GRAAA: Grid Resource Allocation Based on Ant Algorithm

Manpreet Singh
Department of Computer Engineering, M. M. Engineering College, M. M. University, Mullana, Haryana, India
Email: manpreet_nishu@yahoo.com

***Abstract*— Selecting the appropriate resources for the particular task is one of major challenging work in the computational grid. The major objective of resource allocation in grid is effective scheduling of tasks and in turn the reduction in execution time. Hence the resource allocation must consider some specific characteristics of the resources, tasks and then decide the metrics to be used accordingly. Ant algorithm, which is one of the heuristic algorithm suits well for the allocation and scheduling in grid environment. In this paper, a Grid Resource Allocation based on Ant Algorithm (GRAAA) is proposed. The simulation result shows that the proposed algorithm is capable of producing high quality allocation of grid resources to tasks.**

***Index Terms*— Resource Allocation, Task Scheduling, Ant System, Grid**

I. Introduction

Resource allocation and task scheduling are fundamental issues in achieving high performance in grid computing systems. However, it is a big challenge for efficient allocation and scheduling algorithm design and implementation. Unlike scheduling problems in conventional distributed systems, this problem is much more complex as new features of grid systems such as its dynamic nature and the high degree of heterogeneity of jobs and resources must be tackled [1]. By harmonizing and distributing the grid resources efficiently, an advanced resource allocation strategy can reduce total run time and total expense greatly and bring an optimal performance [2] [3].

Ant algorithm is a random search algorithm, like other evolutionary Algorithms [4]. The algorithm is a new model-based bionic approach with different transition and pheromone updating rules. It is inspired in food self-enforcing foraging behaviors exhibited by ant societies. It is algorithm for solving the NP-hard combinatorial optimization problems, such as TSP (Traveling Salesman Problem) [3]. Then it was used in JSP (Job-shop Scheduling Problem) [5][6], QAP (Quadratic Assignment Problem), and so on [7].

The motivation of this paper is to develop a grid resource allocation algorithm that can perform efficiently and effectively in terms of minimizing total execution time and cost. Not only does it improve the overall performance of the system but it also adapts to the dynamic grid system. First of all, this paper proposes a Resource Oriented Ant Algorithm (ROAA) to find the optimal allocation of each resource within the dynamic grid system. Secondly, the simulation of proposed algorithm is presented using Gridsim.

II. Related Work

Recently, many researchers have studied several works on allocation and scheduling in grid environment. Some of the popular heuristic algorithms, which have been developed, are Min-Min [8], the Fast Greedy [8], Tabu Search [8] and an Ant System [9]. Max-Min Ant System (MMAS) [10] limit the pheromone range to be greater than or equal to the low bound value (Min) and smaller than or equal to the upper bound value (Max) to avoid ants to converge too soon in some ranges. [11] use multiple kinds of ant to find multiple optimal paths for network routing. The idea can be applied to find multiple available resources to balance resources utilization in job scheduling. In [12], the scalability of ant algorithm is validated, a simple Grid simulation architecture and design of ant algorithm suitable for Grid task scheduling is proposed.

III. Grid Resource Allocation based on Ant Algorithm (GRAAA)

The GRAAA is a resource allocation framework, which comprises user, resource broker, resources and Grid Information Services (GIS). It adopts ant colony as major allocation strategy as shown in Fig.1.

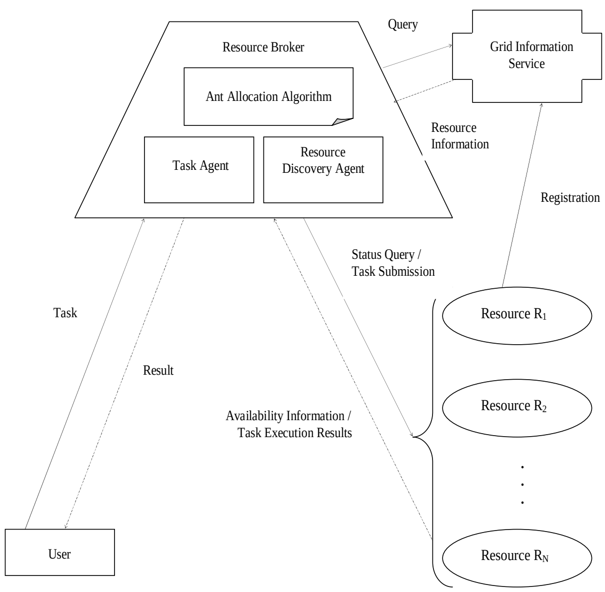

Figure 1: System Model

doi:10.4304/jait.1.3.133-135

The interaction among various entities of system model is as follow:

Step 1: Resource registration to GIS takes place.

Step2: User submit task with complete specification to resource broker through grid portal.

Step3: Task Agent (TA) places all submitted tasks in a task set and activates Resource Discovery Agent (RDA).

Step4: RDA queries GIS regarding resources.

Step5: GIS returns the static attribute of resources such as number of machines, number of processing elements (PE), MIPS (Million Instruction Per second) rating of each PE, allocation policy.

Step6: RDA send query to registered resources for their availability status.

Step7: RDA gets the status information and makes it available to TA.

Step8: TA, by deploying ant algorithm, select a resource for next task assignment and dispatch the task to selected resource through RDA.

Step9: After task execution, results are received from resources and are returned to user by TA.

IV. **R**ESOURCE **O**RIENTED **A**NT **A**LGORITHM (**ROAA**)

Ant algorithm [3] is inspired on an analogy with real life behavior of a colony of ants when looking for food, and is effective algorithm for the solution of many combinatorial optimization problems. Investigations show that: Ant has the ability of finding an optimal path from nest to food. On the way of ants moving, they lay some pheromone on the ground. While an isolated ant moves essentially at random, an ant encountering a previously laid trail can detect it and decide with high probability to follow it, thus reinforcing the trail with its own pheromone. The probability of ant chooses a way is proportion to the concentration of a way's pheromone. To a way, the more ants choose, the way has denser pheromone, and the denser pheromone attracts more ants. Through this positive feedback mechanism, ant can find an optimal way finally.

In ROAA, the pheromone is associated with resources rather than path. The increase or decrease of pheromone depends on task status at resources. The main objective of algorithm is reduction in total cost and execution time. Let the number of tasks (ants) in task set T maintained by task agent is P and the number of registered resources is Q.

When a new resource R_i is registered to GIS, then it will initialize its pheromone based on:

$$\tau_i(0) = N \; x \; M ,$$

where N represent the number of processing elements and M corresponds to MIPS rating of processing element.

Whenever a new task is assigned to or some task is returned from R_i, then the pheromone of R_i is changed as:

$$\tau_i^{\,new} = \rho * \tau_i^{\,old} + \Delta \tau_i ,$$

where Δ_i is pheromone variance and $\rho,\ 0 < \rho < 1$ is a pheromone decay parameter. When a task is assigned to R_i, its pheromone is reduced i.e. $\Delta_i = -C$, where C represents the computational complexity of assigned task. When a task is successfully returned from R_i, $\Delta_i = \Phi * C$, where Φ is the encouragement argument. On the other hand, if task failure occurs at R_i, $\Delta_i = \Theta * C$, where Θ is the punishment argument. It is clear that pheromone increases when task execution at a resource is successful.

The possibility of next task assignment to resource R_j is computed as: $p_j(t) = \dfrac{[\tau_j(t)]^{\alpha} * [\eta_j]^{\beta}}{\sum_r [\tau_r(t)]^{\alpha} * [\eta_r]^{\beta}}$, where $j, r \in available\ resources$. $\tau_j(t)$ denotes the current pheromone of resource R_j and η_j represents the initial pheromone of R_j i.e. $\eta_j = \tau_j(0)$. α is the parameter on relative performance of pheromone trail intensity, β is the parameter on relative importance of initial performance attributes. The process of resource oriented ant algorithm is shown below.

Procedure Resource_Ant_Algorithm
Begin
Initialize parameters and set pheromone trails.
While (Task set $T \neq \Phi$) **do**
Begin
Select next task t from T.
Determine the next resource R_i for task assignment having higher transition probability among all resources (high pheromone intensity), i.e. $p_i(t) = \max_{l \in Q} p_l(t)$.
Schedule task t to R_i and remove it from T i.e. $T = T - \{t\}$.
If (Any task completion or failure occurs) **then**
Update pheromone intensity of corresponding resource and transition probability of all registered resources.
End
End

V. SIMULATION RESULTS

We analyze the ROAA using GridSim simulator [13]. Resource and tasks used in simulation are modeled as shown in Table1. The proposed algorithm is compared with the algorithm already used in GridSim. This algorithm selects the next resource for task assignment in a random fashion (RandomAlgorithm). Other simulation parameters are:

$$\rho = 0.9,\ \alpha = 0.5,\ \beta = 0.5,\ \Phi = 1.1,\ \Theta = 0.8$$

In our simulation, we use 10 heterogeneous grid resources and we run a simulation at five levels of workloads-50 tasks, 100 tasks, 150 tasks, 200 tasks and 250 tasks.

Table 1: Simulation Parameters

Parameter	Value
Number of Resources	10
Number of PE per Resource	4-16
MIPS of PE	300-900
Resource Cost	9 G$
Total Number of Tasks(Ants)	50-250
Length of Task	10000-18000 MI (Million Instructions)

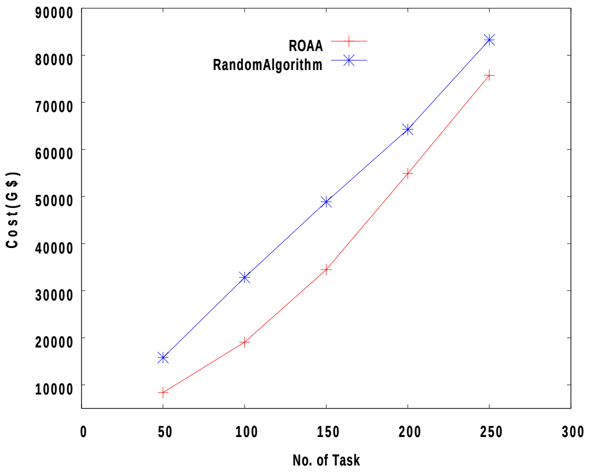

Figure 2: Comparison of Total Cost

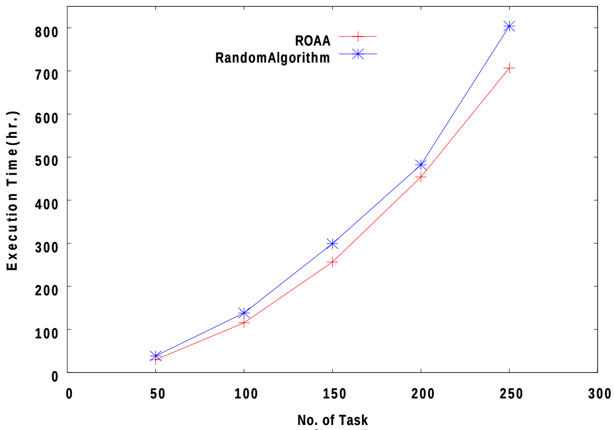

Figure 3: Comparison of Total Execution Time

Each task is submitted into grid system randomly. Fig. 2 and Fig. 3 show that system using ROAA outperforms system using RandomAlgorithm in terms of execution time and cost.

VI. CONCLUSION

In this paper, we described grid resource allocation using an ant algorithm. The results of the experiments are also presented and the strengths of the algorithm are investigated. The simulation results demonstrate that the ROAA algorithm increases performance in terms of reduction in total execution time and cost. In future work, we plan to add the applications of ant level load balancing in addition to implementing this mechanism in a more realistic environment.

REFERENCES

[1] I. Foster, C. Kesselman, and S. Tuecke," The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International Journal of High Performance Computing Applications, vol. 15(3), pp. 200-222, 2001.

[2] C. Chapman, M. Musolesi, W. Emmerich, and C. Mascolo, "Predictive Resource Scheduling in Computational Grids," IEEE Parallel and Distributed Processing Symposium, 2007 (IPDPS 2007), pp. 1-10, 26-30 March 2007.

[3] K. Krauter, R. Buyya, and M. Maheswaran, "A Taxonomy and survey of Grid Resource Management Systems for Distributed Computing," Software: Practice and Experience (SPE) Journal, Wiley Press, USA, vol. 32(2), pp. 135-164, 2002.

[4] M. Dorigo, and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," IEEE Transactions on Evolutionary Computation, vol. 1(1), pp. 53-66, 1997.

[5] A. Colorni , M. Dorigo, and V. Maniezzo, "Ant colony system for job-shop scheduling," Belgian Journal of Operations Research Statistics and Computer Science, vol. 34(1), pp. 39-53,1999.

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

[7] V. Maniezzo, and A. Colorni, "The Ant System Applied to Quadratic Assignment Problem," IEEE Transaction on Knowledge and Data Engineering, vol. 11(5), pp. 769-778, 1999.

[8] T. D. Braun, H. J. Siegel, N. Beck, L. L. Bölöni, M. Maheswaran, A. I.Reuther, J. P. Robertson, M. D. Theys, B. Yao, D. Hensgen and R. F. Freund , "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing, vol. 61(6), pp. 810-837, 2001.

[9] Li, X. Peng, Z. Wang, and Y. Liu, "Scheduling Interrelated Tasks in Grid Based on Ant Algorithm," Journal of System Simulation, 2007.

[10] T. Stutzle, "MAX-MIN Ant System for Quadratic Assignment Problems" Technical Report AIDA-97-04 Intellectics Group, Department of Compute Science, Darmstadt University of Technology, Germany, July 1997.

[11] K. M. Sim, and W. H. Sun "Multiple Ant-Colony Optimization for Network Routing", Proceedings of First International Symposium on Cyber Worlds, pp. 277-281, 6-8 Nov. 2002.

[12] Z. Xu, X. Hou, and J. Sun, "Ant Algorithm-Based Task Scheduling in Grid Computing," Proceeding of the IEEE Conference on Electrical and Computer Engineering, pp. 1107-1110, 2003.

[13] R. Buyya, and M. Murshed, "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing," Concurrency & Computation: Practice & Experience, vol. 14, pp. 1175-1220, 2002.