Modified Ant Colony Optimization (MACO) for cloud workflow scheduling is proposed. The main goal of MACO is to enhance the performance of ACO algorithm.The MACO algorithm inherits the basic ideas from ACO algorithm to decrease the computation time of workflow executing.

The organization of this work is as follows. Section 2 presents some of the related work in thisdirection. Section 3 describes the research methodology for cloud workflow allocation. In section 4, the implementation and simulation results are presented. Finally, section 5 concludes thework.

2. LITERATURE SURVEY

Zheng et al (2011) investigate the possibility to place the Virtual Machines in a flexible way to improve the speed of finding the best allocation on the premise of permitting the maximum utilization of resources. Our scheduling policy achieved by Parallel Genetic Algorithm which is much faster than traditional Genetic Algorithm.Li et al (2011) introduced dynamic cloud scheduling use cases where these parameters are continuously changed, and propose a linear integer programming model for dynamic cloud scheduling. Our model can be applied in various scenarios through selections of corresponding objectives and constraints, and offers the flexibility to express different levels of migration overhead when restructuring an existing infrastructure. The novel cloud scheduling scheme uses SLA(Service Level Agreement) along with trust monitor to provide a faster scheduling of the over flooding user request with secure processing of the request which is given by Daniel et al (2011). The schedule gives improved fast schedule time and low rescheduling ratio than grid scheduling. The proposed schedule scheme provides advantages and better performance for both the cloud user and service provider.Garg et al (2011) propose near-optimal scheduling policies that exploit heterogeneity across multiple data centers for a Cloud provider. We consider a number of energy efficiency factors (such as energy cost, carbon emission rate, workload, and CPU power efficiency) which change across different data centers depending on their location, architectural design, and management system.

Ahn et al (2012) shows that live VM migration can be used to mitigate the contentions on micro-architectural resources. Such cloud-level VM scheduling can widen the scope of VM selections for architectural shared resources beyond a single system, and thus improve the opportunity to further reduce possible conflicts. This work proposes and evaluates two cluster-level virtual machine scheduling techniques for cache sharing and NUMA affinity, which do not require any prior knowledge on the behaviors of VMs. AR leases are most privileged leases with "AR preempts other leases" policy, since they can preempt & suspend other BE leases when demanded by consumers. This leads to two problems: 1) a set of BEs can go suspended infinite number of time & 2) ARs, at the actual time of their resource allocation can be rejected due to presence of other ARs in schedule. Shrivastava and Bhilare (2012) proposed two algorithms 1) Starvation-Removal and 2) AR-to-BE Conversion to solve these problems. In order to better serve cloud's objective of providing low-cost and on-demand services,a two-level and fuzzy clustering based cloud scheduling algorithm named FCTLBS(fuzzy clustering and two level based task scheduling algorithm) was proposed by Li et al (2012).Scheduling was divided into two levels: user scheduling and task scheduling.Shi and Ying (2011) designed an economic scheduling model with business parameters. And a dynamic scheduling algorithm was presented, which made a trade-off between economic effectiveness and performance.

Li et al (2012) propose two online dynamic resource allocation algorithms for the IaaS cloud system with preempt able tasks. Our algorithms adjust the resource allocation dynamically based on the updated information of the actual task executions.Nishant et al (2012) proposed an 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 grid network systems with the main aim of load balancing of nodes.Abrishami et al (2013) adapt the PCP algorithm for the Cloud environment and propose two workflow scheduling algorithms: a one-phase algorithm which is called IaaS Cloud Partial Critical Paths (IC-PCP), and a two-phase algorithm which is called IaaS Cloud Partial Critical Paths with Deadline Distribution (IC-PCPD2). Both algorithms have a polynomial time complexity which makes them suitable options for scheduling large workflows. Tsai et al (2014) begins with a brief retrospect of traditional scheduling, followed by a detailed review of metaheuristic algorithms for solving the scheduling problems by placing them in a unified framework. Armed with these two technologies, this paper surveys the most recent literature about metaheuristic scheduling solutions for cloud. In addition to applications using meta-heuristics, some important issues and open questions are presented for the reference of future researches on scheduling for cloud.

Mateos et al (2013) present a new Cloud scheduler based on Ant Colony Optimization, the most popular bio-inspired technique, which also exploits well-known notions from operating systems theory. Simulated experiments performed with real PSE job data and other Cloud scheduling policies indicate that our proposal allows for a more agile job handling while reducing PSE completion time.Scheduling independent tasks on it is more complicated. In order to utilize the power of grid completely, we need an efficient job scheduling algorithm to assign jobs to resources in a grid. Chang et al (2012) propose an Adaptive Scoring Job Scheduling algorithm (ASJS) for the grid environment. Compared to other methods, it can decrease the completion time of submitted jobs, which may compose of computing-intensive jobs and data-intensive jobs. Pacini et al (2014) surveys SI-based job scheduling algorithms for bag-of-tasks applications (such as PSEs) on distributed computing environments, and uniformly compares them based on a derived comparison framework. We also discuss open problems and future research in the area.

3. RESEARCH METHODOLOGY

This section describes about the research methodology for workflow scheduling in cloud environment.

3.1 Ant Colony Optimiztion

The basic idea of ACO is to simulate the foraging behavior of ant colonies. When an ants group tries to search for the food, they use a special kind of chemical pheromone to communicate with each other. Workflow scheduling based ACO algorithm is used to decrease the computation time of tasks. In ACO, 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 has been allocated). The scheduling of cloud architecture is dynamic in nature and moreover grid middleware and applications are using local scheduling and data co-scheduling. The approach of replication has been also applied and assisted in scheduling and optimization of replication. ACO algorithm can be interpreted as parallel replicated Monte Carlo (MC) systems. Analogously, in ACO algorithms the ants sample the problem’s solution space by repeatedly applying a stochastic decision policy until a feasible solution of the considered problem is found. The sampling is realized concurrently by a collection of differently instantiated replicas of the same ant type. Each ant “experiment” allows to adaptively modifying the local statistical knowledge on the problem structure. The algorithm is recursive in nature. However, the limitation of ACO algorithm is every ant build their own individual result set and it is later on built into a complete solution, long searching time, local optimum and the stagnation phenomenon. Moreover, the capability in a cloud computing environment is full of diversity.

3.2 Modified ACO for Workflow Scheduling

The ants in our proposed algorithm will continuously originate from the head node. These ants traverse the width and length of the network in such a way that they know about the location of under loaded or overloaded nodes in the network. These ants along with their traversal will be updating a pheromone table, which will keep a tab on the resources utilization by each node. We also proposed the movement of ants in two ways similar to the classical ACO, which are as follows:

Forward movement: The ants continuously move in the forward direction in the cloud encountering overloaded node or under loaded node.

Backward movement: If an ant encounters an overloaded node in its movement when it has previously encountered an under loaded node then it will go backward to the under loaded node to check if the node is still under loaded or not and if it finds it still under loaded then it will redistribute the work to the under loaded node. The vice-versa is also feasible and possible. The main task of ants in the algorithm is to schedule the workflow among the nodes. This work introduced update pheromone function for proposed modified ACO. Figure 4.4 shows the overall flow chart of this modified ACO algorithm.

3.2.1. Coding: The length of chromosome is set as the number of tasks *n*, and the genes on chromosome denote the sequence of task/resource pairs. For example, chromosome {1, 2, 5,…., m} means that task T_1 is assigned to R_1, T_2 is assigned to R_1, T_3 is assigned to R_2, T_4 is assigned to R_5 and lastly T_n is assigned to R_m. The set of tasks assigned to R_j is denoted as Y_j={T_1, T_2, T_i}.

3.2.2. Initialization: As mentioned before, the number of computing capability is different. Therefore, initialize the pheromone of as equation (4.3):

$$\tau_j(0) = c_j \times MIPS_j \quad (1)$$

Where c_j is the number of cores on R_j and $MIPS_j$is the computing capability

3.2.3. Vaccine extraction: In this step, ants with desirable qualities are acquired, and extracted as vaccine at some probability. The vaccine is used to improve the diversity of the population, in order to avoid local optimal solutions. The probability of ant *x* on resource *i* being selected as vaccine by equation (4.4)

$$p(x_i) = \frac{\sum_{j=1}^{m}|S_i - S_j|}{\sum_{j=1}^{m}\sum_{i=1}^{m}|S_i - S_j|}, i = 1,2,\dots,m, \quad (2)$$

Where $S_j=\sum_{i-1}^{m} S_{ij}$ denotes the total time of all tasks executed on resource *j*.

3.2.4. Transitivity: The transfer probability at iteration *t* is calculated as equation (4.5). In order to avoid the local optimal solution, introduce a threshold of pheromone. That is, if the amount of pheromone $\tau_{ij}(t)$ is no more than τ_0, the algorithm does not consider the influence from other nodes. Besides, adjust μ if there is no improvement of the local solutions within *N* iterations:

$$\mu_{t+1} = \begin{cases} 0.95\mu_t, & if\ 0.95\,\mu_t \geq \mu_{min}; \\ \mu_{min}, & otherwise \end{cases} \quad (3)$$

Where μ_{min} is the minimum value of μ.

3.2.5. Immune: Perform immune operation by integrating the extracted vaccine into the gene coding.

3.2.6. Update Pheromone: Let $\tau_j(t)$ be the amount of pheromone on resource *j* at time *t*, shown in equation (4.6)

$$\tau_j(t+1) = (1-\mu)\tau_j(t) + \mu\Delta\tau_j \quad (4)$$

Where $\mu \in (0, 1)$

ISSN 2277-3061

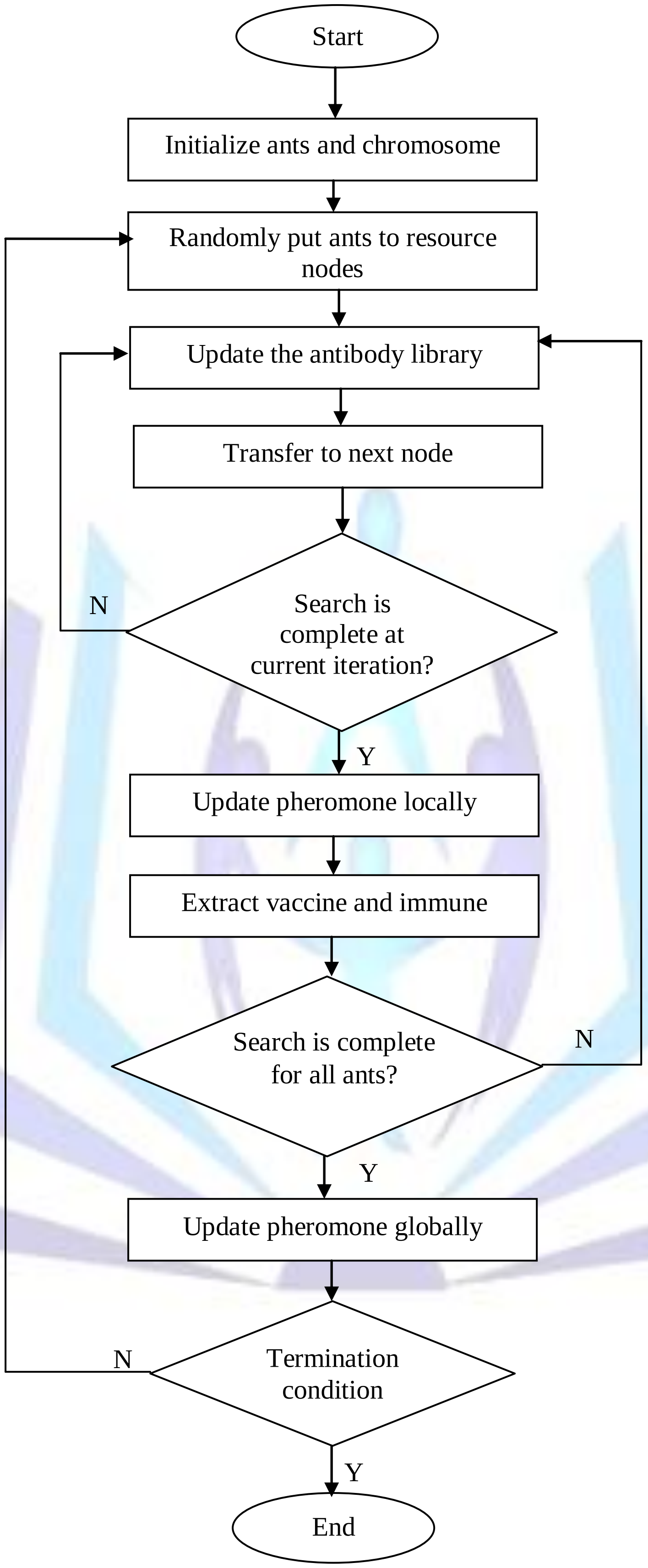

Figure 1: Flowchart for modified ACO

This is a modified approach of ant colony optimization that has been applied from the perspective of cloud systems with the main aim of scheduling the workflows. The main benefit of this approach lies in its detections of overloaded and under loaded nodes and thereby performing operations based on the identified nodes. This simplistic approach elegantly performs our task of identification of nodes by the ants and tracing its path consequently in search of different types of nodes. We have used the same concepts of ant colony optimizations and have only modified the concepts where immune operation is introduced. In our approach the ants continuously update a single result set rather than updating their own result set. In this way, the solution set is gradually built on and continuously improved upon rather than being compiled only once in a while. The other advantage of the approach lies in the fact that the task of each ant is specialized rather than being general and the task depends on the type of first node that was encountered whether it was overloaded or under loaded.

4. EXPERIMENTAL RESULTS

This section presents the metric of comparison, the experimental data, the results and result analysis. As a measure of performance, here makespan, scheduling time, deadline hit are used. We computed the metrics using two heuristics ant colony optimization and modified ant colony optimization. Simulations results demonstrate that number of particles obtain the better solution since more solutions were generated. Initially, test runs were based on the following parameters such as makespan, scheduling time, deadline hit. Each task in the workflow has input and output files of varying sizes.

Table 1: Comparison of workflow scheduling algorithm using various parameters

Scheduling Algorithm	Resource Matrix	Makespan (Seconds)	Scheduling Time (Seconds)	Dead Line Hit (%)
ACO	128 x 8	25	24	58
Modified ACO		20	17	63
ACO	256 x 16	32	30	69
Modified ACO		26	23	77
ACO	512 x 32	46	38	72
Modified ACO		31	30	85
ACO	1024 x 64	52	44	76
Modified ACO		39	39	96

Table 1 shows the comparison of scheduling algorithm using parameters like makespan, scheduling time, dead line hit. The proposed modified ACO has high deadline hit, resource utilization, and less makespan, scheduling time.

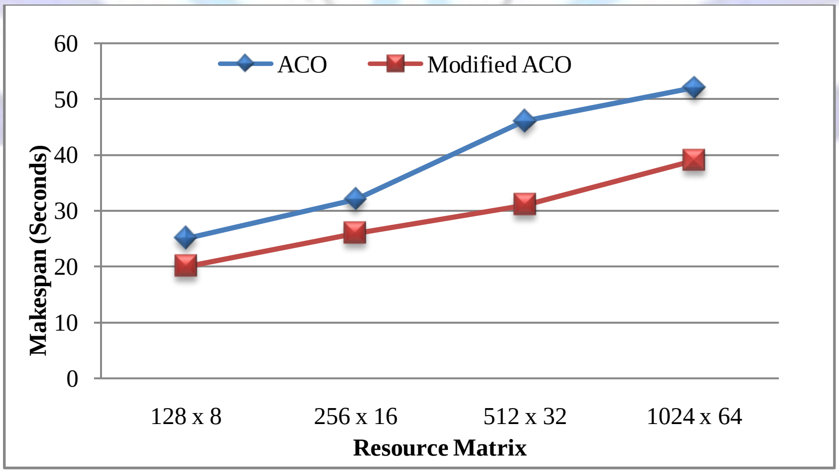

Figure 2: Comparison for makespan

Figure 2 shows the comparison of makespan for ACO and modified ACO. The proposed method of modified ACO has less makespan than ACO technique.

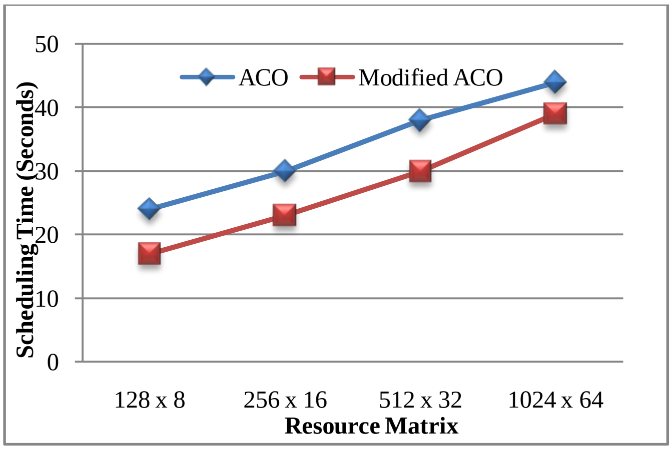

Figure 3: Comparison for Scheduling Time

Figure 3 shows the accuracy for ACO and modified ACO. The proposed method of modified ACO has less scheduling time when compare with ACO.

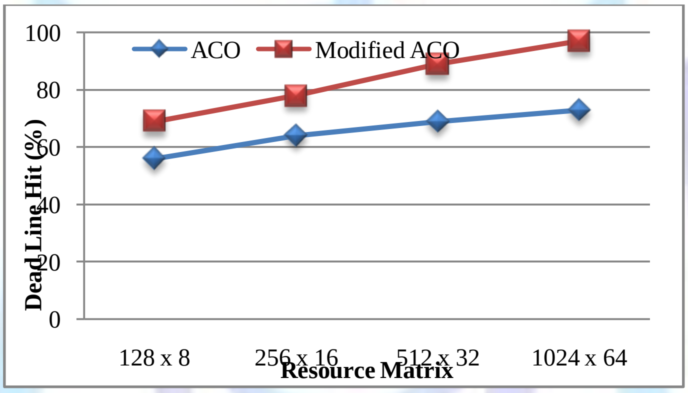

Figure 4: Comparison for Dead Line

The comparison for dead line is shows in figure 4. The proposed modified ACO has achieved deadline with high rate when compare with ACO.

5. CONCLUSION

In this work proposed modified ACO algorithm for improving the cloud workflow scheduling. Modified ACO is used to find the optimal resource allocation for batch tasks in the dynamic cloud system and minimize the makespan on the entire system. It introduces self-adapting criteria for the ACO control parameters. The proposed method is used to optimize the scheduling of workflow and balances the load on compute resources by distributing the tasks to available resources in cloud. The performance is evaluated using makespan, scheduling time, deadline hit parameters. An experimental result shows that the modified ACO algorithm gains both optimal solution and converges faster in large tasks and is proved.

REFERENCES

1. Zheng, Zhongni, Rui Wang, HaiZhong, and Xuejie Zhang. "An approach for cloud resource scheduling based on Parallel Genetic Algorithm." In Computer Research and Development (ICCRD), 2011 3rd International Conference on, vol. 2, pp. 444-447. IEEE, 2011.
2. Li, Wubin, Johan Tordsson, and Erik Elmroth. "Modeling for dynamic cloud scheduling via migration of virtual machines." In Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on, pp. 163-171. IEEE, 2011.
3. Daniel, David, and S. P. Lovesum. "A novel approach for scheduling service request in cloud with trust monitor." In Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on, pp. 509-513. IEEE, 2011.
4. Garg, Saurabh Kumar, Chee Shin Yeo, ArunAnandasivam, and RajkumarBuyya. "Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers." Journal of Parallel and Distributed Computing 71, no. 6 (2011): 732-749.

5. Ahn, Jeongseob, Changdae Kim, Jaeung Han, Young-Ri Choi, and Jaehyuk Huh. "Dynamic virtual machine scheduling in clouds for architectural shared resources." In Proceedings of the USENIX Workshop on Hot Topics in Cloud Computing (HotCloud). 2012.
6. Shrivastava, Vivek, and D. S. Bhilare. "Algorithms to Improve Resource Utilization and Request Acceptance Rate in IaaS Cloud Scheduling."International Journal of Advanced Networking and Applications 3, no. 05 (2012): 1367-1374.
7. Li, Wen-Juan, Qi-Fei Zhang, Ling-Di Ping, and Xue-Zeng Pan. "Cloud scheduling algorithm based on fuzzy clustering." Journal of China Institute of Communications 33, no. 3 (2012): 146-154.
8. Shi, Xuelin, and Ying Zhao. "Dynamic resource scheduling and workflow management in cloud computing." In Web Information Systems Engineering–WISE 2010 Workshops, pp. 440-448. Springer Berlin Heidelberg, 2011.
9. Li, Jiayin, MeikangQiu, Zhong Ming, Gang Quan, Xiao Qin, and ZonghuaGu. "Online optimization for scheduling preemptable tasks on IaaS cloud systems."Journal of Parallel and Distributed Computing 72, no. 5 (2012): 666-677.
10. Nishant, Kumar, Pratik Sharma, Vishal Krishna, Chhavi Gupta, KuwarPratap Singh, and Ravi Rastogi. "Load balancing of nodes in cloud using ant colony optimization." In Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on, pp. 3-8. IEEE, 2012.
11. Abrishami, Saeid, Mahmoud Naghibzadeh, and Dick HJ Epema. "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds." Future Generation Computer Systems 29, no. 1 (2013): 158-169.
12. Tsai, Chun-Wei, and Joel JPC Rodrigues. "Metaheuristic scheduling for cloud: A survey." Systems Journal, IEEE 8, no. 1 (2014): 279-291.
13. Mateos, Cristian, ElinaPacini, and Carlos GarcíaGarino. "An ACO-inspired algorithm for minimizing weighted flowtime in cloud-based parameter sweep experiments." Advances in Engineering Software 56 (2013): 38-50.
14. Chang, Ruay-Shiung, Chih-Yuan Lin, and Chun-Fu Lin. "An adaptive scoring job scheduling algorithm for grid computing." Information Sciences 207 (2012): 79-89.
15. Pacini, Elina, Cristian Mateos, and Carlos GarcíaGarino. "Distributed job scheduling based on Swarm Intelligence: A survey." Computers & Electrical Engineering 40, no. 1 (2014): 252-269.

Authors Biography

1. S.J.MOHANA

She has completed her Post Graduate with M.Phil from Bharathiar University, Coimbatore. Now, she is working as an Assistant Professor in Erode Arts and Science college, Erode. She presented 1 International Conference and 1 International workshop .Many national and international journals are also published. She is intersted in Grid computing, cloud computing, advanced networking. Now she is doing research papers in cloud computing.

2. Dr. M.SAROJA

She had completed her Ph.D., in Bharathiar university, Coimbatore. She is working as an Associate Professor in Erode Arts and Science College, Erode. She has published many national and international journals. She has presented many papers in international and national conferences. Her area of interest includes mobile computing, cloud computing and thin films.

3. Dr. M.VENKATACHALAM

He had completed his Ph.D., in Bharathiar university, Coimbatore. He is working as an Associate Professor & Head, in Erode Arts and Science College, Erode. He has published numerous papers in national and international journals.He has presented many papers in international and national conferences. His area of interest includes thin films, grid computing and cloud computing.