DOI: 10.21917/ijsc.2010.0015

OPTIMIZATION OF GRID RESOURCE SCHEDULING USING PARTICLE SWARM OPTIMIZATION ALGORITHM

S. Selvakrishnan[1] and V. Perumal[2]
Department of Computer Science and Engineering, Saveetha Engineering College, India
E-mail: selvakrishnansekar@gmail.com[1], v.perumal23@rediffmail.com[2]

Abstract
Job allocation process is one of the big issues in grid environment and it is one of the research areas in Grid Computing. Hence a new area of research is developed to design optimal methods. It focuses on new heuristic techniques that provide an optimal or near optimal solution for large grids. By learning grid resource scheduling and PSO (Particle Swarm Optimization) algorithm, this proposed scheduler allocates an application to a host from a pool of available hosts and applications by selecting the best match. PSO-based algorithm is more effective in grid resources scheduling with the favor of reducing the executing time and completing time.

Keywords:
Grid Computing, Grid Resource Scheduling, PSO

1. INTRODUCTION

Resource sharing is the essence character of grid. There are a wide range of heterogeneous and geographically distributed resources in grid. For example, there are single processor, multiprocessors, shared memory machine, distributed memory machines, workstations, SMP, etc., and they are with different capabilities and configurations, and are managed in multiple administrative domains with different policies. In a word, the resource management and scheduling in such a complex environment are confronted with a great challenge [1][2].

Grid resource scheduling has become a popular issue in grid research field. There are many research papers about resource scheduling algorithm reported in recent years, and different algorithms have their advantages and disadvantages, and there is not a perfect scheduling algorithm.

This proposed work mainly focuses on a PSO-based grid resource scheduling algorithm. The rest of the work is organized as follows: Section II gives a detailed literature review and the grid resource scheduling process and Section III describes the ETC (Estimated Time to Complete) matrix generation algorithm. Section IV reviews the PSO algorithm, and then the proposed scheduling strategy is described, which is followed by simulation experiments in Section V. The last section concludes the whole work.

2. LITERATURE REVIEW

The resource scheduling in grid is a NP complete problem. Various algorithms have been designed to schedule the jobs in computational gird.

2.1 ANT ALGORITHM

The ant algorithm is also based upon heuristic approach. It is based on the behavior of real ants. Each ant deposes the chemical pheromone on its path when it searches for food from its nest. When each ant moves in a particular direction, the strength of chemical pheromone increases. With this, other ants could also trail along. This inspired the discovery of ACO algorithm. This algorithm uses a colony of artificial ants that behave as cooperative agents in a mathematical space where they are allowed to search and reinforce pathways (solutions) in order to find the optimal ones(i.e) the shortest path. This approach which is population based has been successfully applied to many NP-hard optimization problems [3][4].

2.2 GRID RESOURCE SCHEDULING PROCESS

When grid system starts, the resources register themselves in GIS (Grid Information System). Users submit tasks to the grid system and the Resource Broker (RB) schedules the submitted tasks to resources. In Resource Scheduling Flow Diagram the task list, including task details, such as application description and user requirement, is submitted to the broker through Grid Interface.

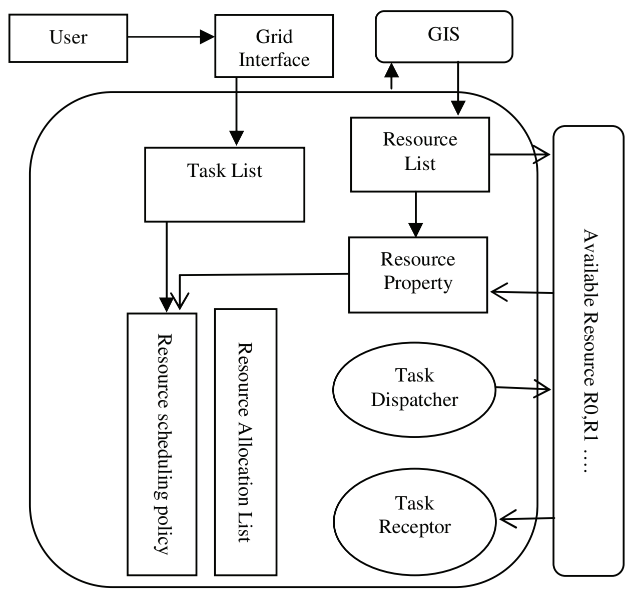

Fig.1. Resource Scheduling Flow Diagram

Once task lists are available, broker interacts with GIS to identify the contact information of resources and then interacts with resources to determine their dynamic information, including availability, number of processing units, processing rating, real-time load condition and configuration (which are defined as Resource Properties). According to user's requirement, depending on scheduling policy, broker schedules resources and creates a mapping relationship table which records where the tasks are assigned to, and the mapping table is stored in broker as a dispatching reference for Task Dispatcher[5]. The tasks are sent to available resources by the Task Dispatcher. The Task Receptor collects the tasks and holds the completed task and reschedules the incomplete tasks [6].

2.3 PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) is a population based search algorithm. The particles fly through a multidimensional search space in which the position of each particle is adjusted according to its own experience and the experience of its neighbors.

In the binary version of this algorithm was presented by Kennedy and Eberhart [10] ,In which, each particle is composed of D elements, which indicate a potential solution. In order to evaluate the appropriateness of solutions a fitness function is always used. Each particle is considered as a position in a D-dimensional space and each element of a particle position can take the binary value of 0 or 1 in which 1 means "included" and 0 means "not included".

Each element can change from 0 to 1 and vice versa. Also each particle has a D-dimensional velocity vector the elements of which are in range[$-V_{max}$,V_{max}] . Velocities are defined in terms of probabilities that a bit will be in one state or the other. At the beginning of the algorithm, a number of particles and their velocity vectors are generated randomly. Then in some iteration the algorithm aims at obtaining the optimal or near-optimal solutions based on its predefined fitness function. The velocity vector is updated in each time step using two best positions, *pbest* and *nbest* , and then the position of the particles is updated using velocity vectors. *Pbest* and *nbest* are D-dimensional, the elements of which are composed of 0 and 1 the same as particles position and operate as the memory of the algorithm. The personal best position, *pbest* , is the best position the particle has visited and *nbest* is the best position the particle and its neighbors have visited since the first time step. When all of the population size of the swarm is considered as the neighbor of a particle, *nbest* is called global best (star neighborhood topology) and if the smaller neighborhoods are defined for each particle (e.g. ring neighborhood topology), then *nbest* is called local [9].

2.4 ETC MATRIX GENERATION

The heterogeneity to be defined mathematically for generating the ETC matrices in any method, In the range based ETC generation technique, the heterogeneity of a set of execution time values is quantified by the range of the execution times [7][8].

The range based method for generating ETC matrices:

```
for  i= 0 to (t-1)
τ [i] = U( 1, R_task )
          for  j= 0 to (m-1)
                    e[i][j] =τ [i] × U( 1, R_mach )
          end for
end for
```

Where *e*[*i, j*] is the estimated expected execution time for the task *i* on the machine ,*L*et *U(a, b)*) be a number sampled from a uniform distribution with a range from *a* to *b*. (Each invocation of *U*(*a*; *b*) returns a new sample.) , '*m'* is the total number of machines in the Heterogeneous Computing (HC) suite, and *t* is the total number of tasks expected to be serviced by the HC system over a given interval of time. Let *Rtask* and *Rmach* be numbers representing task heterogeneity and machine heterogeneity, respectively, such that higher values for *Rtask* and *Rmach* represent higher heterogeneities. Then an ETC matrix *e*[*0..(t -1), 0..(m-1)*], for a given task heterogeneity and a given machine heterogeneity, can be generated by the range-based method.

Table.1. Suggested values for R_{task} and R_{mach} for a realistic HC system for high heterogeneity and low heterogeneity

	High	**low**
Task	10^5	10^1
Machine	10^2	10^1

3. GRID RESOURCE SCHEDULING USING PSO

Algorithm Description:

Step 1: Create and Initialize the *m x n* matrix with particle and position randomly.

Where

'm' denotes a number of node

'n' denotes a number of job

Step 2: Calculate the ETC values of each node using Range Based ETC Matrix

Step 3: Calculate the Fitness of each Particle using fitness function and Find out the Pbest and Gbest[9][10].

$$\text{fitness}=\left(\lambda.\text{makespan}+(1-\lambda).\text{mean}_{\text{flowtime}}\right)^{-1},\ \lambda\in[0,1] \quad (1)$$

(Where $X_k^{(t+1)}$ denotes the updated position of the particle), Where λ is used to regulate effectiveness of parameter.

Step 4: X_k is estimated and in case its value is greater than the fitness value of $pbest_k$, $pbest_k$ is replaced with X_k.

Step 5: Fitness value of *pbest* is greater than *nbest*, then *nbest* is replaced with *pbest*.

Step 6: Until to reach the max Velocity

if it is Satisfying the maximum Velocity

Stop

if it is not Satisfying the maximum Velocity

Update the Position Matrix as,

$$X_k^{(t+1)}=\begin{cases}1 \text{ if } \left(V_k^{(t+1)}(i,j)=\max\left\{V_k^{(t+1)}(i,j)\right\}\right), \forall i\in\{1,2,\ldots\ldots m\} \\ 0 \text{ otherwise}\end{cases} \quad (2)$$

Then update the velocity matrix as,

$$V_k^{(t+1)}(i,j)=w.V_k^t(i,j)+c_1r_1\left(pbest_k^t(i,j)-X_k^t(i,j)\right)+c_2r_2(gbest_k^t(i,j)-X_k^t(i,j)) \quad (3)$$

DOI: 10.21917/ijsc.2010.0015

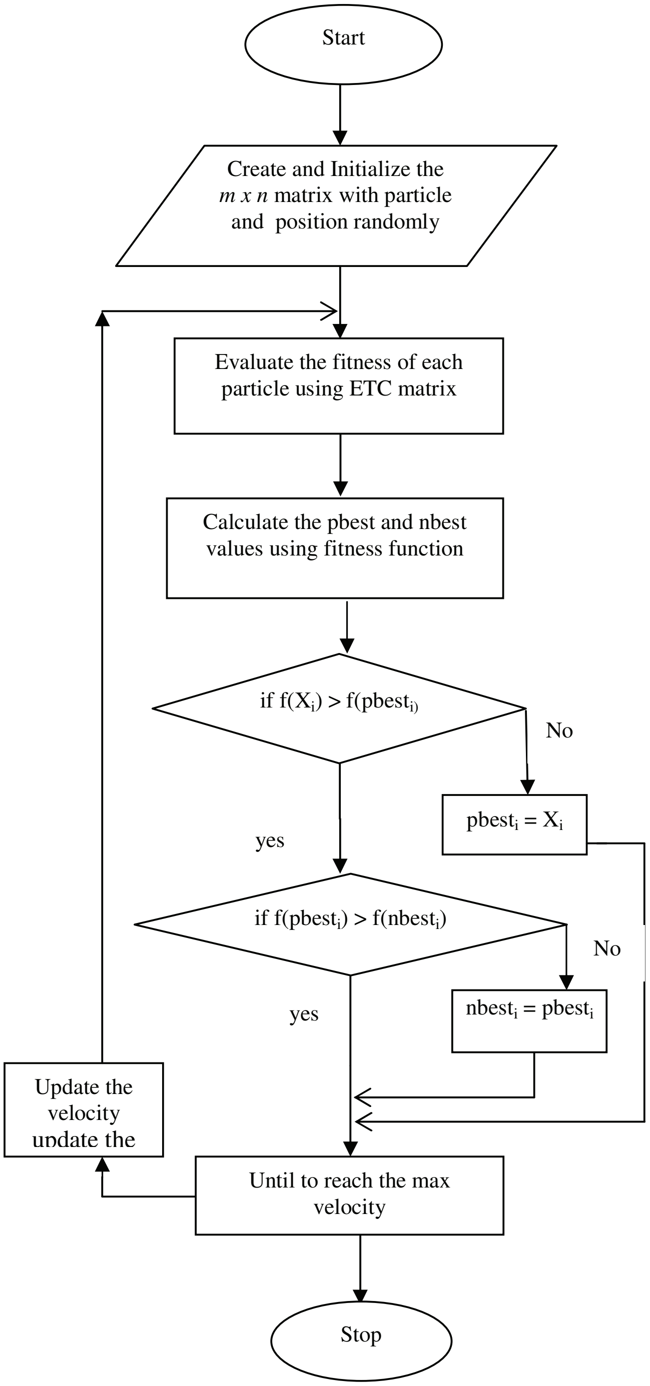

Fig.2. Flow Chart

Where

$V_k^{(t)}(i,j)$ its refer initial velocity

w - Weight inertia factor

c_1 and c_2 are positive constant.

r_1 and r_2 are uniformly distributed random number in [0,1].

pbest – personal best position.

gbest – global best position.

w = wMax-[(wMax-wMin) x iter]/maxIter

Where

wMax= initial weight,

wMin = final weight,

maxIter = maximum iteration number,

iter = current iteration number.

4. SIMULATION RESULTS

In simulation experiment, the resource scheduling based on PSO proposed in this work is implemented in JAVA, The experiment was conducted to obtained the status of Grid with different number of Jobs, Node, Services as input to obtain makespan and flow time (refer Table .2) . Then this obtained result was compared with the scheduling policy based on resource property which was proposed by Bing Tang, Yingying Yin, Quan Liu and Zude Zhou [5].

The results from the literature Survey listed below and the corresponding results produced from the same data by applying proposed algorithm are also listed below [We record the total execution time (the sum of time utilized that every task processed by PE) and the completing time (time utilized that the system needs to complete all the tasks)]. The comparative results are given in Table. 2 which show that the total Flow Time and Completion Time have been reduced using PSO, and Fig. 3 and Fig. 4 show that the completing time decreases dramatically using PSO.

Table.2. Comparison of statistical results between our proposed method and ant colony algorithm

Iteration	**Grid status: Number of (Jobs, Nodes, Services)**	**Number of iterations: (10*m)**	**Ant colony algorithm**		**Proposed PSO**	
			Makespan (μsec)	**Flow time (μsec)**	**Makespan (μsec)**	**Flow time (μsec)**
1	(9,6,8)	60	42121	5448	40409	3390
2	13,4,8)	40	28845	3890	28584	3640
3	(12,5,9)	50	32898	4056	32549	3267
4	(14,4,8)	40	31982	4898	30766	3920
5	(13,5,9)	50	35767	4789	34868	3499
6	(8,6,7)	60	30500	4008	30126	2530.5

DOI: 10.21917/ijsc.2010.0015

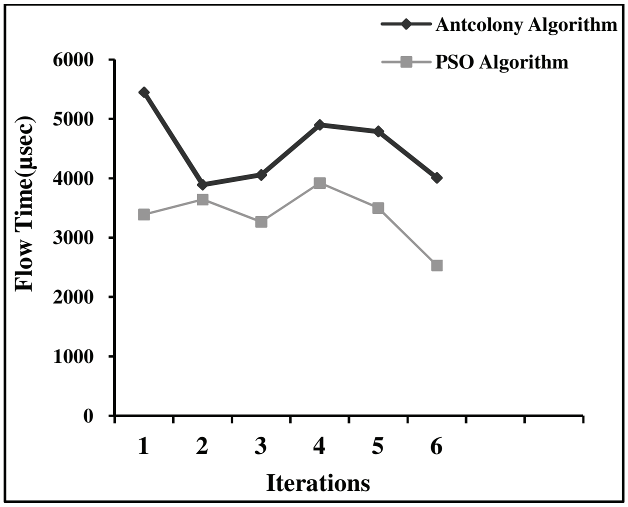

Fig.3. Flow Time

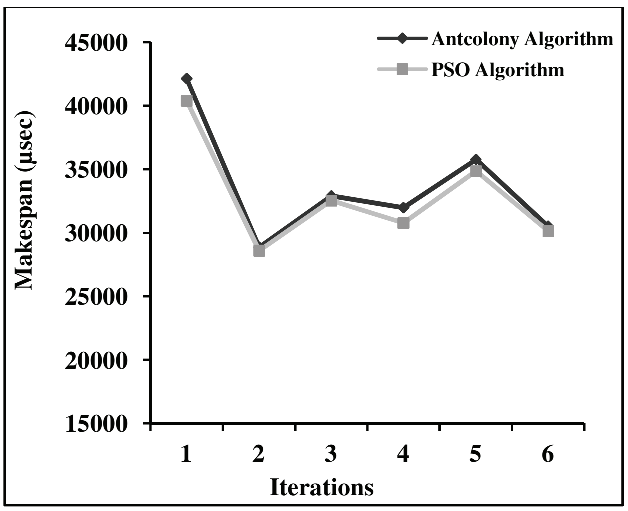

Fig.4. Completion Time

5. CONCLUSION

From the results derived, the PSO is a stochastic optimization technique. Stochastic means random and the outcome result can be changed each time the program starts. This optimization technique will optimize the path of particle and can absolutely reduce the execution time and completion time in Grid resource scheduling.

Now-a-days, many of the researchers need to solve their optimization problems. A lot of applications have been applying various optimization techniques so far. The efficiency of Particle Swarm Optimization (PSO) has been proved. Hereafter this PSO technique can be considered as the best option for their optimization based problems.

REFERENCES

[1] Foster, and C. Kesselman. 2003,"The Grid 2: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, USA.
[2] R. Buyya, D. Abramson, and S. Venugopal. 2005, "The Grid Economy". Proceedings of the IEEE, pp. 698-714.
[3] Kousalya.K, and Balasubramanie.P. 2008, "An Enhanced Ant Algorithm for Grid Scheduling Problem", IJCSNS International Journal of Computer Science and Network Security, Vol..8, No.4, pp.262 – 271.
[4] Rajkumar Buyya, Manzur Murshed. 2002, "GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing", Concurrency and Computation: Practice and Experience, Vol.14, No. 13-15, pp.1175–1220.
[5] Bing Tang, Yingying Yin, Quan Liu and Zude Zhou. 2008, "Research on the Application of Ant Colony Algorithm in Grid Resource Scheduling, Scholl of Information Technology", Wuhan University of Technology, Wuhan, 430070, China.
[6] Abraham. A, Buyya.R and Nath. B. December 2000, "Nature's Heuristics for Scheduling Jobs on Computational Grids", The 8th IEEE International Conference on Adv. Comp. and Communication, Cochin, India, pp. 45-52.
[7] ShoukatAli, HowardJaySiegel, Muthucumaru Maheswaran, Debra Hensgen, and Sahra Ali. "Task Execution Time Modeling for Heterogeneous Computing Systems", OS Research and Evaluation, CA 94043 USA.
[8] M. Kafil and I. Ahmad July-Sep. 1998, "Optimal task assignment in heterogeneous distributed computing systems," IEEE Concurrency, Vol.6, No.3, pp.42–51.
[9] HesamIzakian, BehrouzTorkLanani, Kamran Zamanifar, Ajith. "A Novel Particle Swarm Optimization for Grid Job Scheduling", Islamic Azad University, Ramsar branch, Ramsar, Iran.
[10] J. Kennedy, R.C. Eberhart. 1995, "Particle swarm optimization", in Proceedings of the IEEE International Conference on Neural Networks, Vol.4, pp.1942–1948.