International Journal of Computer Applications Technology and Research
Volume 3– Issue 8, 529 - 532, 2014, ISSN: 2319–8656

Resource Allocation in Computational Grids environment Using Improved Particle Swarm Optimization Algorithm

Mehdi Effatparvar
ECE Department, Ardabil Branch, Islamic Azad University, Ardabil, Iran

Shojaat Hoseinpour
Department of Computer Science and Research Branch Islamic Azad University Germi, Iran

Vahid Asadzadeh
Department of Computer Science and Research Branch Islamic Azad University Germi, Iran

Abstract: Resource allocation in computational grids is considered as a NP-Complete problem due to resources heterogeneity. Grid resources are related to various management areas exerting different management policies. Nowadays, enhancing grid efficiency is regarded as a problem requiring proper and effective schedule. Unfortunately, grid resources dynamic nature, in addition to the variety of users' requests has intensified grid resource allocation. The present paper offers a new heuristic method based on Particle Swarm Optimization (PSO) algorithm for resource allocation in grid environment. The proposed method creates an optimal scheduling in task completion with minimum flowtime and makespane.

Keywords: scheduling, computational grid, heuristic algorithm, resource allocation, optimization algorithm, particle swarm

1. INTRODUCTION

Grid computation make feasible sharing and highly-wide integrating of the distributed resources including super computers, data storage systems, and data resources as well as special instruments accessible to organizations; in addition to finding possible solutions to complex problems in science, engineering, and commerce. The main idea of grid computations emerged when there were no resources within management of a scientific problem solution requiring large amount of calculations or data. Almost any grid system possesses particular software for task scheduling. When an applied grid program is presented by a user, the software designates suitable machine (or machines) over which the program is to be performed. In the simplest case, this task is completely done blindly with a Round Robin algorithm. For instance, totally 4 machines were assumed while the requests of performing various applied programs are gradually issued by different users, through separating applied programs into tasks i.e. the first task is dedicated to the first machine, the second to the second, the third to the third, and so on. It must be mentioned that even in case of using a Round Robin algorithm; task performance is not so simple [5].To put it more precisely, performing the next task request goes to the first following machine essentially is capable of performing the task based on resources (hardware and software).The major problem in each grid is optimal and secure resource management based on resource owners' predetermined accessibility policies to resources. Once a user transfers an application to grid server, it is initially required to have a complete list of grid all accessible computational resources in order to know which ones based on what policies can be applied [1]. Researches show heuristic optimization methods', inspired from nature, higher efficiency compared to other methods. Most of these methods tried to minimize makespan tasks. Swarm Intelligence is a type of Artificial Intelligence Method based on swarm behaviors. There are many heuristic swarm intelligence algorithms including Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Firefly Algorithm (FA) proposed for optimization. Of these, PSO algorithm has been proven to be the best heuristic method; the efficiency of which can be improved through combining to other methods since properties such as high velocity convergence, error tolerance, flexibility, and insensitivity to initial values.

2. LITERATURE

Min-Min scheduling function is a rather acceptable efficiency heuristic method. It is initiated with a group of unallocated tasks in 2 steps: In the first step, a series of tasks is calculated with minimum completion time. In the second, the task with the minimum completion tome is selected and allocated to the resources. Then, the allocated tasks are removed from the list of unallocated tasks and the same procedure is repeated for the other unallocated tasks [2] [3]. The function of Max-Min scheduling is similar to Min-Min method consisting 2 steps. In the first step, a set of tasks is computed with the minimum time of completion. In the second step, the task with the maximum completion time is allocated to the resources. In most cases, Max-Min efficiency and load balancing are better than those of Min-Min in the grid resources [2] [4]. Genetic algorithm is an evolutionary technique to search a wide area General procedure of genetic algorithm search is as follows: First, creating the initial population including a set of chromosomes indicating a possible solution. This solution is mapped between the tasks and grid resources. The next step assesses chromosome where a value is attributed to each. This value represents the delay time of chromosome tasks. The purpose of genetic searching is to find chromosome optimal values. The third step incorporates combination and mutation operations. Combination is a process in which particular chromosomes sequences are removed. Mutation is a process changing particular chromosomes' sequences through using multiple task mappings (new for the present population). This process is repeated until the criteria of the end of the process are satisfied [8]. The algorithms are regarded as the most popular natural heuristic methods in optimization as their simplicity. Simulated Annealing Algorithm is a local search method to find the general optimal solution of a complicated problem. The primary idea of this method was posed in 1953 [7]. In this approach, an object is first heated up to high temperature; then, cooled gradually down so that the system will be maintained at a thermodynamic balance at any time. If the temperature sufficiently decreases, the object reaches a thermal balance called an optimal condition. If, compared with grid system, thermal balance is, indeed, mapping between tasks and grid resources considered as the goal of optimization. Temperature is, then, the sum of one mapping completion times. Cost function and temperature changes cause mapping to be changed. If the next temperature is higher, means that it has not been a good mapping; so, the next case will be most probably accepted since accepting unsuitable conditions provides avoiding local optimality often occurs in local optimum. The system initial temperature is in fact makespan initial mapping implemented as follows: The first mapping is created through a uniform random distribution while the new makespan is evaluated. If the new makespan is greater, then a uniform random number in the range of z ϵ [0, 1] will be selected; z and y are obtained by the relation (1).

(1) $$y = \frac{1}{1+e^{(\frac{old\ makespan - new\ makespan}{temperture})}}$$

If z>y, then the new mapping is maintained; otherwise rejected and the previous mapping will be maintained, too. Therefore, system temperature is cooled down, while most of unsuitable solutions would be hardly maintained. This may be more reasonable as the probability of finding a better solution than an unsuitable solution goes less when the temperature declines more. Following any changes, system temperature is reduced to 90% of its current level (cooling rate); and one stage of algorithm retrieve is completed. The algorithm will be stopped when there is no more changes in makespan as some iterations or system temperature get close to 0.

3. Resource allocation on grid

Resource Allocation is independent tasks incorporates N tasks and M machines. Each task must be processed by each M machine so that scheduling would ultimately reach its minimum duration. The proposed algorithm considered service quality parameters, makespan, flowtime, and task performance cost, respectively. Each task can only be implemented on one of the resources continuing until the performance is completed. The proposed algorithm utilized ETC matrix model [1]. Since the scheduling algorithm is static, it can be assumed that the expected makespan over each resource of i for each task of j is predetermined inserted in matrix ETC [i, j] [9].

Completion_Time [i, j] is equal to task j completion time in resource i calculated as follows:

(2)Completion_Time[i,j]= ETC[i,j]

Makespan is the maximum completion time of Completion_Time [i, j] computed by the following equation:

(3)Makespan= Max (Completion_Time[i,j]) $1 \le j \le N$, $1 \le i \le M$

Flowtime is sum of tasks completion of Completion_Time [i, j] in all resources calculated by the following equation:

(4) $$flowtime = \sum_{i=1}^{m} \sum completion\_Time[i,j]$$

Scheduling of the proposed algorithm is focused on reducing makespan, flowtime, and task completion cost in sending tasks to resources.

4. The proposed Recource allocation algorithm

PSO algorithm, first stated by Kennedy and Russell Eberhart in 1995, is inspired by birds and other animal's migrations. In fact, more experienced bird flies ahead at migration to find food. In other words, it can be stated that particles or birds cooperate to find food. In recent years, this algorithm has been widely used in solving different problems including optimization. Moreover, Hass assessment demonstrated the power of this algorithm in solving such problems. The particle Xi possesses a position vector, velocity vector, and its own fitness amount. In each algorithm retrieval, positions and velocity's amounts change through (5) and (6), respectively:

(5) $v_{id}(t+1)= wv_{id}(t)+c_1r_1(pBest_{id}- x_{id}(t))+ c_2r_2(gBest_d-x_{id}(t))$

(6) $x_{id}(t+1)=x_{id}(t)+v_{id}(t+1))$

In above equation, w is the inertia weighted factor,   is the best previous position of the particle,   , the best previous positions of all previous steps particles, is the velocity of ith particle in repetition of t,   is position of the ith particle in iterations of t, r1 and r2 are two random numbers, and C1 and C2 are two constant coefficients.

4.1 Particle representation

Of the main issues in applying PSO algorithm to solve resource allocation problem is how to turn a scheduling problem into a solution or actually how to create mapping between problem solutions and particles in the PSO algorithm. Within scheduling PSO algorithm, each particle is considered as a possible solution to resource allocation in such a way that each particle vector has a length of N, where N is the total number of input tasks. Each element inside the particle vector is a random integer between 1 and M (total number of resources). For instance, the 2nd particle of task 2, T2, performed on the resource 3, R3is illustrated in Fig 2-2.

Recourses/Tasks	T1	T2	T3	T4
Particle1	R1	R2	R3	R2
Particle 2	R1	R1	R3	R3
Particle 3	R1	R3	R1	R2

Figure 1. Displaying typical particles

4.2 Generating initial population

In a standard copy, initial population of PSO algorithm is randomly created so that a random number between 1 and M, representing the resource number on which the favorable task is performed, is generated. In the proposed method, part of initial population (schedulers) is created by Max-Min method. This leads to intelligently creation of initial population as well as improving their qualities and characteristics in order to reach an optimal or near-to-optimal answer as soon as possible. Part of initial population is randomly created in the suggested method to keep the population diversity; hence, there would be no more in advance algorithm convergence.

4.3 Creating initial velocity vector

A new notion of velocity updating in turbulent PSO algorithm has been introduced based on the minimum velocity constraints [6]. One of the main reasons of PSO algorithm premature convergence is the particles static condition and lack of global search in the problem area. In this model, a procedure is introduced to move low-mobility particles while they are allowed to discover better solutions. If the particles' velocity is reduced (under the threshold) a new velocity will be attributed using following equation. Thus, the turbulent PSO algorithm is provided by the following new velocity equation:

In the above equation, is a random number uniformly selected within the range of [0, 1] and is the scale index to control particular fluctuation ranges based on . is the minimum velocity threshold. is the velocity achieved by the equations (7).

$$\hat{v} = \begin{cases} v_{ij} & \text{if } |v_{ij}| \geq v_c \\ u(-1,1)v_{max/\rho} & \text{if } |v_{ij}| < v_c \end{cases} \quad (7)$$

4.4 Calculating fitness function in proposed algorithm

The main goal of resource allocation with help of PSO algorithm is to minimize makespan and flowtime. The particle having this feature is more suitable for proposed algorithm.

$$(8)\quad x_i(t+1) = x_i(t) + \alpha(rand - 1/2)$$

4.5 Termination conditions

To finish of swarm Intelligence algorithms such as pso, it must be mentioned the termination conditions. This algorithm will be terminated after reaching maximum iteration.

5. Evaluation the proposed algorithm

The proposed algorithm together with other scheduling algorithms were tested condition table 2-5, where all the models were considered identical to properly evaluate task length. Parameter of the limits of task length is indicative of the limits of uniform distribution of task length. The numbers of iterations show totally 20 retrievals have been implemented using the present algorithms to achieve the program makespan Time; and then, the amounts averages were evaluated.

Table 1. Recourses allocation values

Number of Resource	Number of Task	Typical
64	351	YAR-64-110
64	340	YAR-64-100
64	293	YAR-64-90

The proposed algorithm in condition of table 1 was compared with standard PSO algorithms, simulated annealing algorithms, and genetic algorithms. Before assessing results, it was necessary to determine the initial values of the parameters used in the algorithms. These values are shown in table 2.

Table 2. Initial values of the parameters of scheduling algorithms

Algorithm	Parameter	Value
TPSO	Population size	40
	Self-consciousness study factor C_1	1.49
	Swarm consciousness study factor C_2	1.49
	Inertia factor	0.9
GA	Size of the population	20
	Probability of crossover	0.8
	Probability of mutation	0.02
	Scale for mutations	0.1
SA	Number operations before temperature adjustment	20
	Number of cycles	10
	Temperature reduction factor	0.85
	Vector for control step of length adjustment	2
	Initial temperature	50

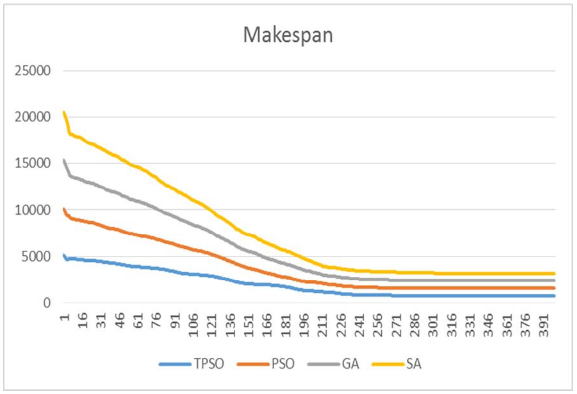

Figure 2. Diagram of makespan

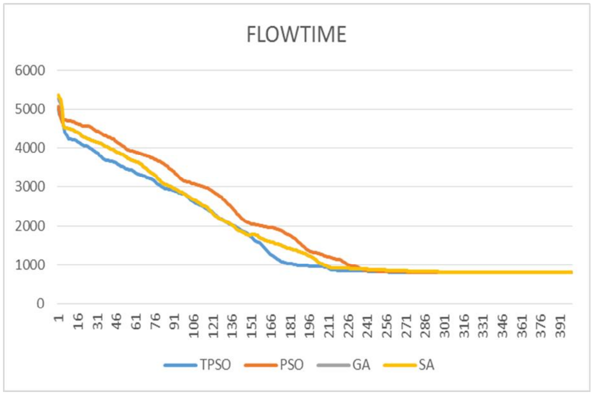

Figure 3. Diagram of flowtime

6. Conclusion

The computational grids provide reliable available to other computational resources. These resources are as Heterogeneous and distributed and are used shared. In additional, resources in grid are belonged to various organizations that have specific management policy and used for different users at different times.. In this complicated media management it cannot be used traditional methods for resources management that try to optimize the efficiency rate at the system level. In this paper, proposed method was presented for scheduling jobs in computational grid. In the proposed method combination of the Turbulent Pso was used.. For this purpose we used multi purposes function. Simultaneously two parameters makespan and flowtime evaluate service quality and minimum sum of the three mentioned parameters. In the proposed method, we have improved the mentioned parameters of service quality such as time of jobs implementation. Mentioned parameters are simulated carefully. The results show the superiority of the proposed method than the compared methods.

8. REFERENCES

[1] Yuanyuan, Z.,Wei, S., Yasushi, I. (2008). “Predict task running time in grid environments based on CPU load predictions”, Future Generation Computer Systems 24 (6), pp. 489–497.

[2] Braun, T.D., et al. (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. 61(6), p.p 810-837.

[3] Chauhan, S.S. and R. Joshi. (2010). "A weighted mean time min-min max-min selective scheduling strategy for independent tasks on grid" , Advance Computing Conference (IACC), 2010 IEEE 2nd International Patiala, pp. 4-9

[4] He, X.S., Sun, X.H. and G. Von L. (2003). "QoS guided min-min heuristic for grid task scheduling" , Journal of Computer Science and Technology, 18(4): p.p 442-451..

[5] Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., Teller E., Chem J., 1953, “Equation of State Calculations by Fast Computing Machines”, 21, 1087-1092.

[6] Liu H., Abraham A., Zhang W.,(2005). ” A fuzzy adaptive turbulent particle swarm Optimization”, Vol 1, No. 1.

[7] Fidanova, S., (2006). ” Simulated Annealing for Grid Scheduling Problem”, Modern Computing,. JVA '06. IEEE John Vincent Atanasoff 2006 International Symposium on Sofia. p.p 41-45.

[8] Yang, G., Hongqiang, R., Joshua Z.H., (2005), “Adaptive grid job scheduling with genetic algorithms”, Future Generation Computer Systems, p.p 151-161.

[9] Sajjad, A. Ch., Seyed Naser, R., Ali H.,(2014), “Job Scheduling on the Grid Environment using Max-Min Firefly Algorithm”, International Journal of Computer Applications Technology and Research, Vol3. Issue 1, p.p 63-67