International Journal of Computer and Communication Technology

Volume 7 | Issue 3 Article 8

July 2016

DYNAMIC JOB SCHEDULING IN GRID COMPUTING

JANI KUNTESH KETAN
Computer Science & Engineering Department CHARUSAT UNIVERSITY changa, Dist. Anand, India,
kunteshjani@gmail.com

ARPITA SHAH
Computer Science & Engineering Department CHARUSAT UNIVERSITY changa, Dist. Anand, India,
arpitashah.ce@ecchanga.ac.in

DYNAMIC JOB SCHEDULING IN GRID COMPUTING

JANI KUNTESH KETAN[1] & ARPITA SHAH[2]

[1,2]Computer Science & Engineering Department CHARUSAT UNIVERSITY changa, Dist. Anand, India
E-mail : kunteshjani@gmail.com, arpitashah.ce@ecchanga.ac.in

Abstract - Grid computing is growing rapidly in the distributed heterogeneous systems for utilizing and sharing large-scale resources to solve complex scientific problems. Scheduling is the most recent topic used to achieve high performance in grid environments. It aims to find a suitable allocation of resources for each job. A typical problem which arises during this task is the decision of scheduling. It is about an effective utilization of processor to minimize tardiness time of a job, when it is being scheduled. Scheduling jobs to resources in grid computing is complicated due to the distributed and heterogeneous nature of the resources. The efficient scheduling of independent jobs in a heterogeneous computing environment is an important problem in domains such as grid computing. In general, finding optimal schedule for such an environment using the traditional sequential method is an NP-hard problem whereas heuristic approaches will provide near optimal solutions for complex problems. The Ant colony algorithm, which is one of the heuristic algorithms, suits well for the grid scheduling environment using stigmeric communication.

Keywords*-component; Grid Computing; job Scheduling; Ant Colony Algorithm.*

I. INTRODUCTION

The term Grid computing originated computer power as easy to access as an electric power grid. Grid computing is the process of applying the resources of many computers in a network to a single problem at the situations like - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. It involves the use of software that can divide and form out pieces of a program to as many as several thousand computers. It looks as distributed and large-scale cluster computing and as a form of network-distributed parallel processing [1].Distributed systems consist of multiple computers that communicate through computer networks. Research by [2] defined that cluster and grid computing are the most suitable ways for establishing distributed systems. Grid computing is proposed to overcome this problem where various resources from different geographic area are combined in order to develop a grid computing environment. To achieve the promising potentials of tremendous distributed resources, effective and efficient scheduling algorithms are fundamentally important. The scheduling problem is defined NP-hard problem [4] and it is not trivial. 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. Once started, they keep running on the same resources without interruption. However, for the dynamic scheduling, reevaluation is allowed of already taken assignment decisions during job execution [3].

II. ORGANIZATION OF PAPER

The organization of the paper further is as follows. The literature review is presented in Section III, existing system is analyzed in Section IV and summary & open issues are discussed in Section V.

III. LITERATURE REVIEW

In the past few years, researchers have proposed scheduling algorithms for parallel system [5 - 9]. However, the problem of grid scheduling is still more complex than the proposed solutions. Therefore, this issue attracts the interests of the large number of researchers [10- 14]. Current systems [15] of grid resource management was surveyed and analyzed based on classification of scheduler organization, system status, scheduling and rescheduling policies. However, the characteristics and various techniques of the existing grid scheduling algorithms are still complex particularly with extra added components.

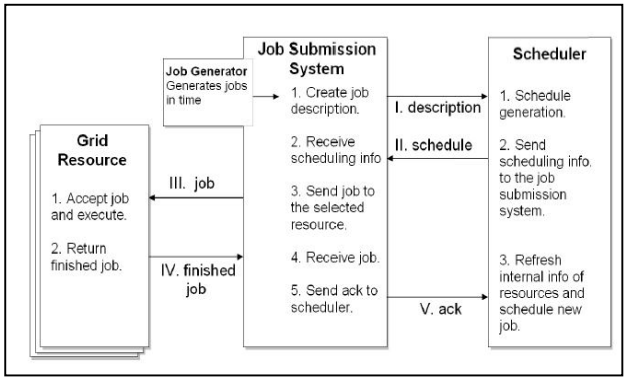

Fig. 1 : Job scheduling work flow

At the present time, job scheduling on grid computing is not only aims to find an optimal resource to improve the overall system performance but also to utilize the existing resources more efficiently. Recently, many researchers have been studied several works on job scheduling on grid environment. Some of those are the popular heuristic algorithms, which have been developed, are min-min [16], the fast greedy [16], tabu search [16] and an Ant System [17].

In 1999, the Ant Colony Optimization (ACO) metaheuristic was proposed by Dorigo, Di Caro and Gambardella, which has been successfully used to solve many NP-problem, such as TSP, job shop scheduling, etc. In the past few years, several researchers proposed solutions to solve grid scheduling problem by using ACO [20]. Several studies have been trying to apply ACO for solving grid scheduling problem. Z. Xu et al [18] proposed a simple ACO within grid simulation architecture environment and used evaluation index in response time and resource average utilization. E. Lu et al. [19] and H. Yan et al. [20] also proposed an improved Ant Colony algorithm, which could improve the performance such as job finishing ratio. However, they have never used the various evaluation indices to evaluate their algorithm. The ACO becomes very popular algorithm to apply for solving grid scheduling problem.

IV. ANALYSIS OF RELATED WORK ON ACO

Before Ant colony optimization algorithms [Dorigo 1996] are multi-agent systems, which consist of agents with the collective behavior (stigmergy) of ants for finding shortest paths. Ant colony algorithms were inspired by the observation of real ant colonies. Ants are social insects, that is, insects that live in colonies and whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. Social insects have captured the attention of many scientists because of the high structuration level their colonies can achieve, especially when compared to the relative simplicity of the colony's individuals. An important and interesting behavior of ant colonies is their foraging behavior, and, in particular, how ants can find shortest paths between food sources and their nest. While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called pheromone, forming in this way a pheromone trail. When more paths are available from the nest to a food source, a colony of ants may be able to exploit the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back. It is also interesting to note that ants can perform this specific behavior using a simple form of indirect communication mediated by pheromone laying, known as stigmergy.

```
1.  procedure ACO
2.  begin
3.  Initialize the pheromone
4.  While stopping criterion not satisfied do
5.  Position each ant in a starting node
6.  Repeat
7.  for each ant do
8.  Chose next node by applying the state
    transition rate
9.  end for
10. until every ant has build a solution
11. Update the pheromone
12. end while
13. end
```

III Pseudo code for Existing Ant colony Algorithm

A. Modified Ant Colony Algorithm for Grid Scheduling

The modified ant colony algorithm has changed the basic Pheromone updating rule of original ant colony algorithm. The improved pheromone updating rule is given by :

$$\tau ij(t)new=[\{(1-)/(1+\rho)\}*\tau ij(t)old]+[\{\rho/(1+\rho)\}*\Delta\tau ij(t)] \quad (1)$$

Where,

$\tau ij(t)$ =Trail intensity of the edge(i,j).

ρ = Evaporation rate.

$\Delta\tau ij(t)$ = Additional pheromone when job moves from scheduler to resource.

The proposed Ant colony algorithm as a whole is a best suited method for tracking problem with large data sets. The above approach was simulated using GRIDSim toolkit and was found to be working efficiently and effectively. Experimental test carried out for a varied range of input set to ascertain the efficiency of the algorithm. From the results it is clearly evident that the proposed Ant colony algorithm offers better optimization a very fast rate.

B. Scheduling in Computational Grid with a New Hybrid Ant Colony Optimization Algorithm

The efficiency depends on makespan and flow-time. The makespan measures the throughput of the system and flow-time measures its QOS . The main objective of the proposed algorithm is to reduce the makespan and to converge towards the optimal solution in a very faster manner. Ant algorithm for scheduling initially started with the value of ETij matrix. ETij matrix defines the expected time taken by machine j to complete task i. Here jobs are considered as independent to each other. ETij matrix consists of N×M entries where N is number of independent jobs and M is number of resources available in grid environment. Increasing the pheromone levels associated with a chosen set of good solutions makes the algorithm faster to converge to a solution. Hence on modifying the pheromone updating rule and probability matrix the solution is converging in a fast manner than the

existing ACO algorithm. The probability matrix calculation uses the the same transition rule to select the job which is to be executed next in the machine. This algorithm guarantees efficient resource allocation of the machines.

C. An Improved Ant Algorithm For Job Scheduling In Grid Computing

In this approach, the general adaptive ant algorithm is used to solve the job scheduling problem in the grid simulator environment. The transition probability is trade-off between visibility, that means the power performance of resource should be chosen with high probability, and trail intensity that means if on path j there is a lot of traffic then it is high desirable. Once some resources load heavy, it comes into being a bottleneck in the grid and influences to completing of jobs. Therefore the load balancing factor is introduced in the ant algorithm to improve the load balancing capability. We define the load balancing factor λ of the resource j, which is related to the job finishing rate in the resource j. In the algorithm, the trail intensity will be changed from ΔT_J to $\Delta T_J + C\lambda_J$ (C>0 is a coefficient of the load balancing factor), the more jobs finished, the more increases the trail intensity, contrarily, the more jobs not completed, the more decreases the trail intensity. The improved algorithm practice will improve the load balance status for the resources in the grid.

D. Ant Colony Algorithm for Job Scheduling in Grid Computing

This proposed algorithm aims to minimize the computational time of each job that must be processed by available resources in grid computing system. The algorithm will select the resources based on the pheromone value on each resource. A matrix that contains the pheromone value on each resource has been used to facilitate the selection of suitable resources to process submitted jobs. The proposed algorithm has been implemented in the grid system architecture which consists of four main components namely the grid information server, grid resource broker, jobs and resources After all ants have constructed a solution, the pheromone trails are updated according to the following formula:

$$T_{Jr}(t+1) = (1-\rho).T_{Jr} + \rho\Delta T_{Jr}^{bs} \quad (2)$$

where $\Delta \tau jr\ best$ = 1/Lbest. This global pheromone update is limited to a specific upper and lower trail limit. The ant which is allowed to add pheromone may be the *iterationbest 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 resource with high pheromone value will be selected by grid resource broker. So j3 will be processed by r2. After assigning j3 to r2, the local pheromone update is performed to the second row of r2. Column 3 is no longer needed because j3 has been assigned. The scheduling process in the proposed algorithm is based on the combination of local pheromone update and trail limits. This proposed technique is different from the previous algorithm based on its initial pheromone value and the used of the PV matrix. The initial pheromone value for this algorithm considered the estimated transmission time and characteristics of each jobs and resources. The local pheromone trail update will reduce the amount of pheromone in assigned resource, to ensure the resource 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 technique to control the value of pheromone updated on each resource to ensure that already assigned optimal resources will not be chosen for newly submitted jobs. The proposed algorithm is simple to be implemented due to the existing of information of each resources and jobs. This algorithm was able to minimize the completion time of each job.

E. An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment

Here the general adaptive ACO algorithm is basically the ACS (Ant Colony System) , is applied to solve the scheduling problem in grid environment. Four main key terms of ACO algorithm are listed and defined below: The expected time complete (ETC) is defined as the amount of time that the job is processed at a machine, and it has no load to the assigned job. Where the complete time (CT) is defined as the wall clock time, which machine completes for each job. Then, *CTij = aj + rj + ETCij*, where *aj* is arrival time of a job *j*th, *rj* is a release time of a job *j*th and *i* is a machine *ith*. The main important of ACO algorithm is to utilize the graph representation. therefore, the design of the graph is to identify the problem to connect the arcs correspondent for each job on each machine. Let *M* be a set of machines *{m1, m2, m3,.., mm}* and let *J* be a set of jobs *{j1, j2, j3, …, jn}*, and *n > m*. Therefore, the graph *G = (M, {CTij}mxn)*. The problem is to find the optimal resources for the jobs. It can minimize the total tardiness time. The algorithm can find an optimal processor for each machine to allocate to a job that minimizes the tardiness time of a job when the job is scheduled in the system.

V. SUMMARY AND OPEN ISSUES

Dynamic job scheduling in grid computing is an NP-hard problem which can be solved using various techniques of cooperative sub-optimal category. Ant colony system is one such type of cooperative sub-optimal solution which can meet dynamically changing needs of a heterogeneous environment like computational grids. The open issues are:

- Hybridization to produce high throughput computing
- Evaluate the different cost measures such as make-span time, Grid Efficiency and job error ratio

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] K. Yan, S. Wang, S. Wang, and C. Chang, "Towards a hybrid load balancing policy in grid computing system," *Expert Systems with Applications*, vol. 36, pp. 12054-12064, 2009.

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

[4] D. Fernandez-Baca (1989), "Allocating Modules to Processors in a Distributed System", *IEEE Transactions on Software Engineering*. Vol.15(11): Pages 1427-1436.

[5] D.G. Feitelson. "Packing schemes for gang scheduling", In *2nd Workshop on Job Scheduling Strategies for Parallel Processing*, volume LNCS 1162, pages 89–100, 1996.

[6] D.G. Feitelson, L. Rudolph, U. Schwiegelshohn, K.C. Sevcik, and P.Wong. "Theory and practice in parallel job scheduling", In *3rd Workshop on Job Scheduling Strategies for Parallel Processing*, volume LNCS 1291, pages 1–34, 1997.

[7] J. Krallmann, U. Schwiegelshohn, and R. Yahyapour. "On the design and evaluation of job scheduling algorithms", In *5th Workshop on Job Scheduling Strategies for Parallel Processing*, volume LNCS 1659,pages 17–42, 1999.

[8] J. M. van den Akker, J. A. Hoogeveen, and J. W. van Kempen. "Parallel machine scheduling through column generation: Minimax objective functions", In Y. Azar and T. Erlebach, editors, *European Symposium on Algorithms*, volume 2996 of *Lecture Notes in Computer Science*, pages 648-659. Springer, 2004.

[9] R.D. Nelson, D.F. Towsley, and A.N. Tantawi. "Performance analysis of parallel processing systems", *IEEE Transactions on Software Engineering*, 14(4):532–540, 1988.

[10] V. Hamscher, U. Schwiegelshohn, A. Streit, R.Yahyapour, "Evaluation of job-scheduling strategies for grid computing", *Proceedings of First IEEE/ACM International Workshop on Grid Computing, Lecture Notes in Computer Science*, vol. 1971, Springer, Berlin, 2000, pp. 191–202.

[11] V. Subramani, R. Kettimuthu, S. Srinivasan, P. Sadayappan, "Distributed job scheduling on computational grids using multiple simultaneous requests", *Proceedings of the 11th International Symposium on High Performance Distributed Computing*, 2002, pp.359–366.

[12] H. Shan, L, Oliker and R. Biswas, "Job Superscheduler Architecture and Performance in Computational Grid environments", *Proceedings of the ACM/IEEE SC2003 Conference (SC03)*, 2003.

[13] C. Ernemann , V. Hamscher and R. Yahyapour, "Benefits of Global Grid Computing for Job Scheduling", *Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04)*, 2004.

[14] K. Li, "Job scheduling and processor allocation for grid computing on Metacomputers ", *Journal of Parallel and Distributed Computing, Elsevier*, 2005

[15] K. Krauter, R. Buyya and M. Maheswaran, "A taxonomy and survey of Grid resource management systems for distributed computing", Software Pract. Exp. 2 (2002) 135–164.

[16] 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 (2001), "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): Pages 810-837.

[17] G. Ritchie and J. Levine, "A fast, effective local search for scheduling independent jobs in heterogeneous computing environments".

[18] Z. Xu, X. Hou and J. Sun, "Ant Algorithm-Based Task Scheduling in Grid Computing", *Electrical and Computer Engineering, IEEE CCECE 2003, Canadian Conference*, 2003.

[19] E. Lu, Z. Xu and J. Sun, "An Extendable Grid Simulation Environment Based on GridSim", *Second International Workshop, GCC 2003*, volume LNCS 3032, pages 205–208, 2004.

[20] H. Yan, X. Shen, X. Li and M. Wu, "An Improved Ant Algorithm for Job Scheduling in Grid Computing", *In Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, 18-21 August 2005.