International Journal of Computer Applications (0975 – 8887)
Volume 62– No.16, January 2013

Job Scheduling in Grid Computing with Cuckoo Optimization Algorithm

Maryam Rabiee
Department of Computer, Science and Research Branch, Islamic Azad University, Khouzestan, Iran

Hedieh Sajedi
College of Science,
University of Tehran, Tehran, Iran

ABSTRACT

Computational grid is a hardware and software infrastructure that provides dependable, inclusive and credible to other computing capabilities. Grid computing intercommunicated with a set of computational resources on a large scale. Scheduling independent jobs is an important issues in such areas as computational grid. Scheduling is the process of assigning jobs to resources in order to achieve different goals. The grid schedule, find the optimal resource allocation to it over heterogeneous resources and maximize overall system performance. As yet evolutionary methods such as Genetic, Simulated Annealing (SA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) to solve the problem in the grid schedule has been adopted. The disadvantage of these techniques premature convergence and trapping in local optimum in large-scale problems. In this paper, a method by Cuckoo Optimization Algorithm (COA) to solve job scheduling in grids computational design, implementation and results are presented. The results show our proposed schedule have more efficient and better performing compared with Genetic and Particle Swarm Optimization.

Keywords

Grid computing, job scheduling, Cuckoo Optimization, Genetic algorithm, Particle Swarm Optimization.

1. INTRODUCTION

At the time is named the communication, need of having the various information and decentralization on the one hand and doing large power computing and having scientific purposes, commercial, military, etc, On the other hand, our primary world of computer, center processing , move to, the modern distributed processing.

Today, large supercomputers doing an organization all computing jobs that can turn to a network of small computers. The companies that having this network can extend an office, a building, a city, or province or even the whole breadth of the continent. The purpose of sharing resources is networking equipment such as printers, scanners, processors and resources among branches. Among the most important resources to share, are processors. The purpose of share processors are raising throughput, faster calculations and solve complex problems.

Grid computing [1] as a new approach to solve large-scale problems in science, engineering and business has emerged. So far, many studies have been done in the area of job scheduling [2] in computational grids. Intelligent optimization method [3] for solving such complex problems are the perfect choice. Therefore, the aims in this paper is finding the best way to solve this complex problem is presented .

This paper is organized as follows: In Section 2, we addressed a history of applying evolutionary algorithms for job scheduling in the Grid. In Section 3, job scheduling in grid computing is presented. In Section 4, the Cuckoo Optimization Algorithm is represented. Problem definition is presented in section 5. Job Scheduling using Cuckoo Optimization Algorithm will be discussed in Section 6. Experimental results are presented in Section 7 and the conclusions are listed in Section 8.

2. RELATED WORK

We address some of researches that are relevant with this issue.

Aggarwal and kent [4] used a Genetic algorithm based scheduler for grid environment. A Directed Acyclic Graph (DAG) represented for each job, taking into account arbitrary precedence constraints and arbitrary processing time. The result showed that scheduler minimize makespan, idle time of the available computing resources. Wang and Duan [5] a new genetic simulated annealing (GSA) algorithm which combines genetic algorithm with simulated annealing algorithm for grid scheduling is proposed. it could avoid trapping in a local minimum effectively and get the global optimization at last. Fidanova and Durchova [6] introduce a jobs scheduling algorithm for grid computing. The algorithm is based on Ant Colony Optimization which is a Monte Carlo method. This algorithm guarantee good load balancing of the machines.

Carretero and Xhafa [7] present the implementation of Genetic Algorithms for job scheduling on computational grids that optimizes the makespan and the total flow time.

Zhang and chen [8] represented a heuristic approach based on Particle Swarm Algorithm is adopted to solving job scheduling problem in grid environment. Experiments showed that the PSO algorithm is able to be better schedule than Genetic algorithm. Mathyalagan and suriya [9] used ACO algorithm for grid scheduling by stigmeric communication. They modified pheromone updating role that solves the grid scheduling problem effectively than that existing ACO. Umale and Mahajan [10] proposes multilevel decision making engine, as special case Two Level Decision Algorithm (TLDA). In the first level decisions we generate initial schedule using ACO algorithm. We refine this schedule using GA.

3. JOB SCHEDULING IN GRID COMPUTING

Scheduling is one of the most important problems in computational systems such as grid. To increase performance, a grid schedule is needed to performance and efficiency. Unfortunately, the dynamic nature of the grid and also the various demands of users, due to the complexity of the problem is the grid schedule. This means that the efficiency of resources, dynamic nature grid is always changing. Reasons of Dynamics grid are listed below [11]:

- Heterogeneous and autonomous: resources in different organizations which are under the control of grid obey the certain domestic policies. These policies specify how resources are shared, who has access to what resource and under what conditions.
- Efficiency variable grid resources: performance of available resources is constantly changed. The change is derived of their autonomy and resource sharing between different users.
- Assign jobs to different demands of users: it is possible for a user time and another financial cost of the operation is important.

The aim of grid schedule is find the optimal resource and allocation to a job, overcoming heterogeneous resources and maximize overall system performance.

Grid scheduling is defined as the process of making scheduling decisions included resources over multiple administrative domains. This process can include searching multiple administrative domains to use a single machine or scheduling a single job to use multiple resources at a single site or multiple sites. We define a job to be anything that needs a resource from a bandwidth request, to an application, to a set of applications. We use the term resource to mean anything that can be scheduled: a machine, disk space, a QoS network, and so on.

4. CUCKOO OPTIMIZATION ALGORITHM

Cuckoo Optimization Algorithm (COA) is one of evolutionary techniques was introduced in 2009 by Yang and Deb [12]. This algorithm is inspired by the lifestyle of a bird called the Cuckoo. This lifestyle is one of the rarest brood parasite in nature. This bird didn't make nest for itself and it be used the nests of other birds for laying eggs. Ability to create eggs like the bird host is reinforced in cuckoo bird. If the bird's host discover eggs that are not mine, it throw away or leave the nest and it makes a nest in other places. Cuckoo eggs are the bigger size of the host bird until cuckoo brood would hatch soon. When the host bird's eggs throws out of the nest or demand food so much to other broods die of hungry. When the cuckoo brood grows and becomes a mature bird continues the mother's life instinctively.

4.1 Generating initial cuckoo habitat

In order to solve an optimization problem, it's necessary that the values of problem variables be formed as an array. In GA and PSO terminologies this array is called "Chromosome" and "Particle Position", respectively. But here in COA it is called "habitat" [13]. To start the optimization algorithm, a candidate habitat matrix is generated. Then some randomly produced number of eggs is supposed for each of these initial cuckoo habitats. In nature, each cuckoo lays between 5 to 20 eggs. These values are used as the upper and lower bounds of egg assigned to each cuckoo at different iterations. Other habit of real cuckoos is that they lay eggs within a maximum distance from their habitat. This maximum area will be called "Egg Laying Radius (ELR)". Each cuckoo has an egg laying radius (ELR) which is appropriate with the total number of eggs, number of current cuckoo's eggs and also variable limits of var_{hi} and var_{low} [13]. So ELR is defined as:

$$ELR = \propto \times \frac{\text{Number of current cucko's eggs}}{\text{Total number of eggs}} \times (Var_{hi} - Var_{low}) \quad (1)$$

Which α is an integer, supposed to handle the maximum value of ELR.

4.2 Immigration of cuckoos

When young cuckoos grow and become mature, they live in their own area and make society for some time. But when the season for egg laying approaches they move to new habitats with the most similar host eggs and with more food for new young birds. Then the cuckoo groups are formed in different areas, the society with the highest fitness value is selected as the goal point, and other cuckoo to move to that point.

When mature cuckoos live in that environment identify cuckoos belong to which groups that are difficult. Now that cuckoo groups are identified their mean benefit value is calculated. The maximum amount of the benefit is determined by the goal group and consequently that group's best habitat is the new destination habitat for moving cuckoos.

When moving toward goal point, the cuckoos do not fly all the way to the destination habitat. They only fly a part of the way and also have a deviation. Figure**1** shows Pseudo code from cuckoo optimization algorithm.

5. PROBLEM DEFINITION

In grid environment, one of the most important issues is how to represent a solution for job scheduling. In this paper, each job is composed of n operations will run on m machine. while a job sign in to the system may be does not need to the all machine or does not require a particular machine more than once, So scheduling may be faced with complex problems, based on the following rules and assumptions:

1. All jobs logged in the system must be use all the machines.
2. Sequence of operations on each machine may be different.
3. No limit exists on processing time on each machine [14].
4. All jobs enter the system at time zero, and the process can be started at the same time [15].
5. No exists expiration time for all jobs.
6. Machines may also be idle during runtime.
7. Each operation able to run just on one machine at any time.
8. Each machine is processed at most one operation at a time.
9. When the operation starts to run, the run is not interrupted, the resources are exclusive and job execution time is atomic.
10. There is not any machine crash, all the machines are available all the time [16].
11. All jobs must be completed.
12. A job must be performed in a specific order and pre-determined.

1. **1.** Initialize cuckoo habitats with random points
2. **2.** Define ELR for each cuckoo
3. **3.** Let cuckoo to lay eggs inside their corresponding ELR
4. **4.** Kill those eggs that are identified by host birds
5. **5.** Eggs hatch and chicks grow
6. **6.** Evaluate the habitat of each newly grown cuckoo
7. **7.** Limit cuckoos'maximum number in environment and kill those that live in worst habitats
8. **8.** Cuckoos find best group and select goal habitat
9. **9.** Let new cuckoo population move toward goal habitat
10. **10.** If stop condition is satisfied end, if not go to 2

Fig1: Pseudo code of Cuckoo Optimization Algorithm

6. JOB SCHEDULING USING CUCKOO OPTIMIZATION ALGORITHM

Based on the other evolutionary algorithms a cuckoo optimization algorithm for solving job scheduling in grid is constructed. Figure 2 shows the model of proposed algorithm, the cuckoo's egg corresponds to a solution of the problem.

Generate initial population is base of beginning algorithm and based on incoming items including the number of jobs, number of machines, the operations of each job and the time required for each operation, algorithm tries to find the optimal solution. We assume the number of machines with the number of operations are equal. We use operation-based presentation to show cuckoo's egg and nests, because in this type of presentation each permutation of cuckoo's eggs is a candidate solution.

A user enter a sequence of n*m. N is the number of jobs and m is the number of machines. While the order of the execution of jobs on machines is different, so we should select an order of machines with minimum execution time for each job. Whereas each operation of job is just implemented by one machine, so one operation can not be assigned to two machines simultaneously. To satisfy this assumption, we should define a penalty coefficient that applies to fitness function (FF). Fitness function is defined by the following formula:

$$FF = \sum_{j=1}^{n} \min(Execution\ times) + FL \qquad (2)$$

Which j is the number of jobs, FL is penalty coefficient. If two machines implement an operation of jobs simultaneously, penalty coefficient caused increase FF and will be disregarded by COA.

After calculating the execution time of jobs and find an optimal sequence of jobs on machines, turn is the allocation machines based on the input sequence. A sequence of operations that the user will be asked to determine which ones job, which the machine is delivered.

Since the machines may be idle during run time, idle time calculating machines is important. The objective is minimize the maximum execution time of machines. We defined a maximum and minimum values for idle time of machines.

To ensure the assumption that two machines not be allocate at a time, we define the other fitness function to prevent the occurrence of this event. Fitness function compute by the following formula:

$$FF = \max(finishing\ time\ of\ machines) + FL \qquad (3)$$

Which FL is penalty coefficient.

7. EXPERIMENTAL RESULTS

In order to test our proposed algorithm, we vary the number of jobs submitted to grid. Grid performance is shown in Table 1. In order to analyze the performance of job scheduling algorithm. we had an experiment with a small job scheduling problem. Figure 3 shows flow execution time algorithm with increase jobs in grid. Each experiment was repeated 10 times.

Our proposed method have been compared with existing algorithms such as Genetic [17] and Particle Swarm Optimization [18]. In order to compare the best execution time obtained Table 2 is very practical.

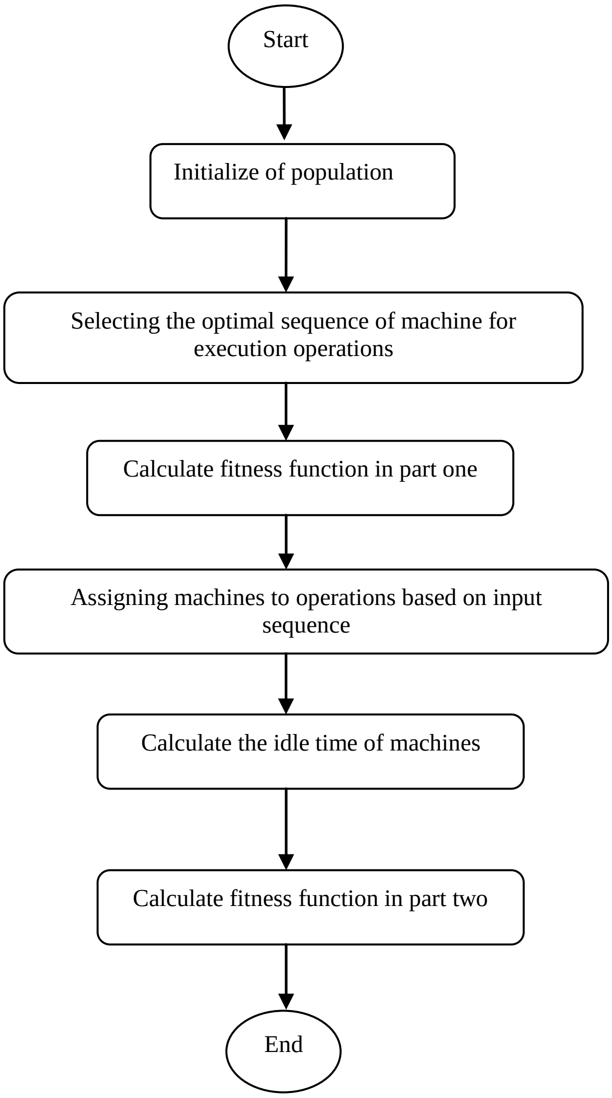

Fig 2: Outlines of proposed algorithm

Table 1: Execution time of jobs on machines

Number of jobs	Idle time(sec)	Minimum execution time(sec)
1	5	6
4	9	14
5	14	19
6	13	22
10	9	32

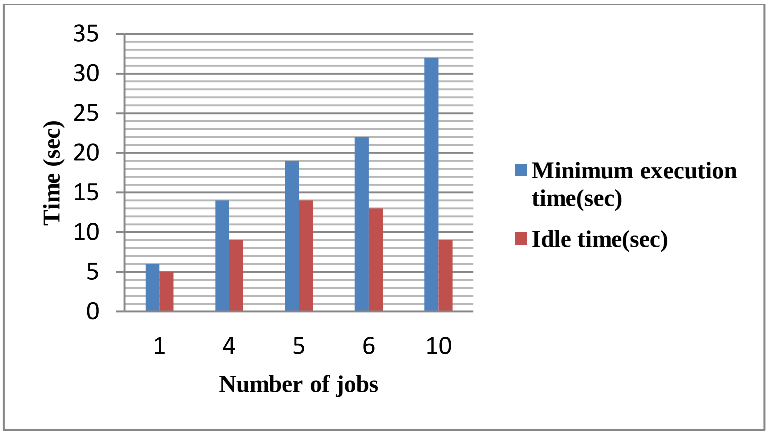

Fig 3: Flow execution time algorithm with increase jobs

Table 2: Compared scheduling algorithms with execution time

Number of jobs	MaxIter	Number of resource	GA	PSO	COA
1			6	6	6
4			15	14	14
5	50	3	23	22	19
6			25	23	22
10			45	39	32

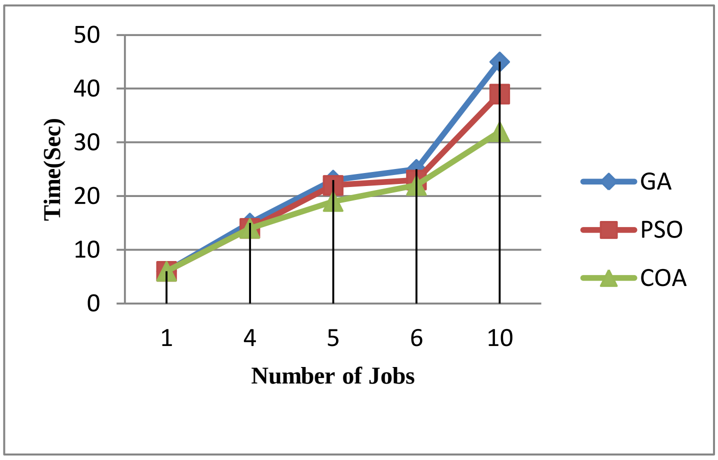

Fig 4: Comparative Performance GA, PSO and COA

8. CONCLUSION

In this paper, a scheduling algorithm based on COA is proposed for job scheduling problem on computational grids. Our approach is to generate an optimal schedule that complete the jobs in a minimum time. Simulation results compared with existing algorithms such as Genetic and PSO and demonstrate COA can allocate jobs to resources better than existing algorithms. Figure 4 shows a simple comparison between our proposed algorithm and some other related algorithms. As this figure shows, our proposed algorithm is better than all other algorithms. Also, our proposed algorithm has an important excellence. It has minimum execution time for job scheduling.

9. REFERENCES

[1] Foste. I, Kesselman. C," The Grid 2: Blueprint for a New Computing Infrastructure", 2nd ed., Morgan Kaufmann, 2004.

[2] Schopf. J," TEN ACTIONS WHEN GRID SCHEDULING", Mathematics and Computer Science Division,2004.

[3] Holland. J, "Adaptation in Natural and Artificial Systems", University of Michigan Press, re-issued by MIT Press ,1992,1975.

[4] Aggarwal, M and Kent, R, "Genetic Algorithm Based Scheduler for Computational Grids", Proceedings of the 19th International Symposium on High Performance Computing Systems and Applications (HPCS'05), 2005.

[5] Wang, J and Duan, Q, "A New Algorithm for Grid Independent Task Schedule：Genetic Simulated Annealing", 2005.

[6] Fidanova .S, Durchova. M, "Ant Algorithm for Grid Scheduling Problem", Springer-Verlag Berlin Heidelberg, pp. 405–412,2006.

[7] Carretero. J, Xhafa. F," GENETIC ALGORITHM BASED SCHEDULERS FOR GRID COMPUTING SYSTEMS", International Journal of Innovative Computing, Information and Control,vol. 3, No.6, Dec 2007.

[8] Zhang. L, Chen. H, R. Sun, S. Jing, B. Yang," A Task Scheduling Algorithm Based on PSO for Grid Computing", International Journal of Computational Intelligence Research, Vol. 4, No. 1, pp. 37–43, 2008.

[9] Mathiyalagan. P, Suriya. S, Sivanandam. N, "Modified Ant Colony Algorithm for Grid Scheduling", (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 02, pp. 132-139, 2010.

[10] Umale. J and Mahajan. S, "Optimized Grid Scheduling Using Two Level Decision Algorithm (TLDA)", 1st International Conference on Parallel, Distributed and Grid Computing (PDGC), 2010.

[11] Baker. M, Buyya. B, Laforenza. D, "Grids and Grid technologies for widearea distributed computing", The Journal of Concurrency and Computation: Practice and Experience, Vol 14, Nov. 2002.

[12] Yang. X, Deb, "Cuckoo Search via L´evy Flights", World Congress on Nature & Biologically Inspired Computing, pp. 210-214 , Dec 2009.

[13] Rajabioun. R, "Cuckoo Optimization Algorithm", Applied Soft Computing, PP. 5508–5518, 2011.

[14] Gao. Y, Rong. H, Zhexue. J, "Adaptive grid job scheduling with genetic algorithms", Future Generation Computer Systems, PP. 151–161, 2005.

[15] Fidanova. S, Durchova. M, "Ant Algorithm for Grid Scheduling Problem", Springer-Verlag Berlin Heidelberg, pp. 405–412, 2006.

[16] Selvarani. S, Sudaha Sadhasivam. G, " IMPROVED JOB-GROUPING BASED PSO ALGORITHM FOR TASK SCHEDULING IN GRID COMPUTING", International Journal of Engineering Science and Technology, Vol. 2, No. 9, pp. 4687-4695, 2010.

[17] Carretero. J," USE OF GENETIC ALGORITHMS FOR SCHEDULING JOBS IN LARGE SCALE GRID APPLICATIONS", ŪKIO TECHNOLOGINIS IR EKONOMINIS VYSTYMAS, Vol XII, No 1, PP. 11-17, 2006.

[18] Mathiyalagan. P, Dhepthie. U. R, S.N.Sivanandam," GRID SCHEDULING USING ENHANCED PSO ALGORITHM", International Journal on Computer Science and Engineering, Vol. 02, No. 02, pp. 140-145, 2010.