J. Basic. Appl. Sci. Res., 3(2s)104-109, 2013

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Comparative Study of Heuristic Algorithms and Nature-Inspired Algorithms for Scheduling in Grid Computing

Abacalat Jalalat[1], Behzad Mahdavi[2], Mohammad Salemi[3], Mehdi Effatparvar[4]

[1]Department of Computer Engineering, Germi branch, Islamic Azad University, Germi, Iran
[2]Department of Computer Engineering, Science and Research branch, Islamic Azad University, Ardabil, Iran
[3]Department of Computer Engineering, Ahar branch, Islamic Azad University, Ahar, Iran
[4]ECE Department, Ardabil Branch, Islamic Azad University, Ardabil, Iran

Received: June 10 2013
Accepted: July 10 2013

ABSTRACT

Distributed systems have an important role in high performance computing. One of the major challenges in this field that has a significant impact on the performance of such systems is the scheduling. The problem of scheduling tasks in heterogeneous systems due to the need for better use of existing computing systems and less time to implement scheduling algorithms, the important. Scheduling problem for heterogeneous distributed computing systems, such as scheduling problems are NP-Complete, and yet numerous models and algorithms to optimize the scheduling problem in heterogeneous systems is presented. In this paper, some of the scheduling algorithms and compare performance of these algorithms in different environments are heterogeneous.
KEYWORDS: Scheduling, Distributed Computing Systems, Scheduling Algorithms, Performance.

1.INTRODUCTION

A distributed heterogeneous computing system is a collection of heterogeneous computers Which are geographically distributed and interconnected via a communication network is very fast together with other applications, data and heterogeneous computing resources to share. Efficient execution of parallel applications on distributed computing systems, primarily used to perform tasks for scheduling parallel applications is dependent on the available processors[1]. Grid computing is also a heterogeneous distributed systems has drawn more attention nowadays And he also has brought many challenges, one of the challenges is the Scheduling Problem. In addition to strengthening the classical scheduling algorithms, a new survey methods, for example, scheduling adaptive Level-Application, Grid economy models, algorithms heuristic and algorithms inspired by nature are applied.

Today, large computational tasks and schedule them to run on heterogeneous systems has received much attention. In such systems, performance processor, the network size and the speed can be varied so that the lines of communication network is composed of a heterogeneous system. The timing of implementation of tasks in the system, including issues is difficult[2].

Grid scheduling is much more complex than local resource scheduling. Because it must manage and control resources on a larger scale. The old methods are not suitable schedules for resource management in the grid. The old ways are the System Centric users will not notice and to increase system performance, increase Utilization and meet the needs of resources. But today, the methods that are User Centric QoS for users.Figure1 shows a view of the scheduling in the Grid.

Resource management in heterogeneous computing environments in a whole is faced with two problems, One of the issues in the context of dynamic heterogeneity of the computing and other resources. Capabilities for heterogeneous computing systems must be adapted to the dynamic environment. Allocate tasks to available resources in a heterogeneous computing systems should be, If the workload is balanced distribution of resources to ensure maximum utilization of available resources. Models, a variety of methods and algorithms for scheduling tasks in heterogeneous computing environments has been reported, In this paper, a detailed review and evaluation of heuristic algorithms and algorithms inspired from nature and we will compare them.

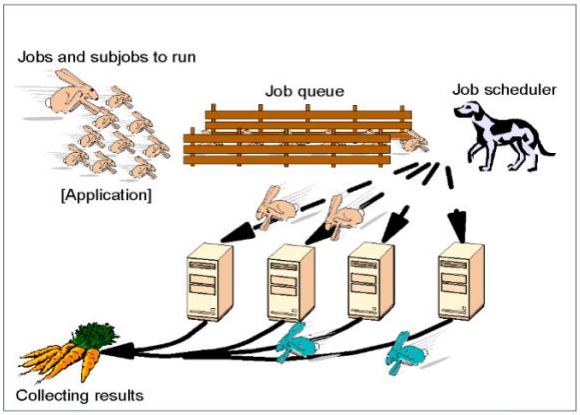

Figure1. View of scheduling in Grid

Corresponding Author: Abacalat Jalalat, Department of Computer Engineering, Germi branch, Islamic Azad University, Germi, Iran.
Email address: Jalalat.abacalat@engineer.com

In Section 2 we a short overview of the classification of scheduling algorithms. In Section 3, heuristic algorithms and algorithms inspired by nature have to be studied. and we will compare them in Section 4. The end conclusion of scheduling algorithms in Grid computing explains.

2. Classification of Scheduling Algorithms

High-level scheduling algorithms are classified into two categories: global and local. Local scheduling techniques on how to allocate decision-making processes to a processor can be implemented. Global scheduling techniques used in information systems allocate processes to multiple processors to optimize overall system performance occurs.

Scheduling the next level in the hierarchy is the choice between static and dynamic scheduling. The static algorithm requires information about all available resources in a heterogeneous computing environment scheduling is available at the moment (at compile time). One of the major advantages of this method is the simplicity of programming scheduling perspective. The main idea of the dynamic scheduling, allocation is requested at the time of application. This method is used when the estimated cost of the program or requests entered into the system as a dynamic problem. Scheduling requests dynamically, consists of two main parts: an estimate of the system state (in addition to the cost estimate static) decision. State estimation system consists of data collection across a heterogeneous computing system and the estimating is. According to the estimates, the assignment of a request to do is select the source[1]. In dynamic scheduling addition to scheduling, quality scheduling is also important.

scheduling algorithms in terms of the other two categories are classified into real-time or online and batch. Upon arrival in real-time working, job allocation to available processors is performed by a scheduler. In case the batch mode, operation allocation of work within specific time scheduling event is called, is done.

Another category of scheduling algorithms can be divided into two categories, scheduling optimal and suboptimal. The optimal scheduling, allocation optimized can be according to some criterion functions such as makespan and maximum use of resources. But given the complexity of heterogeneous computing environments must prove algorithm optimality. Heuristic algorithms, among the suboptimal scheduling requirements. In other categories, the algorithms are distributed against centralized algorithms. Finally, the last category of cooperative and non-cooperative scheduling algorithms. Given that, the grid type of special is a system, the scheduling algorithms in grid on a subset of this category (Figure2) are [3].

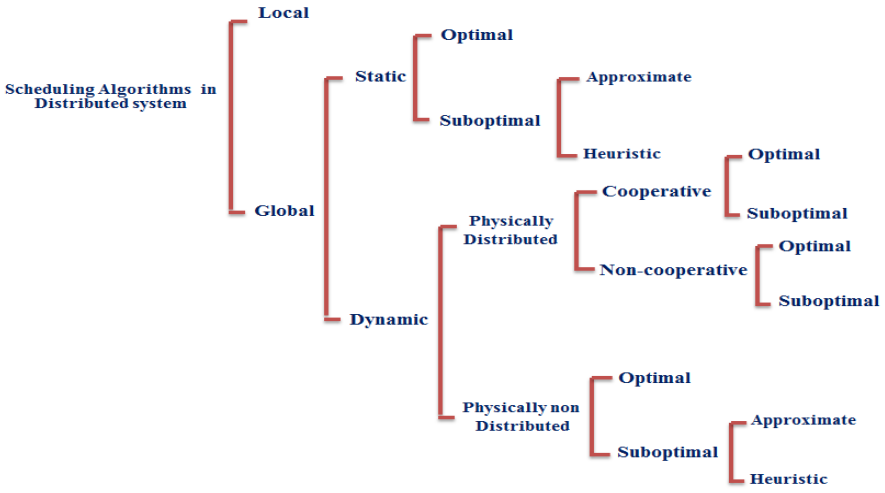

Figure2. Classification of scheduling algorithms in Grid

3 . Heuristic Algorithms and Inspired by Nature Algorithms

3-1. GA

Genetic Algorithm is an evolutionary technique for large space search. The general procedure of GA search is as following:

1) Population generation

A population is a set of chromosomes. Each chromosome represents a possible solution, which is a mapping sequence between tasks and machines. [4] Randomly generate 200 chromosomes.

2)Chromosome evaluation

Each chromosome is associated with a fitness value, which is the total completion time of the task-machine mapping this chromosome represents. The goal of GA search is to find the chromosome with optimal fitness value.

3)Crossover and mutate

Crossover and mutate the chromosomes selected based on selection rules. The selection rules are analogous to evolutionary selection rules. But in many papers chromosomes are randomly selected.

Crossover is the process of swapping certain subsequences in the selected chromosomes. Mutate is the process of replacing certain subsequences with some task-mapping choices new to the current population. Both crossover and mutation are done

randomly [4]. After crossover and mutation, a new population is generated. Then it will be evaluated, and the process starts over until some *stopping criteria* are met.

The stopping criteria can be, for example, 1) no improvement in recent evaluations; 2) all chromosomes converge to the same mapping; 3) cost bound is met.

3-2. Heuristic Min-Mean Job Scheduling

Online mode and batch mode are the two classifications of heuristic scheduling algorithms [5][6]. In online mode, the jobs are scheduled to the resources as soon as it arrives. In batch mode, the jobs are independent, there is no order of execution and jobs are scheduled as a batch every time. Here in this paper, batch mode scheduling is followed. Achieving the minimum makespan is the goal.

To evaluate the mapping heuristic, the expected time to complete [ETC] model is employed. Before execution, the expected execution time of the tasks on the machine should be known, and this is contained in the ETC matrix. Consider for the task ti and the arbitrary machine mj ETC[ti,m j]. This represents the expectedtime of the task i on the machine j. In this matrix, the row represents the expected execution time of a task on different machine and column represents the expected execution time of different tasks on the same machine[7][8][9].

Based on 3 characteristics, the benchmark of instances for distributed heterogeneous computing system is generated.

(1) Machine heterogeneity (low/high)
(2) Task heterogeneity (low/high)
(3) Consistency (Consistent/Inconsistent/Partially Consistent)

Combining these 3 bench marks, 12 combinations of ETC matrices are used to evaluate the heuristic minmean scheduling algorithm.

There are 2 phases in this algorithm [10]. In the first phase, all jobs are assigned to the resources. In the second phase, mean completion time of all jobs is calculated and the jobs are allocated to the machines whose completion time is less than the mean completion time. Machine who has maximum completion time is selected as makespan.

3-3. SA

Simulated Annealing is a search technique based on physical process of annealing, which is the thermal process of obtaining low-energy crystalline states of a solid. The temperature is increased to melt solid. If the temperature is slowly decreased, particles of the melted solid arrange themselves locally, in a stable "ground" state of a solid. SA theory states that if temperature is slowed sufficiently slowly, the solid will reach thermal equilibrium, which is an optimal state.

By analog, the thermal equilibrium is an optimal task-machine mapping (optimization goal), the temperature is the total completion time of a mapping (cost function), and the change of temperature is the process of mapping change. If the next temperature is higher, which means a worse mapping, the next state is accepted with certain exponential probability. The acceptance of "worse" state provides a way to escape local optimality which occurs often in local search. The simulation results prove that GA heuristic has the overall best performance but with most expensive search time cost. SA is not as efficient as its application in other domain science problems. More research on choosing efficient fitness values and selection rules are needed.

The conventional Sufferage heuristic uses MCT as metric for a mapping. "The idea behind Sufferage is that a host should be assigned to the task that would 'suffer' the most if not assigned to that host." The sufferage value of each task is the difference between the first MCT and the second MCT. The Sufferage heuristic doesn't work well for cluster resources because the MCT on the machines belonging to the same cluster are quite close, which makes the sufferage value approaches zero and eliminates cluster machines from selection. So Xsufferage heuristic developed for an application-level scheduler system– APPLeS [11]computes a cluster-MCT to enable the Sufferage heuristic to work in a cluster environment.

3-4. Community – Aware Schedulin Algorithm (CASA)

CASA is a decentralized dynamic heuristic metascheduling algorithm. In CASA, jobs can be rescheduled. In order to overcome the stagnation a probabilistic approach has been used to assign jobs so that the jobs are evenly distributed to all other resources.

CASA is a two phase algorithm [12]. The first phase is the job submission phase where each node receives the jobs that are submitted by local user. Consider a node A, it receives the job, it acts as a initiator node and requests all other nodes using the REQUEST message.

The other nodes who are willing to take the job will reply through ACCEPT message. The node A will evaluate the other participating nodes using the historic data and selects the appropriate node and submits the job to it.

The second phase is the dynamic rescheduling phase, the node which received the job will look for the job which has large enough waiting time and has not been selected recently in the local job queue. That job will be rescheduled to the other nodes.

5 algorithms are discussed in CASA. They are:

(1) Job distribution
(2) Job delegation request acceptance
(3) Job assignment
(4) Job rescheduling
(5) Job rescheduling request acceptance

Based on the above analysis of the algorithm, the algorithms are tabularized in Table 1 with advantages and disadvantages.

3-5. Firefly Algorithm

The firefly algorithm is based on swarm intelligence behavior of firefly. It is a meta heuristic algorithm inspired by the social behavior of firefly. Firefly algorithm finds the global optimal solution. The main focus of firefly algorithm is to complete the task within a minimum makespan and flowtime as well to utilize the grid resource efficiently. Firefly optimization as mentioned in [13] [14] in can be described as

(1) The firefly attracts and is attracted by all other fireflies
(2) The brighter one attracts the less brighter one
(3) The brightness decreases with distance
(4) The brightest firefly can move randomly
(5) The firefly particles can move randomly There are 4 phases in firefly algorithm [15] In the phase 1, the parameters are set (initial population, fitness and attractiveness), number of available resources and list of submitted jobs are identified.

In the phase 2, the brightness of each firefly is found at the source using fitness function and distance is calculated. The less bright fireflies are moved towards the brighter one.

In the phase 3, the new solution is evaluated and light intensity is updated In the phase 4, the fireflies are ranked and current global best is identified. Finally, the iteration parameters are updated All these are done until the termination condition is reached. The termination condition may be number of iteration or the fitness value or sometimes the saturation state.

3-6. Ant Colony Algorithm

The ant colony algorithm for job scheduling in grid aims at submitted jobs to resources based on the processing ability of jobs as well as the characteristics of the jobs.

Ant colony algorithm is the bio-inspired heuristic algorithm, which is derived from the social behavior of ants. Ants work together to find the shortest path between their nest and food source. When the ants move, each ant will deposit a chemical substance called pheromone. Using this pheromone, the shortest path is found.

The same concept is used to assign jobs in grid computing. When a resource is assigning a job and completes, its pheromone value will be added each time. If a resource fails to finish a job, it will be punished by adding less pheromone value. The issue here is the stagnation, where there is a possibility of jobs being submitted to same resources having high pheromone value.

In this ant colony algorithm [16], the load balancing method is proposed to solve the issue of stagnation. The algorithm is as follows

(1) The user will send request to process a job
(2) The grid resource broker will find a resource for the job
(3) The resource broker will select the resource based on the largest value in the pheromone value matrix
(4) The local pheromone update is done when a job is assigned to a resource.
(5) The global pheromone update is done when a resource completes a job
(6) The execution result will be sent to the user When the resource broker select a particular resource

for a job j, jth column of the Pheromone Value matrix will be removed and jobs will be assigned to other resources. Thus the load balancing is achieved.

3-7. PSO algorithm

Particle swarm optimization algorithm or PSO, which is also known as the Bird algorithm, One of the most popular and powerful algorithms for optimization is mainly due to the relatively high speed of convergence is to be used. The algorithm is a little old, but has been applied in many areas, the older algorithms, such as genetic algorithms, and surpass to be considered as a first choice.

PSO is a work based on the principle, Place at every moment of every particle in the search space that has been in the best place and the best place in the whole of his neighbors there, he set up.

To solve a problem, these algorithms with a random start solutions are called the initial batch. Each member of the group, is called a particle. Each particle represents a solution that consists of two parts. One section shows the σ_a (the post) and the other represents the σ_c (thus returning results) is, M is the number of elements in each part of it, is equal to the number worker computers[17]. Optimality of this algorithm is the value σ_c and σ_a.

In this theory, the model is used to implement the master and the workers. Client computer must divide the computational load to be distributed between the computers working and Worker computers to receive their full share of computing load and then waited to begin processing. Every Worker computer after processing is required within the Scheduling specified in the order that they return to the client computer. (If client computer, same local grid scheduler, so Worker computers, the resources that have the ability to run jobs. If client computer, Global Grid scheduler is so Worker computers, same local grid scheduler.) Such that if m, the number of Worker computers, with different arrangements for distribution and return considering the complexity of the optimal solution, the order of O (m! 2) will be. And the number of Worker computers is high, obtaining an optimal schedule, it will be very take time. So far the optimal solution to this problem is not polynomial time complexity.

4. Comparison of heuristic algorithms and algorithms inspired by nature

The algorithm presented in the previous section, we see that each of them for the benefit of those who have some disadvantages. In Table1 we describe the advantages and disadvantages of each of them.

Table1 . Advantages and Disadvantages Algorithms studied

PARAMETERS	ADVANTAGES	DISADVANTAGES
Genetic Algorithm(GA)	1)Overall performance is good 2)Achieving the optimum overall 3)Simply implementing	Cost of search time, is very high
Heuristic Min-Mean Job Scheduling	The main focus of minimizing the makespan is achieved	1)One resource can execute only one job at a time 2)Size and number of resources are static and should be know in prior
Simulated Annealing(SA)	1)Memory consumption is very low 2) Its implementation is relatively simple 3) - Acceptable answers, the focus on local search 4) - Ability to pass a local optimum, due to the random process driven	1)Local optimality problem 2) High dependence on the initial temperature
Community – Aware Scheduling Algorithm (CASA)	1)Stagnation problem is solved through Probabilistic approach 2) Has got strong adoptability	1)Message overhead
Firefly Algorithm	1)Position of the firefly is updated 2)Convergence and cost minimization	1)First come first served policy is followed 2)Initial population is fixed
Ant Colony Algorithm	1)Adaptability 2)Good robustness in dynamic environment	1)Information on each resources is needed 2)The pheromone value has to be updated every time which may result in high processing time
Particle swarm optimization algorithm(PSO)	1)Easy to use 2)Capable of fast convergence to the correct solution. 3) Method does not require previous knowledge	Local optimality problem

5. Conclusions

Grid computing can be a complex task in a shorter time and to solve efficiently the hardware is working. Necessary for the high performance of grid must be the best job Scheduling strategies used. Scheduling work is the most important step in grid computing, where users work are scheduled to different sources. With the traditional algorithms such as genetic algorithms, are efficient, but the cost of the grid is important. Algorithm should allow the determination of the cost for both is to save time for the user, Nature-inspired algorithms to solve these needs are somewhat heuristic algorithms. But still the extent of these algorithms are far from ideal. However, Task Scheduling in parallel and distributed systems have been studied intensely. New challenges in grid environments is still an interesting topic for research projects. By study of scheduling algorithms in grid computing can be realized within the current scenario, The heterogeneity, dynamics, calculations and data separation are the primary challenges in these matters is much research.

REFERENCES

1. Nirmeen, A., Bahnasawy, F.O., Magdy, A.K., Mervat, M. (2011): Optimization procedure for algorithms of task scheduling in high performance heterogeneous distributed computing systems, Egyptian Informatics Journal 12, 219–229.Blanco, D., Boada, B.L., Moreno, L. (2001): Localization by Voronoi Diagrams Correlation, IEEE International Conference on Robotics & Automation Seoul, Korea.
2. Monsefi, R., Hamidzadeh, J. (2009): Computational load scheduling can split taking the time to return results in heterogeneous systems using genetic algorithms, Computer Society of the Fourteenth Conference, Amirkabir University of Technology, Tehran – Iran.
3. Casanova, H., Legrand, A., Zagorodnov, D and Berman F. (2000): Heuristics for Scheduling Parameter Sweep Applications in Grid Environments, in Proc, of the 9th hetero-geneous Computing Workshop (HCW'00), pp. 349-363, Cancun, Mexico.
4. L He, S.A., Jarvis, D.P., Spooner, D., Bacigalupo, G., Tan and Nudd. (2005): Mapping DAG-based Applications to Multiclusters with Background Workload, in Proc. Of IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05), p.855-862.

5. Kun-Ming Yu, and Cheng-Kwan Chen., 2008. “An Adaptive Scheduling Algorithm for Scheduling Tasks in Computational Grid”, in IEEE Seventh InternationalConference on Grid and Cooperative Computing, pp.185-189.

6. Hesam Izakian, Ajith Abraham, and, Vaclav Snasel. “Comparison of Heuristics for Scheduling Independent Tasks on Heterogeneous Distributed Environments”,IEEE.

7. T.Braun, H.Siegel, N.Beck, L.Boloni, M.Maheshwaran, A.Reuther, J.Robertson, M.Theys, B.Yao, D.Hensgen, and R.Freund, “A Comparison Study of Static Mapping Heuristics for a Class of Meta-tasks on Heterogeneous Computing Systems”, In 8 th IEEE Heterogeneous Computing Workshop(HCW’99), pp. 15-29, 1999.

8. Tracy D.Braun, Howard Jay Siegel, and Noah Beck, “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, pp.810-837, 2001.

9. Qinma Kang, Hong He, Hongrun Wang, and Changjun Jiang, 1996. “A Novel Discrete Particle Swarm Optimization Algorithm for Job Schedulingin Grids”, in IEEE Fourth International Conference on Natural Computation, pp.401-405.

10. G. K. Kamalam and V. Murali Bhaskaran, “New Enhanced Heuristic Min-Mean Scheduling Algorithm for Scheduling Meta-Tasks on Heterogeneous Grid Environment”, in European Journal of Scientific Research,Vol.70 No.3 (2012), pp. 423-430.

11. E. Heymann, M. A. Senar, E. Luque and M. Livny, *Adaptive Scheduling forMaster-Worker Applications on the Computational Grid*, in Proc. of the 1st IEEE/ACM International Workshop on Grid Computing, Lecture Notes In Computer Science, Vol. 1971, pp.214-227, Bangalore, India, December 2000.

12. Y. Huang, A. Brocco, N. Bessis, P. Kuonen, B. Hirsbrunner, “Exploring Decentralized Dynamic Scheduling for Grids and Clouds using the Community-Aware Scheduling Algorithm”, in: 2011 Future Generation Computer Systems, Elsevier.

13. Yang, X.S., Nature-inspired metaheuristic algorithms. 2010: Luniver Press.

14. Senthilnath, J., S. Omkar, and V. Mani, Clustering using firefly algorithm: Performance study. Swarm and Evolutionary Computation, 2011.

15. Adil Yousif, Abdul Hanan Abdullah, Sulaiman Mohd Nor and Adil Ali Abdelziz, “SCHEDULING JOBS ONGRID COMPUTING USING FIREFLY ALGORITHM”, in Journal of Theoretical and Applied Information Technology,2011. Vol. 33 No.2.

16. Ku Ruhana Ku-Mahamud and Husna Jamal Abdul Nasir. “Ant Colony Algorithm for Job Scheduling in GridComputing”, in 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and ComputerSimulation, IEEE Computer Society 2010.

17. Monsefi, R., Hamidzadeh, J., Javadi, Z., "Scheduling computational load can split taking the time to return the results of the algorithm in heterogeneous systems using PSO" Computer Society of Iran, Energy Technology Development Center, 2011