International Journal of Computer Applications Technology and Research
Volume 3– Issue 1, 63 - 67, 2014, ISSN: 2319–8656

Job Scheduling on the Grid Environment using Max-Min Firefly Algorithm

Sajjad Asadzadeh Chalack
Department of Computer Science and Research Branch
Islamic Azad University
Ardabil, Iran

Seyed Naser Razavi
Computer Engineering Department, Faculty of Electrical and Computer Engineering, University of Tabriz, Iran

Ali Harounabadi
Member of Science Board of Computer Group in Azad Islamic university of Tehran Center, Iran

Abstract: Grid computing indeed is the next generation of distributed systems and its goals is creating a powerful virtual, great, and autonomous computer that is created using countless Heterogeneous resource with the purpose of sharing resources. Scheduling is one of the main steps to exploit the capabilities of emerging computing systems such as the grid. Scheduling of the jobs in computational grids due to Heterogeneous resources is known as an NP-Complete problem. Grid resources belong to different management domains and each applies different management policies. Since the nature of the grid is Heterogeneous and dynamic, techniques used in traditional systems cannot be applied to grid scheduling, therefore new methods must be found. This paper proposes a new algorithm which combines the firefly algorithm with the Max-Min algorithm for scheduling of jobs on the grid. The firefly algorithm is a new technique based on the swarm behavior that is inspired by social behavior of fireflies in nature. Fireflies move in the search space of problem to find the optimal or near-optimal solutions. Minimization of the makespan and flowtime of completing jobs simultaneously are the goals of this paper. Experiments and simulation results show that the proposed method has a better efficiency than other compared algorithms.

Keywords: Scheduling, Grid computing, Firefly algorithm, Max-Min algorithm, Makespan, Flowtime.

1. INTRODUCTION

Grid computing enables the sharing of a wide integrating of distributed resources including supercomputers, data storage systems, data resources, and as well as special tools that are available to certain organizations and provides the ability to solve complex problems in science, engineering and in commerce. The main motivation of the grid computing was created when the available resources in a range of management was not available to resolve a scientific problem which requires huge computations or data. There is an ability on grid computing that Influences human life surprisingly such as the effect of electricity grids, and be the next revolution after the Internet and World Wide Web revolutions. Grid is composed of a set of virtual machines that each has its varied resources and services and provides access to resources based on its special policy. Hence, grid's resources and services are very different and distributed in different geographical areas. Grid resources are recorded within one or more of information service. Users submit their requests to the resource broker and grid resources broker discoveries appropriate resources to apply this request by searching in grid resources, and then schedules them on the discovered resources. Computational grid is a software and hardware infrastructure that provides reliable, stable, comprehensive and cheap access to other local resources [4]. Computational grid is a shared environment that is implemented by establishment of lasting and standard service. These services support the creation and sharing of distributed resources. Nowadays increasing grid efficiency is a problem. To increase the efficiency of the grid, a properly and efficiency scheduling is needed. Unfortunately, dynamic nature of grid resources as well as different demands of users caused the complexity of grid scheduling problem. Dynamic of resource efficiency is due to Heterogeneous, autonomy and being shared of grid resources [7]. The goal of grid scheduling problem is the optimal assignment of jobs to resources.

Research has shown that heuristic optimization methods inspired from nature have more impact and efficient than other methods. Most of these methods try to minimize the maximum execution time of jobs. Swarm Intelligence is a kind of artificial intelligence methods based on swarm behavior. Many swarm Intelligence algorithms are proposed for optimization such as: Ant colony, Particle swarm and firefly algorithms. Among them, firefly algorithm is the best Heuristics method because of features such as high speed convergence, being insensitive to the initial values, flexibility and having the high error tolerability. On the other hand it has the local search, falls into the trap of local optimality and acts weakly in global search. It is proven that this algorithm efficiency is improved by combining other methods.

2. LITERATURE

Scheduling approach of Min-Min algorithm is a heuristic method that has a relatively reasonable efficiency and starts with a group of unallocated jobs that consists of two stages. In the first stage, a set of jobs are computed by minimum time. In the second stage, job is selected with the minimum completion time and be allocated to resources. Then allocated job is removed from unallocated jobs and this process is repeated for other unallocated jobs [5][2]. Scheduling approach of Max-Min algorithm is similar to Min-Min method and consists of two stages. In the first stage, a set of jobs are computed by minimum time. In the second stage, job is allocated to resources with the maximum completion. In most cases, efficiency and Max-Min load balancing is better than Min-Min in grid resources [5][3]. PSO is parallel search algorithms based on population that starts with a set of random answers (Particles), then PSO continues searching in problem space to find optimal answer by updating particles positions. Each particle is specified as multi-dimensional (depending on the type of problem). An important problem in the use of particle swarm optimization algorithm for solving optimization problems is how to create a mapping between problem and particles vector. The problem dimension in this algorithm is the number of jobs considered. So length of each particle and the velocity vector is considered as the number of jobs. Each particle as x_{id} has a position vector, a velocity vector and its own fitness value[6]. During each algorithm iteration, values of positions and velocity are changed by the flowing equations.

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

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

In the above equation, w is the inertia weight factor, $pBest_{id}$ is the best previous position of particle, gBestd is the best previous position of all particles, v_{id} is the velocity of ith particle at iteration t, x_{id} is the position of particle ith at iteration t, r_1, r_2 are random numbers, and c_1, c_2 are constants [5].

3. SCHEDULING JOBS PROBLEM ON GRID

Scheduling problem of independent jobs, include N jobs and M machines. Each jobs should be processed somehow by each of M machines, to minimized the total length of schedule at last. In the proposed algorithm the quality of service parameters, makespan and flowtime of jobs are considered respectively. Each job can be run only on one source and does not stop until the end of the run. In the proposed algorithm the ETC matrix model is used that is explained in [3]. Since the scheduling algorithm proposed is the static, it is assumed that expected execution time for each job j on each resource i, has already been determined and is located within the matrix ETC [i, j].

Completion_Time [i,j] is equal time that job j be completed on resource i and is computed as follows.

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

Makespan: maximum completion_Time [i,j], that is computed as follows.

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

Flowtime: sum of the completion time of jobs [i, j] over all resources, that is computed by follow equation.

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

The goal of scheduling in the proposed algorithm is to submit each of the jobs to each of the resources to minimize makespan and flowtime of the jobs at last.

4. THE PROPOSED SCHEDULING ALGORITHM

In the proposed scheduling algorithm, hybridization of firefly with Max-Min algorithm in the scheduling problem solving of independent jobs in the computational grid is used. Before presenting the algorithm, it is necessary to examine what parameters are needed to solve the scheduling problems using the firefly algorithm[8].

4.1 Fireflies representation

Search procedure in the firefly algorithm is each firefly compares with others, if firefly light is less than compared firefly, moves towards firefly with more lights (problem of finding maximum point), this act causes particles focuses around a particle which has more lights, and in the next iteration of the algorithm, if a particle with more lights exist, particles move towards it again. Search stages must be iterated in more numbers. If the population size is increased in the wide problem space such as grid system, time complexity of algorithm rises and algorithm efficiency falls strongly. In such algorithms, the solution of this problem in generating initial population. This means that we should apply solutions to reduce initial population size as far as possible. For this goal we should generate more qualitative initial population until the number of comparisons reduces. So we can reduce the number of iterations of algorithms and achieve to optimal or near-optimal solution in the less time. We have used the Max-Min algorithm to generate part of the initial population. One of the features of this algorithm is that maintains load balancing and establishes it at the same first stage. One of the most important points in the use of firefly algorithm in scheduling problem solving of independent jobs is that how to convert a scheduling problem to a solution, or indeed how can create a mapping between solutions and fireflies in firefly algorithm. In scheduling firefly algorithm, each firefly is a solution for allocation of tasks, so the length of each firefly vector is N, which N is the total number of input tasks. Each element inside the firefly vector is a random number between 1 to M (M is the total number of resources).

	T1	**T2**	**T3**	**T4**	**T5**
Firefly1	R2	R5	R3	R2	R1
Firefly2	R1	R4	R5	R3	R2
Firefly3	R1	R3	R5	R4	R2

Figure 1. Typical Fireflies

4.2 Generation of the initial population of fireflies

In the proposed method, a part of the initial population individuals is generated by Max-Min algorithm and some randomly. With this method the generated population has a good qualification and the other part of the population such that a random number between 1 to M indicating the number of resource is generated until the job specified is executed on it. Randomness helps to maintain population variety and the selection chance to be given to individuals of the population. The length of each of fireflies would be equivalent to the number of jobs. The size of fireflies indicates the number of candidate solutions or the value of searching in the problem space.

4.2.1 The method of generating primary population by Max-Min algorithm

Max-Min scheduling approach is a heuristic method. At first a set of jobs with minimum completion time is computed and the job with maximum completion time is selected and allocated to the resources.

For example, matrix ETC in table 2 is indicated for 6 jobs over 4 resources.

Table 1. Matrix ETC for 6 jobs over 4 resources

Task\Resource	R_0	R_1	R_2	R_3
T_0	200	250	220	300
T_1	150	170	190	160
T_2	300	320	180	360

T_3	400	380	350	310
T_4	100	120	140	160
T_5	220	250	280	200

According to the Max-Min method, a set of jobs with minimum completion time of each matrix row is computed and job with maximum completion time is selected and allocated to resources. The result of this job is as follows.

[200, 150, 180, 310, 100, 220]

[310]

Namely the task of T_3 is allocated to R_3 resources. This job is done for other unallocated tasks with regard to executed time of R_3 resources. Finally the diagram of tasks allocation to resources is in the form of figure (2).

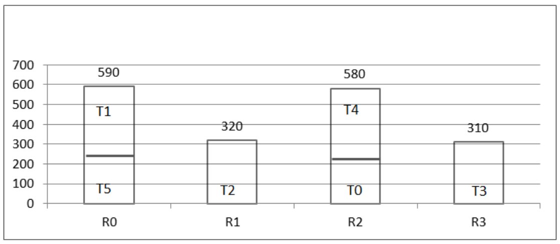

Figure 2. Tasks allocation by Max-Min algorithm

4.3 Evaluation of the fireflies

The next stage is measurement of fireflies light that depends on the considered problem. For this goal we have used the multi objective fitness function that includes two parameters of QOS (flowtime, makespan) for evaluation of fireflies.

4.3.1 The first fitness function

One of the parameters of evaluating the goodness of schedules in this problem is measurement of makespan that can be computed using the following equation.

(4) Min(Max(Completion_Time[i,j]))

4.3.2 The second fitness function

Sum of the completion time of all jobs should be minimum flowtime that can be computed by the following equation.

(5) $\text{Min}(\sum_{i=1}^{m} \sum \text{completion}_{\text{timei}}, [i,j])$

4.4 Distance

Distance between any two firefly i and j such as x_i and x_j can be defined by Cartesian distance R_{ij} using the below equation respectively, such that $x_{i,k}$ is the k'th section of the spatial coordinates x_i of firefly i, and d is the number of dimensions of the problem.

(6) $r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}$

4.5 The attractiveness

Computation of the attractiveness function for a firefly is shown in the following equation, where r is the distance between each pair of fireflies, β_0 is the first attractiveness (r=0), and γ is the absorption coefficient that controls light intensity.

$$\beta_0 = \beta_0 \, e^{-\gamma r i,j^2} \quad with \ m \geq 1$$

(7)

4.6 The movement

The firefly I is attracted by firefly j, that is brighter, according to the equation (7), where x_i is the current position or solution of a firefly, $\beta_0 e^{-\gamma r_{i,j}^2 (x_i - x_j)}$ is the rate absorption of the firefly by adjacent fireflies. *α(rand-1/2)* is the random movement rate of a firefly, α coefficient is a random parameter with the problem interest, α∈[0-1]whereas rand of the obtained numbers is determined from the uniform distribution in the space.

(8) $x_i(t+1) = x_i(t) + \beta_0 \, e^{-\gamma r i,j^2}(x_i - x_j) + \alpha(rand - 1/2),$

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

4.7 Termination conditions

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

5. PSEUDO CODE OF THE PROPOSED ALGORITHM

Max-Min Firefly Algorithm

1. Begin

2. Finding some part of the solutions by Max-Min

3. Initialize population by the result of Step 2

4. Objective function f(x), x = (x1,…, xd)T

5. Generate initial population of fireflies xi (i = 1, 2,…, n)

6. Light intensity Ii at xi is determined by f(xi)

7. Define light absorption coefficient γ

8. while (t <MaxGeneration)

a. for i = 1 : n all n fireflies

b. for j = 1 : n all n fireflies (inner loop)

c. if (I_i < I_j), Move firefly I towards j; end if

c.1. Vary attractiveness with distance r via exp[-γr]

c.2. Evaluate new solutions and update light intensity

end for j

end for i

*9. Rank the fireflies and find the current global best g**

10. End while

6. SIMULATION

In this section, efficiency of the presented algorithm in the previous section that was carrying out scheduling jobs in a computational grid will be evaluated. This algorithm tries to carry out scheduling act of a number of independent jobs in a grid media. This jobs are belong to an application that user is delivered it to grid for run. The user accompany with program specifies the quality of considered service, namely time optimization strategy for system. By selecting time optimization strategy, user can ask from grid system to run its applied program in the least possible time. It can be shown by simulation that how is the efficiency of time optimization algorithm in compared together. All of the experiments is done on a system with dual core processor of 2.40 MHz, 3 GB memory and windows 7. Simulation is done by Matlab R2010a and all of the algorithms are simulated in this environment. The proposed algorithm with several other scheduling algorithms is examined according to conditions of table 2 and for suitable consider of jobs length is considered equal in all models experiment. The parameter of the range of jobs length indicates the range of uniform distribution of jobs length. The numbers of jobs parameter indicates that for obtaining runtime of program using the available algorithms, 400 repeats is done and then the mean values are selected for consideration. Figure (3), shows that in the two proposed methods, makespan is minimized than other compared algorithms. Figure (4), shows that in the two proposed methods, flowtime is minimized than other compared algorithms.

Table 2. Primary values of scheduling algorithms parameters

Algorithm	Parameter	Value
Firefly	Population size	40
	light absorption coefficient(γ)	1
	the randomization parameter(α)	0.2
	maximum attractiveness value(β_0)	2.0
Particle swarm optimization	Population size	40
	Self-consciousness study factor C_1	1.49
	Swarm consciousness study factor C_2	1.49
	Inertia factor	0.9

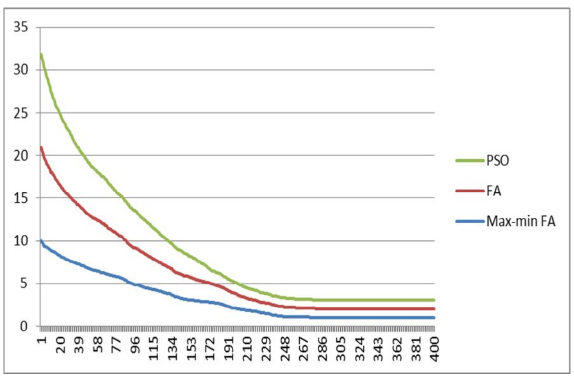

Figure 3. Diagram of makespan

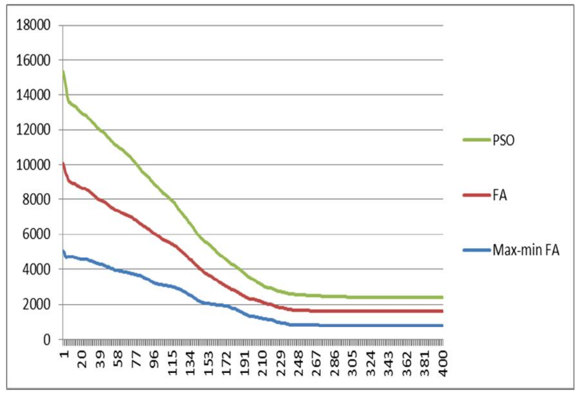

Figure 4. Diagram of flowtime

7. CONCLUSION

The computational grids provide reliable and cheap available to other computational resources. These resources are as Heterogeneous and distributed and are used shared. On the other hand, resources in grid are belonged to various organizations that have specific management policy and used for different users at different times. Hence, owners and users of resources have different aims, strategies and supply and demand. 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 firefly, Max-Min was used. The most concentration was over two factors, first was generation of primary population done using Max-Min algorithm and cause to be better and thus the convergence algorithm speed rises and the optimum solution is reached sooner. The second factor was evaluation of the solutions problem. 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] 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.

[2] Casavant, T.L., Kuhl, J.G. 1988 A taxonomy of scheduling in general-purpose distributed computing systems. IEEE Transactions on Software Engineering.

[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) IEEE 2nd International Patiala.

[4] Foster, I., Kesselman, C., Nick, J. and Tuecke, S. 2002 The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Computer.

[5] 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.

[6] Hesam, I., Behrouz, T.L., Ajith, A., Vaclav, S. 2010 A DISCRETE PARTICLE SWARM OPTIMIZATION APPROACH FOR GRID JOB SCHEDULING. International Journal of Innovative Computing, Information and Control.

[7] Sarkar, V. 1989 Determining average program execution times and their variance. in: Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation.

[8] Yang, X. 2010 Nature-Inspired Metaheuristic Algoritm, University of Cambridge: Luniver Press.