Engineering Sciences International Research Journal
ISSN (Print) : 2330 – 4338 ; Volume 1 Issue 1 ǁ February. 2013 ǁ

An Effective Approach for Grid Task Scheduling Using Continuous PSO

Priya Jain[1], Amit Sinhal[2]

Abstract: Grid computing is a technology used to harness computing powers from various sources for solving problems which contains large number of computing cycles and use them in harmony to achieve a specific goal. This is also used to achieve high performance in computing environment to access large amount of data. It increases the efficiency by reducing the time significantly. It's various application are job scheduling, resource management, information management etc. Typical applications like weather forecasting, protein folding and earthquake simulation are prime candidates for a grid infrastructure. In the field of grid computing tasks scheduling is a big challenge. Each day new algorithms are proposed for assigning tasks to the resources. In this paper we use the technique of Particle Swarm Optimization (PSO) to solve the task scheduling problem in grid computing. The aim of using this technique is to use the given resources optimally and assign the task to the resources efficiently.

Keywords: ABC, GA, Grid scheduling, PSO.

1. INTRODUCTION

In the current scenario due to the growth of internet and availability of powerful computers with high speed network along with low cost components, is pushing the researchers and engineers to think about new technology, information and its services. With this technology geographically distributed resources can be accessed. With the era of the above "Grid Computing" is the emerging technology Grid computing is the combination of different resources collected from different administrative domains. It is used to solve the problems related to business, research and technical which require large number of processing cycles.

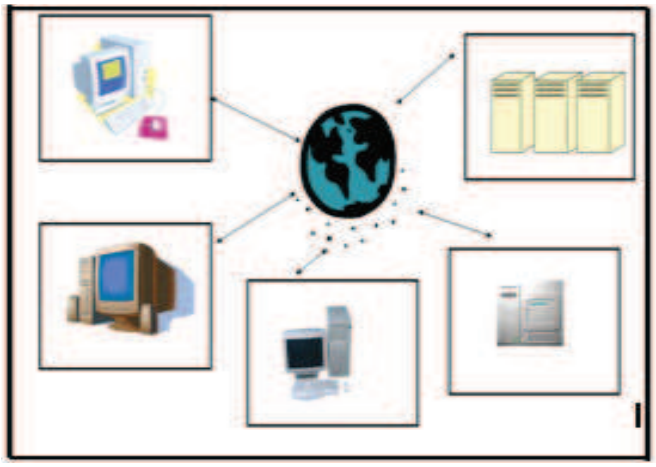

Fig: 1 Architecture of Grid Computing

Area in which Grid computing is applicable are as follows On-demand Computing, Distributed Computing, Data Intensive computing, Data Mining, High throughput Computing, Collaborative Computing, etc. Challenges faced during implementing Grid are: Data Movement, Data Replication, Resource Management and Job Submission. In this paper various task scheduling algorithm has been compared for effective resource management in grid computing.

2. TASK SCHEDULING ISSUES IN GRID COMPUTING

Complex computational problems can be solved by, computational grid. It consists of various software and hardware components as well as various resources to solve the given problem. Such as a set of printers and scanners which are used for printing and scanning a set a document. The task scheduling problem is intended to minimize the computation time and to utilize all the resources effectively [2, 7].

The task scheduling problem arises a hold up in a situation when available resources are less than the number of tasks. Consider a scenario wherein there are p, p={1,2,3,4,.....P} tasks to be done and there are q, q={1,2,3,4.....Q} resources available. With the condition that the task is not allowed to be migrated between resources.

When such a situation arises and if we have q < p then a new algorithms for task scheduling needs to be deployed because now inefficient resource allocation can greatly hamper the efficiency and throughput of the scheduler but if we have q > p then there is no reason for developing new algorithms for task scheduling because then resources can be allocated to the tasks on FCFS basis.

To formulate the problem, define Ta a={1,2,3,…P } as p independent tasks permutation and R_b b={1,2,3,…Q } as q computing resources. Suppose that the processing time $P_{a,b}$ for

task a computing on b resource is known. The completion time F(p) represents the total cost time of completion [7].

The aim is to find a permutation matrix m = (M_{ab}), with M_{ab} =1 if resource a performs task b and if otherwise, M_{ab}=0, which minimizes the total costs.

$$F(x) = \Sigma\Sigma P_{a,b} * M_{ab} \quad (1)$$

$$\Sigma Mab=1, \forall b \in T, \quad (2)$$

$$Mab \in \{0, 1\}, \forall a \in R, b \in T \quad (3)$$

The minimal F(p) represents the length of schedule whole tasks working on available resources. The scheduling constraints (2) guarantee that each task is assigned to exactly one resource.

3. ABC ALGORITHM

The Artificial Bee Colony (ABC) algorithm is a swarm based meta-heuristic algorithm used to simulate the intelligent foraging behavior of a honey bee swarm, the colony contains three agents of artificial bees: employed bees, onlookers and scouts The Employed Bee stays on a food resource and provides the neighborhood of the same in its memory. The number of the employed bees is equal to the number food resources , if food resource position cannot be improved further limit cycles it is assumed to be abandoned The Onlooker Bee will collect the information of food resources from the employed bees in the hive and select one of the food resource to gathers the nectar. The Scout is responsible for finding new food item or, the new nectar and resources. The nectar amount is analyzed and if it is found the nectar of previous one is less than new resource in their memory, they forget the previous positions and memorize the new position.

4. GENETIC ALGORITHM

The GAs is random searching methods based on the evolution selection and the natural phenomena. These algorithms are started with a set of random solution called initial population. Each member of this population is called a chromosome. Each chromosome is a problem solution which consists of the string called genes.

A set of chromosomes in each iteration of GA is called a generation. The chromosomes are evaluated by their fitness functions. The offspring is created by applying some operators on the current generation. These operators are crossover which selects two chromosomes of the current population, combines them and generates a new offspring, and mutation which changes randomly some gene values of a chromosome and creates a new offspring. Then, the best children and maybe their parents are selected by evolutionary select operator according to their fitness value(s).

5. PARTICLE SWARM OPTIMIZATION

PSO is a stochastic global optimization algorithm based on simulation of social behavior in which exchange of information takes place instead of mutation between individuals called particles, of the population known as swarm. The fundamental concept of PSO consists of accelerating each particle toward its pbest and the gbest , in this algorithm particle keeps track of its coordinates in the solution space which are associated with the best solution (fitness) that has achieved so far by that particle. This value is called personal best, *pbest*. Instead of this one best value will be tracked by PSO is one that is obtained by any particle in its neighborhood This value is called *gbest*. PSO follows following steps, first of all it initialize particles with random position and velocity vector, then for each and every particle P fitness value need to be evaluated, then it will check if the fitness value (p) is better than fitness (pbest) in that case pbest = p. After that all the pbest value has been compared and in that the best one is assigned as gbest and at last the velocity and position of the particle will be updated

The i^{th} particle of the swarm is represented by an n-dimensional vector, $P_i = (p_{i1}, p_{i2}, p_{i3} \ldots\ldots px_{in})^T$ for n-dimensional search space, The velocity of this particle is represented by another n-dimensional vector $V_i = (v_{i1}; v_{i2} \ldots\ldots v_{in})$ *T*. The previously best visited position of the i^{th} particle is denoted as $P_i = (p_{i1}, p_{i2, pi3} \ldots pin)$ *T*. `*g*' represents the best particle index in the swarm. The velocity and position of thee i^{th} particle is updated using the velocity update equation and the position updated equation respectively.

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - Pid) + c_2 r_2 (p_{gd} - Pid) \text{ -----} \quad (4)$$

$$p_{id} = p_{id} + v_{id} \text{ ----------} \quad (5)$$

where $d = 1, 2....n$; $i = 1; 2....S$, where *S* represents swarm size; c_1 and c_2 are constants, called cognitive and social scaling parameters respectively (usually, $c_1 = c_2$; r_1 , r_2 are random numbers, uniformly distributed in [0, 1]. Equations (4) and (5) are the initial version of PSO algorithm. V_{max}, is used to limit the velocities of the particles arbitrarily and enhance the resolution. For better controlling of inertia weight exploration and exploitation has been done. Enhancement in the above development encourages purging the need for V_{max}. The resulting velocity update equation becomes:

$$v_{id} = w * v_{id} + c_1 r_1 (p_{id} - Pid) + c_2 r_2 (p_{gd} - Pid) \text{ ---------} \quad (6)$$

Eberhart and Shi (2000) [14] indicate that the optimal strategy is to initially set *w* to 0.9 and reduce it linearly to 0.4,

allowing initial exploration followed by acceleration toward an improved global optimum.

6. PROPOSED METHODOLOGY

PSO has been seen rapid development and improvement, with lots of successful applications to real-world problems. Attempts have been made to improve the PSO performance and a few PSO variants have been proposed to improve the performance of existing algorithm. Much work focused on parameters settings of the algorithm and on combining various techniques for the same. However, most of these improved PSOs manipulate the control parameters or hybrid operators without considering the varying states of evolution. Hence, these operations lack a systematic treatment of evolutionary state and still sometimes suffer from deficiency in dealing with complex problems.

In this paper we have proposed a solution for grid scheduling using continuous version of PSO. For solving any optimization problem we have to first formulate the problem accordingly. To solve the problem, representation of the individual and fitness value is required. PSO algorithm is based on population and each population have its own fitness value according to which it is compared from others, so we have to first represent the grid scheduling problem in terms of PSO. In this we have various task and resources to achieve the optimized result for effective utilization of task with available resources. PSO is based on population concept and each individual in population represents a solution, in case of grid scheduling problem, solution is a sequence of tasks which are to be performed. So we have to first formulate each individual of PSO.

Table1: Values of position vector, sequence and set of resources

Dimension	P_{id}	S_{id}	R_{lm}
0	4.8	4	4
1	2.55	6	1
2	4.90	1	2
3	7.46	5	0
4	0.05	9	4
5	-2.67	3	0
6	-0.28	8	3
7	6.72	0	0
8	2.50	7	2
9	1.28	2	2

Grid task scheduling is a discrete optimization problem. For implementing discrete version in grid task scheduling there is a requirement of new vector to be generated from individual of a PSO. To generate new discrete vector value index value is noted according to the dimension values of a particular individual in ascending order such as shortest dimension value is noted first then second shortest dimension value and so on up to the last dimension value. The above generated discrete vector represents the sequence of grid task. From the set of sequences, we represent dimension as a number of task and value as an initial sequence for finding optimal sequence. Position vector $Pid=\{p_1, p2, p3,.....p_d\}$ where i is the particular individual and d represents the dimension index, is calculated using PSO. The position vector transforms the partial from its position.. The position vector Pid has continuous values. A new discrete vector $S_{id}= [s_{i1}, s_{i2},....s_{id}]$ will get generated with the help of dimension values of Pid. S_{id} represents task sequence of i partial to the d dimension in the processing order.

Set of resources is represented by $R_{lm}= \{r_1,r_2,r_3.....r_y\}$, where l represents a particle/ sequence and m represents the tasks which are assigned to a resource. After S_{id}, set of resources is calculated using equation (10).

$R_{lm}=S_{id}$ mod M i.e.

value of task set mod Total resources (10)

Each resource will contain a resource id for their unique identification. Such as r_1 is the resource id of first resource for e.g. if we have 12 tasks which are to be performed on 6 available resources then we have 12 different dimension value based on the continuous position values. For the respective continuous values new discrete vector is generated S_{id}, which is a sequence of task simplified by the particle Pid. Position vector Pid is calculated using PSO = {4.8, 2.55, 4.90, 7.46, 0.05, -2.67, -0.28, 6.72, 2.50, 1.28}

Then new vector S_{id} is generated by using the index of the dimension value of position vector Pid, we have S_{id} value as {4, 6, 1, 5, 9, 3, 8, 0, 7, 2}. The shortest value of position vector Pid is -1.55 and the index value of this value is 1 so first value for sequence vector S_{id} is 1, the second shortest value of position vector Pid is -0.28 and the index value of this value is 8 so first value for sequence vector S_{id} is 4 and so on all the values calculated. Equation (10) is then used to determine the associated resources for the calculated tasks in the sequence. We can calculate the resource set as {4, 1, 2, 0, 4, 0, 3, 0, 2, 2} The table1 represents the dimension values of position vector Pid, sequence S_{id} and Set of resource $R_{lm..}$

After representation of each individual we have to calculate fitness value of each individual. On the basis of fitness value we determine the optimal solution. In case of grid scheduling problem optimal solution is the minimization the value of equation (2).

ISBN: 978-93-81583-90-6

Our main objective is to minimize the fitness value, an individual who have the minimum fitness value is considered as the optimal solution.

7. ALGORITHM:GRID SCHEDULING USING PSO

To solve the grid scheduling problem we have used the Particle Swarm Optimization (PSO). We set an initial population by selecting random starting sequences from the set of x! Sequences; where x is the total number of tasks. After getting the initial particle we calculate fitness value of each particle, according to equation (2). After that we calculate best among the entire particle and set it as an initial global best.

PSO update equation is used to update old population and generate new sequences and then their resources are calculated. This sequence, along with its resource is then used to find the fitness value of each individual of each particle of the population. Algorithm 1 is the proposed algorithm for the grid scheduling problem.

7.I Algorithm 1: Algorithm for Grid Scheduling Using PSO

In Initialization phase:

Step 1: Repeat steps 1 to 9 where s=0 to Swarm size

Step2: Repeat steps 2 to 6 where d=0 to dimension size

Step3: Initialize particles randomly

Step 4: New task sequence vector S_{id} is generated by the particle position index values

Step5: Compute resource for that particle/sequence S_{id}

Step6: End of the d loop

Step7: Calculate the fitness value of initialized particle

Step8: Calculate the global best position

Step9: End of the s loop

In Update Phase:

Step10: Repeat

Step11: Repeat steps 11 to 19 where s=0 to each swarm size

Step12: Repeat steps 12 to 16 where d=0 to problem dimension

Step13: Using PSO update equation, update particles

Step14: New task sequence vector S_{id} is generated by the particle position index values

Step15: Calculate the resources for that sequence S_{id}

Step16: End of the d loop

Step17: Calculate the fitness value of updated particle

Step18: if needed update historical information for global best (Pg)

Step19: End of the s loop

Step20: Until stopping criteria not meet.

8. EXPERIMENTAL RESULTS

For every algorithm there are some control parameters which are used for its efficient working. Hence, there are some controls parameters for PSO also. We did an extensive literature survey and carried out our own experiments for determining the values of these control parameters. From this we found that the values which we have taken in this experiment are standard values and they are also suitable for this experiment.

The first control Parameter is Maximum function evaluation and the value of this parameter we have taken in our experiment as 20,000. The next parameter in our experiment is maximum number of population and we have taken its value to be 40. Another control parameter is number of runs and we have taken its value in our experiment as 30. It must be noted that each run contains maximum function evaluation, which is 20,000 in our experiment. The fourth control parameter is Dimension and it depends upon the number of tasks. The next control parameter is the value of c1 & c2 which we have taken as 1.14. And w is also a control parameter and we have taken its value as 0.7.

To test the efficiency of our algorithm results of PSO is compared with ABC and GA algorithm results. The information regarding the number of tasks, number of resources, and amount of time is known in prior for task completion. We just need to find the sequence which will provide us the optimal results. We conducted the experiment by varying the number of resources as well as varying the number of tasks and then we compared our results with that of ABC and GA algorithm. In particular, we have taken three cases in which we have taken different number of resources and tasks.

Experiment 1: In this experiment we have considered 5 resources and 17 tasks for comparing ABC, GA and PSO. Following are the execution time (in units) taken by the above algorithm respectively.

Table2: Execution time calculated by ABC and PSO for17 tasks by 5 resources

ABC algorithm	Genetic Algorithm	Proposed methodology(PSO)
7174.0	4078.0	3029.0

The sequence generated by ABC is: 14, 8, 0, 16, 4, 13, 6, 1, 3, 9, 11, 15, 12, 10, 5, 7, 2.

The sequence generated by GA is: 15, 7, 0, 16, 2, 13, 5, 1, 3, 9, 11, 8, 12, 10, 4, 14, 6.

The sequence generated by our proposed PSO with is: 0, 3, 9, 5, 13, 2, 12, 6, 1, 4, 16, 7, 8, 10, 11, 14, and 15.

Fig:1 Graph representing comparison of various algorithm for 5 resources and 17 task.

Experiment 2: In this experiment we have considered 3 resources and 28 tasks for comparing ABC, GA and PSO. Following are the execution time (in units) taken by the above algorithm respectively.

Table3: Execution time calculated by ABC and PSO for 28 tasks by 3resources.

ABC Algorithm	Genetic Algorithm	Proposed methodology(PSO)
7145.0	7078.0	7004.0

The sequence generated by ABC is:24,9,7,2,26,0,27,8,13,3, 18, 10, 1, 23, 5, 17, 14, 4, 15, 12, 6, 16, 20, 11, 22, 25, 19, 21.

The sequence generated by GA is:19,8,7,2,26,0,27,9,13,3, 18, 10, 1, 23, 14, 17, 5, 4, 15, 11, 6, 16, 12, 20, 22, 25, 24, 21.

The sequence generated by our proposed PSO is: 0, 5, 15, 9, 22, 4, 19, 8, 2, 3, 26, 6, 1, 14, 16, 7, 27, 12, 10, 21, 17, 23, 25, 18, 13, 24, 11, 20.

Experiment 3: In this experiment we have considered 12 resources and 30 tasks for comparing ABC, GA and PSO. Following are the execution time (in units) taken by the above algorithm respectively.

Table4: Execution time calculated by ABC and PSO for 30 tasks by 12 resources

ABC Algorithm	Genetic Algorithm	Proposed methodology(PSO)
6136.0	6081.0	5921.0

The sequence generated by ABC is: 28, 14, 20, 25, 10, 7, 17, 4, 9, 22, 6, 11, 24, 18, 0, 29, 15, 1, 26, 12, 13, 19, 29, 5, 3, 27, 2, 21, 16, 8.

The sequence generated by GA is: 18, 12, 20, 10, 2, 7, 17, 4, 9, 3, 6, 14, 24, 28, 0, 29, 15, 1, 26, 11, 13, 19, 16, 5, 22, 27, 25, 21, 29, 8.

The sequence generated by our proposed PSO is: 0, 5, 17, 10, 24, 6, 21, 9, 2, 4, 29, 3, 1, 15, 18, 7, 13, 8, 23, 16, 26, 28, 19, 14, 25, 27, 11, 20, 12, and 22.

From the table 2, table 3and table 4, it is clear that PSO takes less execution time than ABC and Genetic algorithm. This can be seen in the below graph.

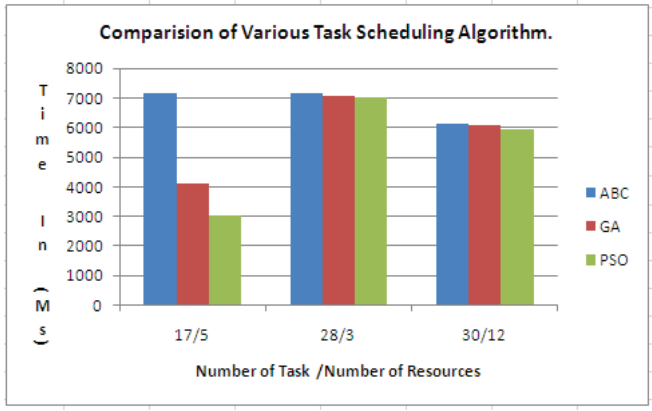

Fig. 2. Graph representing comparison of ABC, GA and PSO Algo.

9. CONCLUSION

In this work Particle swarm optimization (PSO) has been described and compared with Artificial Bee Colony algorithm and Genetic Algorithm for optimal task allocation with given resources. The simulation carried out concludes that proposed PSO performs better than the ABC and Genetic Algorithm. The procedure followed in grid scheduling consists of the generation of the population according to the algorithm, then the task sequence and resources associated with the task sequence are generated and the positions, task sequence and resource set are updated and then the globally best position or sequence is calculated. It is repeated again and again till the maximum number of function evaluation. As future work we have the intention to apply other types of nature inspired algorithms to the grid scheduling problem, comparing their results with the ones accomplished by the PSO.

10. REFERENCES

[1] Foster and C. Kesselman (editors), *The Grid: Blueprint for a Future Computing Infrastructure*, Morgan Kaufman Publishers, USA, 1999.

[2] Abraham, R. Buyya and B. Nath, *Nature's Heuristics for Scheduling Jobs on Computational Grids*, The 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000), pp. 45-52, December 2000.

[3] M. Clerc and J. Kennedy, "*The particle swarm explosion, stability, and convergence in a multidimensional complex space*", *IEEE Trans. on Evol. Computation*, 6(1), pp. 58–73, 2002.

[4] *Future Generation Computer Systems* Elsevier, pp.151- 161, 2005.

[5] M. Aggarwal, R.D. Kent and A. Ngom, *Genetic Algorithm BasedScheduler for Computational Grids,* in Proc. of the 19th Annual International Symposium on High Performance Computing Systems and Application (HPCS'05),pp.209-215, May 2005.

[6] S. Song, Y. Kwok, and K. Hwang, *Security-Driven Heuristics and A Fast Genetic Algorithm for Trusted Grid Job Scheduling,* in Proc. of 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), pp.65-74, April 2005.

[7] Lee Wang, Howard Jay Siegel, Vwani P. Roychowdhury, and Anthony A. Maciejewski, *Task Matching and Scheduling in Heterogeneous Computing Environments Using a Genetic Algorithm Based Approach*, Journal of Parallel and Distributed Computing , *Citeseer*47, pp.8-22, 1997

[8] Zhang, L.; Chen, Y.; Sun, R.; Jing, S. & Yang, B. *A task scheduling algorithm based on PSO for grid computing* International Journal of Computational Intelligence Research, 4, 37-43, 2008.

[9] Nath B, Lim S, Bignall R J, *A Genetic Algorithm For Scheduling Independent Jobs On Uniform Machines With Multiple Objectives,* Proceedings, International Conference on Computational Intelligence and Multimedia Applications, Australia, pp. 67-74, 1998.

[10] J. Kennedy and R. Eberhart, "*Particle swarm optimization*", in Proc. IEEE International Conference Neural Networks, vol. 4, pp. 1942 – 1948, 1995.

[11] Lin JianNing and Wu HuiZhong, *Scheduling in Grid Computing Environment Based onGenetic Algorithm*, Journal of Computer Research and Development,pp.2195-2199,Vol. 4,No.12,Dec 2004.

[12] Tatsuya Nomura, *An Analysis on Linear Crossover for Real Number Chromosomes in an Infinite Population Size*, In Proc. ICEC'97, Indi-anapolis, pp. 111- 114, 1997.

[13] Y. Shi and R. C. Eberhart, "*A modified particle swarm optimizer*", in Proc. IEEE International Conference on Evolutionary Computation, Piscataway, pp. 69-73, 1998.

[14] R. C. Eberhart and Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization Congress on Evolutionary Computing, vol. 1, pp. 84-88, 2000.

* * *

[1]TIT Bhopal, India/ M.TECH (IT)/R.G.P.V./ priyainfo005@gmail.com
[2]TIT Bhopal, India/ Assistant Prof. /R.G.P.V./ amit_sinhal@rediffmail.com