Hybrid Ant Colony Optimization for Grid Computing

Husna Jamal Abdul Nasir[a], Ku Ruhana Ku-Mahamud[b]

[a]*College of Arts and Sciences*
Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia
E-mail: oxalic2131@gmail.com

[b]*College of Arts and Sciences*
Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia
E-mail: ruhana@uum.edu.my

ABSTRACT

A hybrid ant colony optimization technique to solve the stagnation problem in grid computing is proposed in this paper. The proposed algorithm combines the techniques from Ant Colony System and Max – Min Ant System and focused on local pheromone trail update and trail limit. The agent concept is also integrated in this proposed technique for the purpose of updating the grid resource table. This facilitates the hybrid ant colony optimization technique in solving the stagnation problem in two ways within one cycle, thus minimize the total computational time of the jobs.

Keywords:
Grid Computing, Stagnation, Hybrid Ant Colony Optimization.

1.0 INTRODUCTION

Grid computing which is based on the cluster computing concept is a new trend of computational environment. There are many challenges to schedule jobs in the grid computing environment to avoid stagnation which occurs when all jobs used the same resources at the same time. Dynamic and unpredictable behavior in grid computing such as the computational performance changes from time to time, the networks connections become unreliable, nodes may join or leave the system at any time and node may become unavailable without any notification further contribute to the difficulty in solving the problem. The words node and resource have the same meaning and will be used interchangeably in this paper.

In any grid environment, there exists more than one resource. The challenge is to find the best or optimal resource to process a particular job in term of minimizing the computational time. Computational time is a measure of how long that resource takes to complete the job. Optimal resources always have a good processor, high CPU speed and large memory space. When the optimal resource is assigned to process the job, the computational time is expected to be low, thus stagnation in grid environment can be avoided. Stagnation will occur if the computational time is high. Therefore, grid computing system needs a good scheduling technique in allocating jobs to resources. Stagnation in grid computing system may occur when all jobs are assigned to or required the same resource. The resource will end up having a high workload. An effective job scheduling algorithm which is job and resource independent is needed to make sure that not all jobs are assigned to the same resource.

In static scheduling (Chtepen, 2005), jobs are assigned to the appropriate resources before their execution begin. Once started, they keep running on the same resources without interruption. However, in dynamic scheduling, reevaluation is allowed of already taken assignment decisions during job execution. It can trigger job migration or interruption based on dynamic information about the status of the system and the workload.

Ant colony optimization (ACO) has been applied in solving many problems in scheduling such as Job Shop Problem (JSP), Open Shop Problem (OSP), Permutation Flow Shop Problem (PFSP), Single Machine Total Tardiness Problem (SMTTP), Single Machine Total Weighted Tardiness Problem (SMTWTP), Resource Constraints Project Scheduling Problem (RCPSP), Group Shop Problem (GSP) and Single Machine Total Tardiness Problem with Sequence Dependent Setup Times (SMTTPSDST) (Dorigo & Stützle, 2004). A recent approach of ACO researches in the use of ACO for scheduling job in grid computing (Lorpunmanee, Sap, Abdullah & Chompoo-inwai, 2007). ACO algorithm has been used in grid computing because it is easily adapted to solve both static and dynamic combinatorial optimization problems.

ACO which is based on ant biological behavior takes inspiration from the foraging behavior of some ant species. The presence of these and other unique characteristics have made ant societies an attractive and inspiring model for building new algorithms. Workers of ant colony specialize in particular tasks and perform them more than other workers do. For example, the soldiers aim for protection, the scouts specialize in searching for food sources, and the queen's task is producing new ants.

This paper presents a global inspired job scheduling mechanism to solve the stagnation problem in grid computing system. Section 2 describes the use of ant colony optimization algorithms in grid computing while the heuristic analysis of job scheduling in grid computing is explained in Section 3. The proposed algorithm is presented in Section 4 and concluding remarks are highlighted in Section 5.

2.0 ACO ALGORITHMS IN GRID ENVIRONMENT

Generally, when a job is submitted to a grid computing environment, it needs to be assigned to the available resources. During the scheduling process, the optimal resource is more likely to be selected for the submitted jobs which eventually lead to stagnation in grid environment. ACO algorithm can solve this problem by allocating suitable resource to each submitted job to avoid stagnation. This is done by considering the values of pheromone, a chemical substance used for indirect communication between the ants, for resource selection.

Ant colony system (ACS) (Gambardella & Dorigo, 1996), a variant of ACO, had been applied to solve the stagnation problem in grid computing (Chang, Chang & Lin, 2007), (Lorpunmanee et al., 2007). It was first introduced to improve the performance of Ant System (Colorni, Dorigo & Maniezzo, 1991), (Dorigo, Maniezzo & Colorni, 1991), (Dorigo, 1992). ACS differs in three main aspects from ant system. First, ACS uses a more aggressive action choice rule than AS. Second, the pheromone is added only to nodes belonging to the global-best solution. Third, each time an ant uses a node to process the job, it removes some pheromone from the node. The three main phases of the ACS algorithm constitute the ants' solution construction, global pheromone trail update, and local pheromone trail update. In ACS the pheromone updating consists of two stages: local update and global update. While ants build their solutions, at the same time they locally update the pheromone level of the visited nodes by applying the local update.

The aim of the local update rule is to make better use of the pheromone information by dynamically changing the desirability of node. Using this rule, ants will search in a wide neighborhood of the best previous solution. The pheromone level on the nodes is highly related to the value of evaporation. The pheromone level will be reduced and this will reduce the chance that other ants will select the same option. When all ants have completed their solutions, the pheromone level is updated by applying the global updating rule only on the paths that belong to the best solution.

The study of dynamic job scheduling in grid environment that applied ant colony optimization was proposed by (Lorpunmanee et al., 2007) which aimed to minimize the total job tardiness time. In that study, the process to update the pheromone trail strength on each resource followed the local update and global update rules as in ACS. Comparison of results with First Come First Serve (FCFS), Minimal Tardiness Earliest Due Date (MTEDD) and Minimal Tardiness Earliest Release Date (MTERD) shows that ACO algorithm performed the best.

A bio-inspired adaptive job scheduling mechanism in grid computing was proposed by (Li, 2006) with the goal to minimize the execution time of the computational jobs by effectively taking advantage of the large amount of distributed resources. The research designs various software ant agents with simple functionalities. Comparison of the bio-inspired adaptive scheduling was also performed with the random mechanism and heuristic mechanism. The results showed that a bio-inspired adaptive job scheduling has good adaptability and robustness in a dynamic computational grid.

Balanced job assignment based on ant algorithm for computing grids called BACO was proposed by (Chang et al., 2007). The objective of the research is to decrease the computation time of jobs executing in Taiwan UniGrid environment which also considers the loading of each resource. BACO algorithm chooses optimal resources to perform jobs according to resource status and the size of the given job. The local and global pheromone update functions used were able 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. It offers the job scheduler the newest information of all resource for the next job submission. Experimental result shows that the BACO is capable of balancing the entire system load regardless of the size of the jobs.

An improved ant algorithm for job scheduling in grid computing which is based on the basic idea of ACO was proposed by (Yan, Shen, Li & Wu, 2005). A change in the pheromone update function 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 and the pheromone of each resource will be updated after completion of the job. The encouragement and punishment and local balancing factor coefficient are defined by users and are used to update pheromone values of resources. If a resource completed a job successfully, it will be added more pheromone 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.

Simple grid simulation architecture for resource management and task scheduling was proposed in (Xu, Hou & Sun, 2003). The scalability of ant algorithm was also validated in the study. The ant algorithm for grid task scheduling is integrated into the simulation architecture and good results were obtained in terms of response time, resource average utilization and task fulfill proportion.

To date, ACS is the most popular variant of ACO that has been successful in solving stagnation in grid computing. However, more work is needed to enhance the performance of the algorithm in this application domain.

3.0 HEURISTIC ANALYSIS OF JOB SCHEDULING IN GRID COMPUTING

The objective of every job scheduling algorithms on the grid computing is to minimize the computation time of the computational jobs by effectively taking advantage of the optimal used of the resources. The scheduling process in the work by (Li, 2006) is designed based on various agents with simple functionalities. In the study, all agents do not communicate directly with each other except communication via the pheromone values stored in a global grid resource table. The grid resource table is the only global data structure in the schedule algorithm. This table keeps track of the available nodes and the pheromone values associated with them. The values of pheromone will decreases by the evaporating process. An ant agent can deposit their pheromone in the grid resource table.

In this study, the agent concept introduced in (Li, 2006) has been adopted as shown in Figure 2. Specialized agents are categories as scout, tester, worker, cleaner and queen and the scheduling process is described in the following 5 stages.

1. The queen spawns cleaners, scouts, workers and testers.
2. A scout goes to information service provider to define nodes providing computational services. A scout finds available nodes and adds them to the grid resource table with initial pheromone value, θ.
3. When job is submitted to the grid computing, a worker will schedule the job to an available node. A node with high pheromone value will be selected.
4. Tester carries out the sample program on every node. When the program complete, a tester will update the pheromone value.
5. The evaporating process will decrease the amount of pheromone in each node. When the pheromone value is lower than threshold value, the cleaner will remove it from the grid resource table. This means that node has been unavailable for a long time.

The scheduling process introduced by (Li, 2006) did not take into consideration of the machine utilization status which could aggravate the stagnation situation.

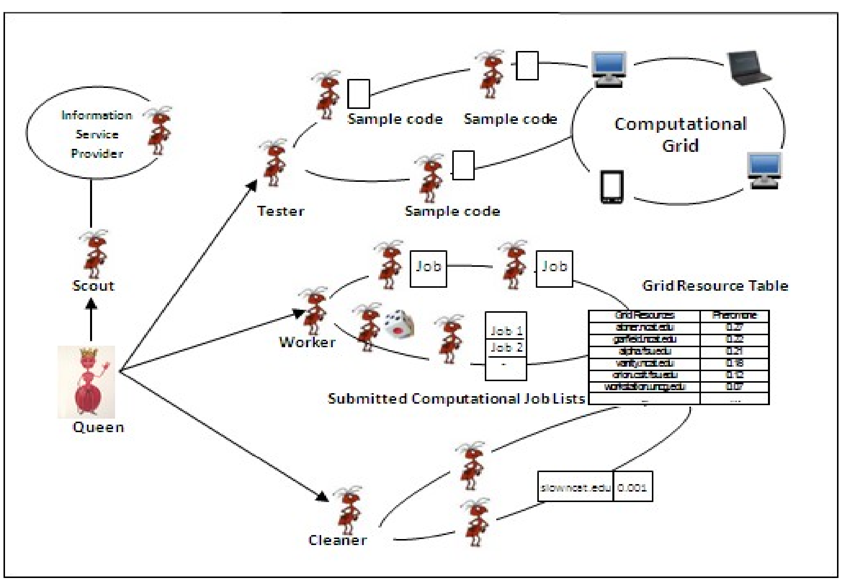

Figure 2: Behaviors of Ant Agents

4.0 PROPOSED HYBRID TECHNIQUE

Search stagnation is the situation where all ants used the same resources to complete the jobs. A tradeoff between the influence of the heuristic information and the pheromone trails has to be determined.

In this proposed technique, ACS and Max-Min Ant System (MMAS) (Stützle & Hoos, 2000) features are applied to solve the stagnation problem in grid computing environment. The proposed technique has used the agent concept in (Li, 2006) for updating the grid resource table. The local pheromone update in ACS is introduced with the global pheromone update in MMAS in updating the pheromone in grid resource table.

MMAS introduced explicit maximum and minimum trail strength on each node. The imposed trail limits have the effect of indirectly limiting the probability p_{ij} of selecting a next node to an interval $[p_{min},p_{max}]$ with $0<p_{min}\leq p_{ij}\leq p_{max} \leq 1$. If ant has single possible choice for the next node, then $p_{min} = p_{max} = 1$. The trail strengths are initialized to p_{max} for all nodes. The evaporation will reduce the pheromone strength by factor ρ in each iteration. The trails strength in bad node will decrease slowly but the good arc will maintain their strength. So, this arc is selected more often by other ants.

The proposed algorithm consists of four main features to be implemented. The first feature is the pheromone initialization value where the initial value is given by

$$trail\,\tau_0 = \frac{1}{m.\left(\sum_{i=1}^{m} T_{i\,Actual} + \sum_{i=1}^{m}\left(\sum_{j=1}^{n} T_{i,j}\right)_{Expected}\right)} \qquad (1)$$

where $\sum_{i=1}^{m} T_{i\,Actual} + \sum_{i=1}^{m}\left(\sum_{j=1}^{n} T_{i,j}\right)_{Expected}$ is the total computational time.

$\sum_{i=1}^{m} T_{i\,Actual}$ is the actual computational time, which is already complete on machine *i* and the expected computational time of jobs, which is scheduled on machine *i* is given by

$$\sum_{i=1}^{m}\left(\sum_{j=1}^{n} T_{i,j}\right)_{Expected}$$

The second feature to be considered is the state transition rule of the machine. A set of ant is created which started with unscheduled job in machine. They move randomly until a feasible solution is constructed. From machine *i* to move to machine *m*, the ant will use the following formula:

$$m = \begin{cases} \arg\max_{u\in U} [\tau(i,u)].[\eta(i,u)]^{\beta} & \\ S & if\ q\leq q_0 \end{cases} \qquad (2)$$

where *q* is a random number and q_0 is a parameter $(0\leq q_0\leq 1)$ which determines the relative importance of exploitation versus exploration. $\tau(i,u)$ is the pheromone trail with the job j to be scheduled at machine *i* to machine *u*, *U* is the machine with no schedule yet. $\eta(i,u)$ is the heuristic desirability of assignment job *j* on machine *i* directly moves to *u*. *S* is a random variable selected according to the probability distribution.

The local and global pheromone trail updates are other features to consider in this proposed algorithm. The local pheromone trail update is applied immediately after having crossed a node during the tour construction. The pheromone value is given by

$$\tau_{iu} = (1-\xi).\tau_{iu} + \xi.\tau_0 \qquad (3)$$

where ξ,0< ξ<1 and τ_0 are two parameters. The value of τ_0 is set to be the same as the initial value for the pheromone trails. A good value for ξ was found to be 0.1, while a good value for τ_0 was found to be $1/nC^{nn}$, where n is the number of resources and C^{nn} is the length of a nearest-neighbor resource. The effect of the local pheromone update is to make an already chosen resource less desirable for a following ant. So, the exploration of not yet visited resource is increased.

The global pheromone update is similar to the method introduced in MMAS. In this case, after all ants have constructed a solution, the pheromone trails are updated according to the following formula:

$$\tau_{iu}(t+1) = (1-\rho).\tau_{iu}(t) + \Delta\tau_{iu}^{best} \qquad (4)$$

where $\Delta\,\tau_{iu}^{best} = 1/L^{best}$. The ant which is allowed to add pheromone may be the *iteration-best solution* or *global best solution.* If a specific resource is often used in the best solution, it will receive a larger amount of pheromone and stagnation will occur. So, lower and upper limits on the possible pheromone strengths on any resource are imposed to avoid stagnation. The imposed trails limits have the effects of limiting the probability ρ_{iu} of selecting resource *u* when ants is in node *i* to an interval $[p_{min},p_{max}]$, with $0<p_{min}\leq p_{ij}\leq p_{max}\leq 1$. With this minimum trail limit, the resource is less desire to be selected by the jobs since it will select the resource that has the upper trail limit.

This proposed hybrid ant colony optimization approach can easily solve the stagnation problem in grid computing as it will execute the pheromone trail update in two ways in one cycle. The limits placed on the pheromone values can avoid the optimal resource being overload. In this situation, stagnation problem that occurs in grid computing can be prevented.

CONCLUSION

The proposed hybrid approach to solve stagnation problems in grid computing integrates local pheromone trail update with trail limits. The local pheromone trail update will reduce the amount of pheromone in visited node, so the node they have visited is less desirable for other ants while the trail limit, which is the allowed range of the pheromone strength, is limited to maximum and minimum trail strength. This is a strategy to control the amount of pheromone updated on each node. The proposed technique is simple to be implemented in solving the stagnation problem in grid computing environment since the pheromone update process will be executed in two ways in one cycle. It can also reduce the time to manage resources in grid computing.

REFERENCES

[1] Chang, R., Chang, J., & Lin, P. (2007). Balanced Job Assignment Based on Ant Algorithm for Grid Computing. *IEEE Asia-Pacific Services Computing Conference,* 291-295.

[2] Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed optimization by ant colonies. *Proceedings of European Conference on Artifical Life*, Paris, France, Amsterdam: Elsevier Publishing, 134-142.

[3] Chtepen, M. (2005). *Dynamic scheduling in grids system*. Sixth FirW PhD Symposium, Faculty of Engineering, Ghent University, 110

[4] Dorigo, M. (1992). *Optimization, learning and natural algorithms.* PhD thesis, Politecnico di Milano, Italy.

[5] Dorigo, M., Maniezzo, V., & Colorni, A. (1991). The ant system: an autocatalytic optimizing process. Technical Report TR91-016, Politecnico di Milano.

[6] Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization.* Cambridge, Massachusetts, London, England: MIT Press

[7] Fidanova, S., & Durchova, M. (2006). Ant Algorithm for Grid Scheduling Problem, *Large Scale Computing, Lecture Notes in Computer Science, 3743,* 405- 412.

[8] Gambardella, L. & Dorigo, M. (1996). Solving symmetric and asymmetric TSPs by ant colonies. *Proceedings if the IEEE Conference on Evolutionary Computation*, Nagoya, Japan, 622-627.

[9] Li, Y. (2006). A bio-inspired adaptive Job Scheduling Mechanism on a Computational Grid, *IIJCSNS International Journal of Computer Science and Network Security, 6(3),* 1-7.

[10] Lorpunmanee, S., Sap, M. N., Abdullah, A. H., & Chompoo-inwai, C. (2007). An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment. *International Journal of Computer and Information Science and Engineering, 1(4),* 207-214.

[11] Stützle, T., & Hoos, H. H. (2000). MAX-MIN ant system. *Future Generation Computer Systems, 16*, 889-914.

[12] Xu, Z., Hou, X., & Sun, J. (2003). Ant Algorithm-Based Task Scheduling In Grid Computing. *Electrical and Computer Engineering, IEEE CCECE, Canadian Conference,* 1107- 1110.

[13] Yan, H., Shen, X., Li, X., & Wu, M. (2005). An Improved Ant Algorithm for Job Scheduling in Grid Computing. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 5,* 2957-2961.