ISSN:0975-9646
S. Umarani et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2) , 2012,3388-3393

Efficient Multiple Ant Colony Algorithm for Job Scheduling In Grid Environment

S.Umarani[1], L.M.Nithya[1], A.Shanmugam[2]
[1]*SNS College of Technology, Coimbatore*
[2]*Bannari Amman Institute of Technology, Erode.*

Abstract:-Grid computing is now being used in many applications that are beyond distribution and sharing of resources. One primary issue associated with the efficient utilization of heterogeneous resources in a grid environment is task scheduling. Task Scheduling is an important issue of current implementation of grid computing. The demand for scheduling is to achieve high performance computing. If large numbers of tasks are computed on the geographically distributed resources, a reasonable scheduling algorithm must be adopted in order to get the minimum completion time. Typically, it is difficult to find an optimal resource allocation for specific job that minimizes the schedule length of jobs. So the scheduling problem is defined as NP-complete problem and it is not trivial. Heuristic algorithms are used to solve the task scheduling problem in the grid environment and may provide high performance or high throughput computing or both. In this paper, a new improved multiple ant colony optimization scheduling algorithm is proposed. The proposed scheduler allocates the best suitable resources to each task with minimal makespan and execution time. The experimental results are compared which shows that the algorithm produces better results when compared with the multiple ant colony algorithm.

Keywords: grid computing; task scheduling; multiple ant colony optimization; ant colony algorithm; improved multiple ant colony algorithm

1. INTRODUCTION:

Grid computing has emerged as an important new field, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and in some cases, high-performance orientation. Grid is a wide-scale, distributed computing infrastructure that promises to support resource sharing and coordinated problem solving in dynamic, multi institutional Virtual Organization [7]. Grid computing is applying the resources of many computers in a network for a single problem at the same time - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data.

Scheduling in a grid computing system is not as simple as scheduling on a many machine because of several factors. These factors include the fact that grid resources are sometimes used by paying customers who have interest in how their jobs are being scheduled. Also, grid computing systems usually operate in remote locations so scheduling tasks for the clusters may be occurring over a network. It is because of these reasons that looking at scheduling in grid computing is an interesting and important problem to examine [1].

There are two types of scheduling namely static scheduling and dynamic scheduling in grid computing system. For the static scheduling, jobs are assigned to suitable resources before their execution begin. However, for the dynamic scheduling, reevaluation is allowed of already taken assignment decisions during job execution [2].

Figure 1: Grid Computing Environment

It can trigger job migration or interruption based on dynamic information about the status of the system and the workload. This job scheduling is the mapping of jobs to specific physical resources, trying to minimize some cost function specified by the user. This is a NP-complete problem and different heuristics may be used to reach an optimal or near optimal solution. Effective computation and job scheduling is rapidly becoming one of the main challenges in grid computing and is seen as being vital for its success.

In this paper, we propose a new improved multiple ant colonies optimization approach for achieving optimal task scheduling in grid computing. This paper is organized as follows. Section II describes the use of ant and multiple ant colony algorithms in grid computing. In Section III, It covers problem formulation and methodology used in existing system. In Section IV, our new improved MACO approach is proposed for the scheduling problem. In Section V, computational experiments and comparison studies are reported. Some concluding remarks are made in Section VI.

2. Related Works on ACO and MACO in Grid Computing Environment:

One motivation of grid computing is to aggregate the power of widely distributed resources, and provide non-trivial services to users. To achieve this goal, an efficient grid scheduling system is an essential part of the grid. Rather than covering the whole grid scheduling area, survey provides a review of the subject mainly from the perspective of scheduling algorithms. The state of current research on scheduling algorithms for the new generation of computational environments will be reviewed in this section. Here, a collection of some commonly used heuristics algorithms literature has been selected and discussed in the following section.

Ant Colony Optimization (ACO) [3] has been used as an effective algorithm in solving the scheduling problem in grid computing. ACO is inspired by a colony of ants that work together to find the shortest path between their nest and food source. Every ant will deposit a chemical substance called pheromone on the ground after they move from the nest to food sources and vice versa. Therefore, they will choose the shortest or optimal path based on the pheromone value. The path with high pheromone value is shorter than the path with low pheromone value. This behavior is the basis for a cooperative communication.

Balanced job assignment based on ant algorithm for computing grids called BACO was proposed by [4]. The research aims to minimize the computation time of job executing in grid environment which focused on load balancing factors of each resource. By considering the resource status and the size of the given job, BACO algorithm chooses optimal resources to process the submitted jobs by applying the local and global pheromone update technique to balance the system load. Local pheromone update function updates the status of the selected resource after job has been assigned and the job scheduler depends on the newest information of the selected resource for the next job submission. Global pheromone update function updates the status of each resource for all jobs after the completion of the jobs. From the experimental result, BACO is capable of balancing the entire system load regardless of the size of the jobs.

It is based on the improved ACO that proposed by [5]. The pheromone update function in this research is performed by adding encouragement, punishment coefficient and load balancing factor. The initial pheromone value of each resource is based on its status where job is assigned to the resource with the maximum pheromone value. The strength of pheromone of each resource will be updated after completion of the job. If a resource completed a job successfully, more pheromone will be added by the encouragement coefficient in order to be selected for the next job execution. If a resource failed to complete a job, it will be punished by adding less pheromone value. The load of each resource is taken into account and the balancing factor is also applied to change the pheromone value of each resource.

An Enhanced Ant Colony Algorithm proposed in [6] is allocating an application to a host from a pool of available hosts and applications by selecting the best match. The proposed algorithm uses two types of ET (Execution Time) matrices and finds the list of available resources in the grid environment, form the ET matrix and start the scheduling. Based on the ET matrix the algorithm calculates the probabilistic makespan. The proposed scheduler allocates adopt the system environment freely at runtime. This resource optimally and adaptively in the scalable, dynamic and distribute controlled environment.

Multiple Ant Colony Optimization (MACO) approach [7] presented for load balancing in circuit-switched networks. MACO uses multiple ant colonies to search for alternatives to an optimal path. One of the impetuses of MACO is to optimize the performance of a congested network by routing calls via several alternatives paths to prevent possible congestion along an optimal path.

In MACO, each group of mobile agents corresponds to a colony of ants, and the routing table of each group corresponds to a pheromone table of each colony [8]. By adopting the MACO approach, it may be possible to reduce the likelihood that all mobile agents establish connections using only the optimal path [8]. The advantage of using MACO in circuit-switched routing is that it is more likely to establish connections through multiple paths to help balance the load but does not increase the routing overhead.

IMACO framework is proposed in [9]. In this framework there are two levels of interaction the first one is the colony level and the second one is the population level. The colony level interaction can be achieved through the pheromone depositing process within the same colony; the pheromone updating mechanism is responsible for the implementation of this kind of interaction. The population level interaction is achieved by evaluating the pheromones of different colonies using some evaluation function; the responsibility here is of the pheromone evaluating mechanism [10].

IMACO-AVG method [11] proposed exploration and exploitation process. Exploration and exploitation is controlled by the parameter q_0 whose value is in [0, 1]. It is usually used in ant's probabilistic decision as trade-off between exploitation and exploration.

Setting q_0 to zero means that the algorithm uses a pure exploration while pure exploitation is reached by setting q_0 to one. This technique enables the utilized ant colonies to work with different levels of exploration. Some will prefer high exploration of new areas of search space while other colonies will prefer high exploitation search history.

Information exchange in multiple ant colony algorithms [12] is proposed that parallelization where an information exchange between several colonies of ants is done every k generations for some fixed k. They show by using simulations how much the running time of the algorithm

decreases with an increasing interval between the information exchanges. But it is not discussed how this influences the quality of the solutions.
Through our literature survey on current scheduling algorithms are working in the grid computing scenario, we can find that heterogeneity, dynamism, computation and data separation are the primary challenges concerned by current research on this topic.

3. MULTIPLE ANT COLONY ALGORITHM AND PROBLEM DESCRIPTION:

Recently, ant colony optimization (ACO) has been suggested to solve the task scheduling problem. But ACO approaches using a single colony system may suffer from specific local optima because of its tendency to use the positive feedback mechanism of pheromone, multiple ant colonies optimization (MACO) is employed to avoid this by using several ant colonies to solve combinatorial optimization problems cooperatively. To improve the performance of ACO approaches, MACO considers both positive and negative feedbacks in searching solutions, sharing the search information, and exploring a large area of the search space with mutual cooperation of ant colonies. So MACO approaches have been explored for several optimization problems [14].
MACO seems to be appropriate approach to improve the performance of ACO algorithm. This algorithm offers good opportunity to explore a large area of the search space and find optimal solution. All colonies construct their solutions in parallel [13] and interaction mechanisms are designed for sharing experiences among colonies. Assume that M ant colonies would be used to tackle the scheduling problem, and each colony contains N ants for the search procedure.

The Framework of the MACO Approach:

- First the initialization of the algorithm is presented,
- Second Local and global phenomenon updating is performed,
- Third construct the solution for determine the makespan time and then terminal test of the algorithm is take place.

Initially several colonies of ant system are created, and then they perform iterating and updating their pheromone arrays respectively until one ant colony system reaches its local optimum solution. Every ant colony system owns its pheromone array and parameters and records its local optimum solution. Furthermore, once an ant colony system arrives at its local optimum solution, it updates its local optimum solution and sends this solution to global best-found center. In general, the approach can be briefly sketched as follows.

Initialization of Algorithm:

Assume that M ant colonies would be used to tackle the scheduling problem, and each colony contains N ants for the search procedure. We denote by ant (m, n) the nth ant in the mth colony. At the beginning, ants are distributed on computing nodes randomly, and the pheromone value τ_j^m of the m^{th} colony on node Cj is initialized as a small value.

Interaction of Multiple Ant Colony System:

Pheromone is used as the interaction mechanism not only between the ants of the same colony but also among ant colonies. As traditional ACO approaches, each colony has its own pheromone to interact between the ants of the same colony. Furthermore, pheromone information is also used for the interaction of ant colonies. The colony level interaction is achieved by evaluating the pheromones of different colonies. More specifically, the evaluated pheromone τj on node Cj in terms of pheromone values of all colonies is defined as follows,

$$\tau_j = \Sigma_{m\varepsilon [1, M]} \tau_j^m \; / \; M \qquad (1)$$

Where M-no of ant colonies in system, τ_j – is computed in terms of pheromone values of all colonies
The pheromone evaluation mechanism averages the pheromone values of all colonies, which represents information of all colonies. Based on the average of the available experiences of ants of all colonies, an ant will decide how to choose an edge.

Pheromone Updating:

As ant (m, n) completes its tour, local pheromone updating is applied on the visited nodes. More specifically, if ant (m, n) assigns task Ti to node Cj, the local pheromone update is given by:

$$\tau_j^m = (1-\rho)\, \tau_j^m + \rho\, \Delta\, \tau_j^{mn} \qquad (2)$$

Where ρ- Evaporation rate

In global pheromone updating phrase, after all ants of all colonies construct their solutions, the ant achieving the best-so-far solution in its colony will deposit an amount of pheromone on the edges of its path according to the following rule

$$\tau_j^m = (1-\lambda)\, \tau_j^m + \lambda\, \Delta\, \tau_j^{mn} \qquad (3)$$

Where λ- Evaporation

Thus, the ant finding the solution with the minimum makespan can lay a larger intensity of the pheromone on its tour.

Solution construction:

In this phase, ant (m, n) moves through computing nodes and assigns task Ti to node Cj probabilistically in terms of pheromone and heuristic information until all tasks have been allocated. The probability $P^{mn}{}_{ij}$ for ant (m,n) to assign the task to node is defined as;

$$P^{mn}{}_{ij} = \tau_j\, \eta^{mn}{}_{ij} \,/ \sum \tau_j\, \eta^{mn}{}_{ij} \qquad (4)$$

Where $\eta^{mn}{}_{ij}$ the heuristic information value for evaluating the assignment of task Ti to node Cj for ant (m, n). τ_j - is computed in terms of pheromone values of all colonies.
This approach terminates when the global pheromone updating is not improved in successive predefined no of solutions.

4. Proposed Enhanced Multiple Ant Colony Algorithms for Job Scheduling in Grid Computing:

One of the main challenges is to find the best or optimal resources to process a particular job in term of minimizing the job computational time. The scheduling problem aims to minimize the total execution time of tasks as well as to achieve a well-balanced load across all nodes in grid system. Therefore in this paper one of the main objectives is to minimize the makaspan, which represents the latest completion time among all the tasks as well as utilizing the resources by efficient way.

The proposed enhanced multiple ant colony algorithm is used to solve the large complex problems. It requires grid scheduling to achieve high performance. Before starting the grid scheduling, the expected execution time for each task on each machine must be estimated and represented by an ET matrix [15]. The ET matrix will have N x M entries, where N is the number of independent jobs to be scheduled and M is the number of resources which is currently available. Each job's workload is measured by millions of instructions and the capacity of each resource is measured by MIPS. Each row of ET matrix consists of the estimated execution time for a job on each resource and every column of the ET matrix is the estimated execution time for a particular resource of list of all jobs in the job pool. The completion time of ith job on the jth machine is

$$CT_{ij}^{m} = Ready_i + EET_{ij}^{m} \quad (5)$$

Where EETij is the expected execution time of task Ti on node Cj when task Ti is assigned to node Cj, otherwise EETij is 0. Ready (i) is the machine availability time, i.e. the time at which machine mj completes any previously assigned tasks.

The proposed efficient MACO has changed the basic pheromone updating of multiple ant colony algorithm .In multiple any colonies; all the ants maintain a separate list in each colony. 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 local updating rule

$$\tau_j^{m} = 1 / EET_{ij}^{m} \quad (6)$$

This local pheromone updating rule is implemented after each ant completes its tour in each ant colony. Based on local pheromone updating we calculate the global pheromone value. Moreover, global pheromone updating is done by the best ant of each colony after all ants of all colonies complete their tours,

$$\tau_j^{m} = (1-\lambda)\, \tau_j^{m} + \lambda\, \Delta\, \tau_j^{mn} \quad (7)$$

The colony level interaction is achieved by evaluating the pheromones of different colonies. More specifically, the evaluated pheromone τj on node Cj in terms of pheromone values of all colonies is defined as follows,

$$\tau_j = \Sigma_{m\varepsilon\,[1,\,M]}\, \tau_j^{m} \;/\; M \quad (8)$$

Where M-no of ant colonies in system, τ_j - is computed in terms of pheromone values of all colonies

The probability $P^{mn}_{ij\ for}$ ant (m,n) to assign the task to node is defined as;

$$P^{mn}_{ij} = \tau_j\, Ꝺ^{mn}_{ij} \;/\; \sum \tau_j\, Ꝺ^{mn}_{ij} \quad (9)$$

Where $Ꝺ^{mn}_{ij}$ the heuristic information value for evaluating the assignment of task *Ti* to node *Cj* for ant (*m*, *n*). τ_j - is computed in terms of pheromone values of all colonies.

The pheromone evaluation mechanism averages the pheromone values of all colonies, which represents information of all colonies. Based on the average of the available experiences of ants of all colonies, an ant will decide how to choose an edge.

The inclusion of expected execution time matrix in local pheromone updating is improving the performance of multiple ant colony algorithms. This improvement is in terms of the decrease in makespan time.

5. Experimental Results:

The proposed multiple ant colonies construct their solutions in parallel and interaction mechanisms are designed for sharing experiences among colonies. The performance evaluation of our proposed EMACO algorithm and the comparison study with other algorithms for task scheduling have been done on GridSim, which is developed to support simulation of heterogeneous grid resources and application models.

Comparison and discussion:

From comparison purposes, ACS was run using 10, 20, 30, 40, 50,60,70,80 and 80 ants such that the number of ants used by ACS is equal to the sum of ants of all colonies in MACO and EMACO. Here we use 4 gridlets and 3 resources and the results of the experiments are shown in table 1 and 2.

Table 1: probabilistic makespan of MACO

Probability	Makespan
[0][0]	818
[0][1]	652
[0][2]	563
[0][3]	506
[1][0]	491
[1][1]	404
[1][2]	348
[1][3]	343
[2][0]	309
[2][1]	280
[2][2]	176
[2][3]	163

The inclusion of EETij execution time of the ith job by the jth machine (predicted) in the calculation of probability, that the jth machine will be free, has shown a positive result in performance improvement. This improvement is in terms of the decrease in makespan time.

Table 2: Probabilistic makespan of EMACO

Probability	Makespan
[0][0]	799
[0][1]	550
[0][2]	490
[0][3]	476
[1][0]	359
[1][1]	350
[1][2]	332
[1][3]	312
[2][0]	275
[2][1]	230
[2][2]	141
[2][3]	132

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. The search is performed on each problem processor and continues until there is no further improvement in the global pheromone of the solution. These results are shown in figure 2.

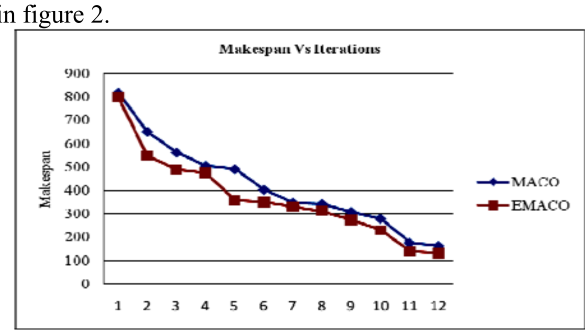

Figure 2: MACO and EMACO performance comparison based on probabilistic makespan

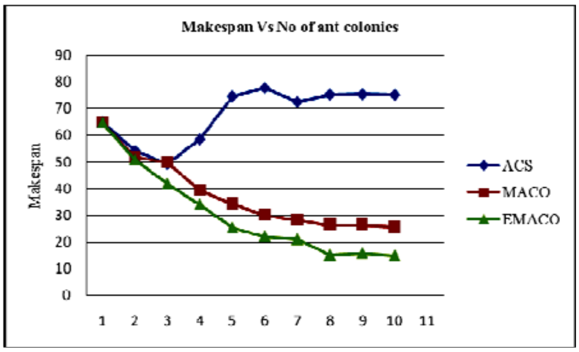

Figure 3: ACO, MACO and EMACO performance comparison

From the experimental results of figure 3, we can see that all the optimal makespan obtained by EMACO is better than the MACO and ACO. Here we use ten ant colonies and the makespan of ACO decreased roughly as the no of ants increased. Due to the influence of stagnation situation, ACO cannot benefit from the increase of the number of ants. In contrast, the solution quality of MACO was improved through the cooperation of multiple colonies. But EMACO can be improved using some form of local searching process.

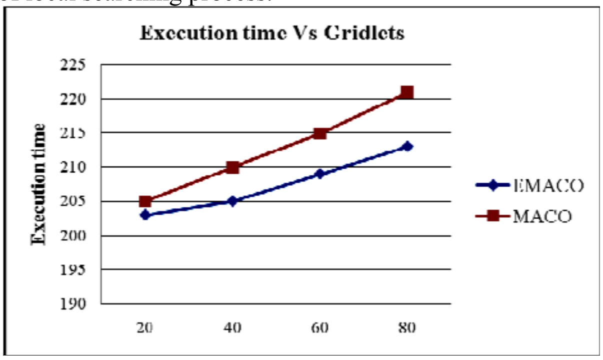

Figure 4: MACO and EMACO performance comparison based on simulation time

Figure 4 shows that the simulation time for enhanced multiple ant colony algorithm and multiple ant colony algorithm. Enhanced multiple ant colony algorithm produces minimum simulation time when compared with traditional process. This is because the groups of all ant colonies are constructing their local pheromone with the help of expected execution time matrix in parallelization. Therefore the proposed multiple ant colonies algorithm is a best suited method for tracking problem with large data sets. From the results it is clearly evident that the proposed efficient multiple ant colonies algorithm offers better optimization a very fast rate and produced better makespan. In other words, these results prove the effectiveness of the EMACO algorithm.

6. CONCLUSION:

Grid computing aims to assign tasks to computing nodes and minimize the execution time of tasks as well as workload across all nodes. Despite of the intractability, the scheduling problem is of particular concern to both users and grid systems. The proposed EMACO approach achieves optimal schedule than the traditional multiple colony algorithm. In EMACO, experiences are shared from the cooperation of multiple ant colonies. At the same time, the positive and negative feedback are applied to avoid stagnation situation encountered in searching. Here the modified pheromone updating is allocating the resources and obtain the shortest path optimally and adaptively in scalable, dynamic and distributed environment. This enhancement process has been achieved optimal scheduling by completing the tasks with minimum execution time as well as utilizing the resource in an efficient way.

REFERENCES:

[1] Foster and Kesselman.C, “The Grid:Blueprint for a Future Computing Infrastructure” Morgan Kaufman Publishers, USA, 1999.

[2] M. Chtepen, “Dynamic scheduling in grids system,” Sixth Firw PhD Symposium, Faculty of Engineering, Ghent University, pp. 110, 2005.

[3] Stefka Fidanova and Mariya Durchova “Ant Algorithm for Grid Scheduling Problem”, Large Scale Computing, Lecture Notes in Computer Science No. 3743, Springer, 2006, pp.405-412.

[4] Chang.R, Chang.J, and Lin.P "Balanced Job Assignment Based on Ant Algorithm for Grid Computing," presented at Proceedings of the 2nd IEEE Asia-Pacific Service Computing Conference, 2007, pp. 291-295.

[5] Yan.H, Shen.X, Li.X, and Wu.M,(2005) "An improved ant algorithm for job scheduling in grid computing," presented at Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, vol. 5, 2005, pp. 2957-2961.

[6] Kousalya.K and Balasubramanie.P ,“An Enhanced Ant Colony Algorithm for Grid Scheduling Problem”, IJCSNS International Journal of Computer Science and Network Security, Vol.8, No.4,2008

[7] Kwang Mong Sim, and Weng Hong Sun, " Multiple Ant Colony Optimization for Load Balancing" journal of computer applications,2008,pp-572-574.

[8] Kwang Mong Sim and Weng Hong Sun," Ant Colony Optimization for Routing and Load - Balancing: Survey and New Directions", Ieee Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 33, No. 5, September 2003.

[9] H. Kawamura, M. Yamamoto, K. Suzuki, and A. Ohuchi, "Multiple ant colonies algorithm based on colony level interactions", IEICE Trans. Fundamentals, Vol. E83-A, No.2, 2000, pp. 371-379.

[10] J. Jong, and M. Wiering, "Multiple ant colony system for the bus-stop allocation problem", Proc of the Thirteenth Belgium-Netherlands Conference on Artificial Intelligence BNAIC’01, Amsterdam, Netherlands, 2001, pp. 141–148.

[11] Alaa Aljanaby , Ku Ruhana Ku-Mahamud , Norita Md. Norwawi,” Interacted Multiple Ant Colonies Optimization Framework: an Experimental Study of the Evaluation and the Exploration Techniques to Control the Search Stagnation,” International Journal of Advancements in Computing Technology Volume 2, Number 1, March 2010.

[12] Martin Middendorf, Frank Reischle, Hartmut Schmeck ” Information Exchange in Multi Colony Ant”, Institute four Angewandte Informatikund Formale Beschreibungsverfahren, Universityat Karlsruhe, D-76128 Karlsruhe, Germany.

[13] Kruger.F, Middendorf.M, and Merkle.D,” Studies on a Parallel Ant System for the BSP Model” Unpub. Manuscript, 1998.

[14] Liang Bai, Yan-Li Hu, Song-Yang Lao, Wei-Ming Zhang,” Task Scheduling with Load Balancing using Multiple Ant Colonies Optimization in Grid Computing” International Conference on Natural Computation (ICNC), 2010.

[15] K. Kousalya and P. Balasubramanie ”To Improve Ant Algorithm’s Grid Scheduling Using Local Search”, International Journal Of Computational Cognition Vol. 7, No. 4, December 2009.