COMPUSOFT, An international journal of advanced computer technology, 3 (6), June-2014 (Volume-III, Issue-VI)

ISSN:2320-0790

An International Journal of Advanced Computer Technology

Grid scheduling: Comparative study of MACO & TABU search

[1]Aanchal Sewaiwar, [2]Utkarsh Sharma
[1]MCIT (LNCT), [2]LNCT (IT)
RGPV, BHOPAL

Abstract—Grid computing is progressively considered as a Next-generation computational platform that supports wide-area parallel and distributed computing. Scheduling jobs to resources in grid computing is difficult due to the distributed and heterogeneous nature of the resources. In Grid computingfinding optimal schedules for such an environment is (in general) an NP-hard problem, and so heuristic technique must be used. The aim of grid task scheduling is to achieve highsystem throughput and to distribute various computing resources to applications. Many different algorithms have been proposed to solve this problem. Some of these algorithms are based on heuristic techniques that provide an Optimal or near optimal solution for large grids.In this paper shows the grid scheduling algorithms and comparison between them.

keywords—Grid Computing,job scheduling,ACO,MACO,tabu search.

I. INTRODUCTION

Grid computing is the collection of computer resources from various locations to reach a common goal. The **grid computing** can be thought of as a distributed computing with non-interactive workloads that involve a large number of files.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[2]. 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.

There are two types of scheduling namely static scheduling and dynamic scheduling in grid computing system. For the static scheduling, jobsare assigned to suitable resources before their execution begin. However, for the dynamic scheduling, re-evaluation is allowed of already taken assignment decisions during job execution [1][2].

As the grid tasks scheduling faced with an NP-complete problems, which has aroused the concern of many scholars, becomes a focus of grid study. About the tasks scheduling algorithm, domestic and foreign had already done massive researches. At present, tasks scheduling algorithms are mainly based on Minimum Completion Time (MCT), User-Directed Assignment (UDA), Minimum Execution Time (MET), Fast Greedy, Max-min, genetic algorithm (GA), Simulated Annealing (SA) and so on[1][2].

In this paper we study about the various types of tasks scheduling techniques like: ant colony optimization, multiple ant colony optimization, enhance multiple ant colony optimization and taboo search algorithm.

A. ACO (Ant Colony Optimization) [1] algorithm is presented by the Italian researcher named M. Dorigo [18][19]. The algorithm imitates the food-seeking behaviour of ant. Ant Colony Algorithm (ACA) is mainly used to solve dynamic and static multi-dimensional optimization problems. TSP (Traveling Salesman Problem) is one of classic static problems.

B. Recently, ant colony optimization (ACO) has been suggested to solve the task scheduling problems in grid computing. 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. MACO uses multiple ant colonies to search for alternatives to an optimal path. One of the motivations of MACO is to optimize the performance of a congested network by routing calls via several alternatives routes to prevent possible congestion along an optimal path.

C. Tabu Search algorithm is one of scheduling techniques in grid computing. TS is a metaheuristic scheduling method used to guide optimization algorithm in the search for a globally optimal solution. The basic norm of TS is to pursue Local Search whenever it encounters a local optimum by allowing non-improving moves; cycling back to previously visited solutions is prevented by the use of memories, called Tabulists,that store the recent history of the search. TS deal with various techniques for making the search more effective.

In this paper we study all these four grid task scheduling algorithms and show the comparative results between them. Section II describes the use of Ant colony algorithms in grid computing. In Section III describes the use of multiple ant colony algorithms in grid computing. In Section IV, computational experiments and comparison studies are reported. And some concluding remarks are made in Section V.

II.RELATED WORKS ON ACO IN GRID COMPUTING ENVIRONMENT

Ant colony optimization (ACO) was first introduced by Marco Dorigo as his Ph.D. thesis and was used to solve the TSP problem [2]. Ant colony algorithm is an intended ecological algorithm, which comes from the nature of ants to find the shortest path from nest to food and to find back to the nest path method. It is a distributed and heuristic algorithm. The ACO algorithm has excellent properties such as parallelism, robustness and good effects in solving a variety of combinatorial optimization problems. Its formal description is as follows:Supposed an n-city TSP problem, the number of artificial ants is m, behaviour of each artificial ant conforms to the following logic:
1) According to the concentration of pheromone on the path, the next path is selected;
2) The cycle of path walked is not selected for the next path;
3) When a cycle completes, the pheromone is released in line with the entire length of the path, the pheromone concentration on the travelled path is updated. *Tij*(t) indicates the pheromone concentration at time t, when the ants completes one cycle, the corresponding edge of the pheromone concentration is updated according to equation(1).

$$Tij(t+1)= \rho \times Tij(t)+\Delta Tij(1)$$

ρ : The constant coefficient in the range [0,1]
ΔTij: Determined by $\Delta Tk = \sum_{k=1}^{m} \Delta T_{ij}^{k}$ means that the k ants increase pheromone concentration at time t to t+1 between the edge (i, j), equation(2) determines its value.

$$\Delta T_{ij}^{k} = \frac{Q}{L_k} \qquad (2)$$

Q: Constant used to represent that the ants releases thetotal amount of pheromone with completing a full path to search.
Lk: The total cost of k ants, which is equal to the sum of the cost of the required segment of the path. Higher the total cost of the ants is, lower the concentration of pheromone releases in the unit path. It is clear that the ants will not release pheromone on not through the path. According to the basic formal description of ant colony algorithm, the ant colony algorithm is essentially a pheromone for the media, indirect spreads optimal solution information through the swarm intellect, solves the optimal solution with the gradual convergence. The ant colony algorithm has the ability to find better solutions. The algorithm use the principle of positive response, and speeds up the process of evolution, so the ant colony algorithm for combinatorial optimization problems is faster. Of course, this feature also makes the accuracy of the ant colony algorithm for the optimal solution to not high, and they often obtain suboptimal solutions. However, many applications of gird computing have real-time requirements. Shorten the time are more important to obtain the optimal scheduling in many cases. The ant colony algorithm is more in line with the real-time requirements of gird computing.

A. The Framework of the ACO Approach:

Initializationof the algorithm. All pheromone units and parameters are initialized.

1) ***Initialization of ants*:** Assume that a group of *M* ants are used in the algorithm. At the beginning of each iteration, all ants are set to initial state. Each ant chooses aconstructive type (forward or backward) and a heuristic type(duration-greedy, cost-greedy, or overall-greedy). Based onthe constructive type, each ant builds its tackling sequenceof services.
2) ***Solution construction*:***M* ants set out to build *M*solutions to the problem. The construction procedureincludes *n* steps. *n*is the number of services in the workflow.In each step, each ant picks up the next service in its tacklingsequence and maps it to one implementation out of theservice's implementation domain using pheromone andheuristic information. The algorithm also estimates theearliest start time and earliest end time of services in termsof the information of partial solution built by each ant. Thisinformation is helpful to guide the search behavior of ants.
3) ***Local updating*:** Soon after an ant maps a service *Si* toSP_i^j, the corresponding pheromone value is updated by alocal pheromone updating rule.
4) ***Global updating*:** After all ants have completed theirconstructions, global pheromone updating is applied to thebest-so-far solution. The cost and makespan of all solutionsare evaluated. The pheromone values related to thebest-so-far solution is significantly increased. Moreover, some parametersof the algorithm are adaptively adjusted inthis procedure.
5) ***Terminal test*:** If the test is passed, the algorithm is end.Otherwise, go to step 2) to begin a new iteration.

III. RELATED WORKS ON MACO IN GRID COMPUTING ENVIRONMENT

Recently, ant colony optimization (ACO) has beenusing to solve the task scheduling problem in gird computing. But ACOapproaches using a single colony structure may suffer fromspecific local optima because of its trend to use thepositive feedback mechanism of pheromone value, multiple antcolonies optimization (MACO) is employed to avoid thisby using numerous ant colonies to solve combinatorialoptimization problems helpfully. To improve theperformance of ACO approaches, MACO considers bothpositive and negative feedbacks in searching results,sharing the search information, and exploring a large areaof the search space with mutual teamwork of antcolonies. So MACO approaches have been explored forseveral optimization problems.

MACO is to be appropriate approach to improve theperformance of ACO algorithm. This algorithm offersgood opportunity to explore a huge area of the searchspace and find optimal solution. Every colonies constructtheir solutions in parallel [13] and interaction mechanismsare designed for sharing their experiences among colonies.Assume that M ant colonies are used to tackle thescheduling problem, and each colony contains N ants forthe search procedure.

The Framework of the MACO Approach:

1. initialization of the algorithm is presented,
2. Local and global phenomenon updating is performed,
3. 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 theycomplete their iterating and updating pheromonearrays respectively, until one ant colony reaches itslocal optimum solution. Every ant colony system have its ownspheromone array and parameters and records in its localoptimum solution. Furthermore, once an ant colonysystem arrives at its local optimum solution, updates itslocal optimum solution and sends the current solution to globalbest-found canter. In general, the approach can be brieflysketched as follows.

1) Initialization of Algorithm:

Assume that M colonies of ant would be used to tackle thescheduling problem, and each colony contains N ants forthe search procedure. We denote by ant (m, n) the nth antin the mth colony. In this firstly, ants are distributedon computing nodes randomly, and the pheromone value$\tau_j^m$of the mth colony on node c_j is initialized as a smallvalue.

1.1. Interaction of MACO System:

For interaction between ants in the same colony, ant are used the pheromone as a mechanism but this mechanism are also work among ant colonies. As traditional approaches of ant colony system, each colony hasits own pheromone to interaction between the ants of thesame colony. Furthermore, pheromone value is alsoused for the interaction between ant colonies,and the interaction between coloniesis achieved by evaluating the pheromone values ofdifferent colonies.

More specifically, the evaluatedpheromone τ_j on node c_j in terms of pheromone values ofall colonies is defined as follows,

$$\tau_j = \frac{\sum_{m\epsilon\ [1,M]} \tau_j^m}{M} \quad (3)$$

Where M-no of ant colonies in system, τ_j – is computed interms of pheromone values of every colonies.

The pheromones evaluation mechanism averages thepheromone values of each colony, which representsinformation of all colonies in system. Based on the average of theavailable experiences of ants of all colonies in a system, an ant willdecide how to choose an edge.

2) Pheromone Updating:

As ant (m, n) completes its trip, local pheromoneupdating is applied on the visited nodes.Explicitly, if ant (m, n) assigns task Ti to node Cj, thelocal pheromone update is given by:

$$\tau_j^m = (1-\rho)\tau_j^m + \rho\Delta\tau_j^{mn} \qquad (4)$$

Where ρ- Evaporation rate

In global pheromone updating phase, after all ants of allcolonies construct their solutions, and the ant achieving thebest-so-far solution in colonies, will deposit an amountof pheromone on edges of its path according to thefollowing rule

$$\tau_j^m = (1-\lambda)\tau_j^m + \lambda\Delta\tau_j^{mn} \qquad (5)$$

Where λ- Evaporation

Thus, the ant finding the solution with the smallestmakespan can place a larger intensity of the pheromone onits trip.

3) Solution construction:

In this phase, ant (m, n) moves through computing nodesand assigns task $T_i$to node $C_j$probabilistically in terms ofpheromone and heuristic information till all tasks havebeen allocated. The probability $P^{mn}{}_{ij}$for ant (m,n) toassign the task to node is defined as;

$$P^{mn}{}_{ij} = \tau_j \text{ƅ}^{mn}{}_{ij} / \sum \tau_j \text{ƅ}^{mn}{}_{ij} \qquad (6)$$

Where $\text{ƅ}^{mn}{}_{ij}$the heuristic information value for evaluatingthe assignment of task $T_i$to node $C_j$for ant (m, n). τ_i- iscalculated in terms of pheromone values of all colonies.These approaches terminated when the global pheromone updating is not improved in successive predefine number ofsolutions.

IV. RELATED WORKS ON TABU SEARCH IN GRID COMPUTING ENVIRONMENT

Tabu search was introduced as a high-level algorithm that is used other specific heuristic to guide the search. The goal is performing an intellectual exploration of the search space that would of the search space that would allow to avoid receiving trapped into local optima. In Algorithm we show a generic pseudo-code for Tabu Search. This general template offers several different possibilities for highly specifying our tabu search algorithm for a concrete problems just by designing its inner heuristics and appropriate data structures.

As can be seen from the Algorithm, one of the distinguishing features of tabu search versus other heuristics is the use of a historical memories, which consists of a short term memory i.e. recency with information on recently visited solutions, and a long term memory i.e. frequency storing information gathered during the whole complete exploration process.

Algorithm: Template for Taboo Search Algorithm

```
1.  Compute an initial solution s; let ˆs   s;
2.  Rearrange the tabu and aspiration conditions;
3.  while not termination-condition do Generate a subset
    of solutions N∗(s) that do not violate the taboo
    conditions or hold the aspiration criteria;
4.  Choose the best s′ 2 N∗(s) with respect to the cost
    function; s   s′;
5.  if improvement(s′, ˆs)) then ˆs   s′;
6.  Update the recency and frequency;
7.  if (intensification condition) then Perform
    intensification procedure;
8.  if (diversification condition) then Perform
    diversification procedures;
9.  end while
10. return ˆs;
```

The movements of these two lists are considered in tabu and then they can't be used. Thus, some aspiration criteria are needed for removing the tabu movements. In addition, we need to specify some inner heuristics like the local search, used for exploring the neighbourhood solution, or the intensification and diversification processes, for appropriately managing the exploration/exploitation trade-off on the search space. We describe the components of our algorithm:

- Solution Representation: S is represented as a vector of size nb_jobs, in which S[i] shows the machine where job i is assigned to.
- Movements: Two types of movements are considered for this representation [13]: transfer and swap. Transfer moves a job from first machine to second, while swap exchange two jobs assigned to different machines.
- Initial solution: It is generated with the Min-Min method. It starts by computing a matrix completion[i][j] with in the time when every job would finish in each machine (completion[i][j] = ETC[i][j] + ready[j]). Then we choose the job k and machine m with the initial completion time (completion[k][m]≤ completion[i][j] ∀i, j), remove job k from the job set, and update completion[i][j] for the another jobs. This process is repeated until all jobs are assigned.
- Historical memory: There are three different memories have been used:

  1) The recency memory (a tabu list with the last time each job was assigned to every machine)

  2) Tabu hash table stored visited solutions.

  3) Frequency memory (storing how many times Every job has been assigned each machine).
- Aspiration criteria: In tabu search algorithm two criteria are used to accept tabu movements:
  1) fitness(they are accepted when yielding to better solutions)
  2) makespan (improve the makespan of the solution on movements are allowed).
- Neighbourhood exploration: It is done by means of load balancing: movements made among jobs assigned to most-loaded and less-loaded machines. Our tabu search consider the first neighbour's providing the best improvements in terms of the completion time, i.e., the time when a machine finishes all its tasks, equation – in the solution, or the least worst movement in case no better solutions are found.

$$completion[m] = ready\ times[m] + \sum_{\{j \epsilon Jobs\ \mid\ schedule\ [j]=m\}} ETC[j][m] \qquad (7)$$

- Intensification: It is executed when there is shows that the current solution is in a promising area. Thus, a deeper exploration of this area is done by rewarding (attributes of) the current solution, making their presence in the new ones. For that we use a combination of these three strategies (they are applied in sequentially until no improvements are possible):
  1) The most promising attributes of solutions (from the frequency memory) are rewarded. And the most frequent job assignments are chosen with probability 0.75, while in other case (probability 0 .25) a roulette-wheel is used in the assignment.
  2) The values for the maximum and minimum load parameters are temporarily changed by using the existing state of the grid (in terms of the workloads on machines) with the only condition that the number of resulting transfers and swaps are limited by their respective upper bounds. Then, the neighbourhood is defined by these new values.
  3) The structure of the neighbourhood node is changed by applying all the possible swap movements between two

machines, and then performing just one transfer from one machine to the other.

- Diversification: In tabu search algorithm we use three *soft* diversification methods (performing slight perturbations to the solution), and a *strong* diversification (hard perturbation):
    1) Using the job distribution (or influential diversification [9]): It lies in reallocating the jobs to machines with the purpose of both long and short jobs are assigned to every machine, therefore improving the load balancing of resources.
    2) Using penalization of ETC matrix values: We use the frequencymemory to correct the corresponding ETC values of most frequent job-machine assignments.
    3) Freezing jobs:Jobs that have most frequently changed their assignments are frozen (we set tabu to all movements involving the job during the iteration). After the diversification, the tabu status of this job(s) is cancelled.
    4) Strong diversification:Also called the hard perturbation. In this we perform a large perturbation of the current solution by randomly changing the assignments of a sufficient number of jobs.

V. COMPUTATIONAL RESULT

Table 1[12]

Parameter	MACO	TABU
MAKESPAN TIME	5755264	5761063
NET WATITING TIME	0.30394590	0.16991811
MACHINE USAGE	16hou rs	18 hours
RESPONSE TIME	68025.85	66020.96
WEIGHT USAGE	54.38	54.28
RUN TIME	5.56	11.15

VI. CONCLUSION

In this paper we study about the grid computing and its task scheduling algorithms like Ant Colony Optimization, Multiple Ant Colony Optimization, and Tabu Search Algorithm. In this we study about all these algorithms and comparison between them. In my research we find the MACO perform better than tabu search [14]. So that the result of MACO has less makespan time and less run time.

REFERENCES

[1] S.Umarani1, L.M.Nithya, A.Shanmugam,Efficient Multiple Ant Colony Algorithm for Job Scheduling In Grid Environment,S. Umarani et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2).

[2] Ku Ruhana Ku-Mahamud andHusna Jamal Abdul Nasir, “Ant Colony Algorithm for Job Scheduling in Grid Computing,2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation.

[3] Ting-ting Zhao, Ting Liu,Analysis of the Hybrid Scheduling Algorithm in the Gird Environment,Proceedings of 2012 International Conference on Modelling, Identification and Control, Wuhan, China, June 24-26, 2012.

[4] Youchan Zhu, Qiujuan Wei,An Improved Ant Colony Algorithm for Independent Tasks Scheduling of Grid,Volumn 22010 IEEE.

[5] JamshidBagherzadeh*, MojtabaMadadyarAdeh,An Improved Ant Algorithm for Grid Scheduling Problem, Proceedings of the 14th International CSI Computer Conference (CSICC'09)2009 IEEE

[6] D.Maruthanayagam,Dr. R.Uma Rani,Enhanced ant colony system based on rasa algorithm in grid scheduling,Vol. 2 (4) , 2011, 1659-1674, International Journal of Computer Science and Information Technologies.

[7] YizhiWang, Yuanxiang Ma, Grid Task Scheduling Based on Chaotic Ant Colony Optimization Algorithm, 2012 2nd International Conference on Computer Science and Network Technology.

[8] Bing Tang, Yingying Yin, Quan Liu and Zude Zhou, Research on the Application of Ant Colony Algorithm in Grid Resource Scheduling,

[9] MohdKamirYusof and MuhamadAzaharStapa, Achieving of Tabu Search Algorithm for Scheduling Technique in Grid Computing Using GridSim Simulation Tool: Multiple Jobs on Limited Resource, Vol. 3, No. 4, December, 2010, International Journal of Grid and Distributed Computing.

[10] FatosXhafa, Javier Carretero, Bernab´eDorronsoro andEnrique Alba, A tabusearch algorithm for scheduling independent jobs in computational grids, Computing and Informatics, Vol. 28, 2009, 1001–1014, V 2009-Mar-2.

[11] AysanRasooli, Mohammad Mirza-Aghatabar, SiavashKhorsandi, Introduction of Novel Dispatching Rules for Grid Scheduling Algorithms,May 13-15, 2008 Kuala Lumpur,

Malaysia, Proceedings of the International Conference on Computer and Communication Engineering 2008.

[12] SonalNagariya, Mahendra Mishra, Manish Shrivastava, valume 87- no.6, February 2014, International Journal of Computer Application(0975- 8887).

[13] Majid YOUSEFIKHOSHBAKHT, Mohammad SEDIGHPOUR, a combination of sweep algorithm and elite ant colony optimization for solving the multiple traveling salesman problem, Volume 13, Number 4/2012, pp. 295–301, proceedings of the romanian academy, series A.

[14] Paola Pellegrini, Daniela Favaretto, Elena Moretti, Multiple Ant Colony Optimization for a Rich Vehicle Routing Problem: a Case Study.

[15] FatosXhafa, Joanna Kołodziej, Leonard Barolli, Akli Fundo, A GA+TS Hybrid Algorithm for Independent Batch Scheduling in Computational Grids, 2011 International Conference on Network-Based Information Systems.

[16] Bajeh, A. O., Abolarinwa, K. O., Optimization: A Comparative Study of Genetic and Tabu Search Algorithms, *International Journal of Computer Applications (0975 – 8887) Volume 31– No.5, October 2011*

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

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

[19] Alba, E.—Almeida, F.—Blesa, M.—Cotta, C.—D´ıaz, M.—Dorta, I.—Gabarr´o, J.—Le´on, C.—Luque, G.—Petit, J.—Rodr´ıguez, C.—Rojas, A.—Xhafa, F.: Efficient parallel LAN/WAN algorithms for optimization.TheMallba project. Parallel Computing, Vol. 32, 2006, No. 5–6, pp. 415–440.

[20] Ali, S.—Siegel, H.J.—Maheswaran, M.—Hensgen, D.—Ali, S.: RepresentingTask and Machine Heterogeneities for Heterogeneous Computing Systems. Tamkang Journal of Science and Engineering, Vol. 3, 2000, No. 3, pp. 195–207.

[21] H¨ubscher, R.—Glover, F.: Applying Tabu Search With Influential Diversificationto Multiprocessor Scheduling. Comput. Oper. Res., Vol. 21, 1994, No. 8,pp. 877–884.

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

[23] 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

[24] R.Thamilselvan and Dr.P.Balasubramanie, Integrating Genetic Algorithm, Tabu Search Approach for Job Shop Scheduling, Vol. 2, No. 1, 2009, International Journal of Computer Science and Information Security.

[25] Maruthanayagam,Dr. R.Uma Rani,Enhanced ant colony system based on rasa algorithm in grid scheduling,CA:UniversityScience,1989.