A SURVEY ON OPTIMIZATION BASED JOB SCHEDULING IN GRID COMPUTING

Gogila G. Nair*

Karthikeyan P*

Abstract —

Grid computing is presently a full of life analysis space. The main motivation of Grid computing is to combine the facility of cosmopolitan resources, and supply non-trivial services to users. An economical Grid planning system is a vital part of the Grid. Instead of covering the total Grid planning space, this survey provides a review of the topic primarily from the angle of different grid scheduling algorithms. Scheduling refers to the mapping of tasks to resources that may be distributed in various administrative domains. Motivation of the survey is to encourage the amateur researcher in the field of grid computing, so that they can understand the concept of optimized job scheduling easily and can contribute in developing more efficient optimized job scheduling algorithm.

Index Terms - Grid computing, Task scheduling, Scheduling algorithms, Chemical Reaction Optimization.

* Department of Computer Science and Engineering, Karunya University. India

I. INTRODUCTION

Grid computing combines computers from various administrative domains to achieve a common goal in order to solve a single task. Recent research on the availability of powerful computers, high-speed networks, low-cost commodity components and the popularity of the internet has led to the emergence of a new paradigm known as Grid computing. Grid facilitates large-scale distributed resource sharing. The Grid consists of information which is both static and dynamic in nature. Grid is an infrastructure that enables the integrated collaborative use of high-end computers, networks, databases and scientific instruments owned and managed by multiple organizations [8]. Grid technology has a wide range of applications in many fields of science and engineering, such as astronomy, meteorology, bioinformatics, transportation, financial modeling, drug discovery, high energy physics, data mining, and image manipulation.

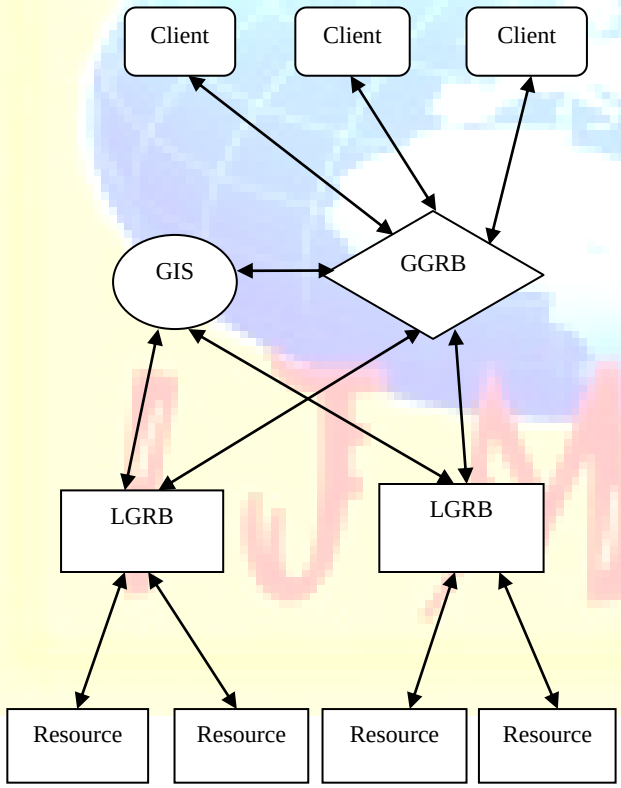

Figure 1: The architecture of the Grid System

The main goal of scheduling is to achieve the highest possible system throughput and to match the applications requirements with the available computing resources. A grid [11] usually consists of five parts (Figure 1): clients, the Global and Local Grid Resource Brokers (GGRB and LGRB), Grid Information Server (GIS), and resource nodes. The clients register their requests of processing their computational tasks at GGRB. Resource nodes register their donated resource at LGRB and process clients' tasks according to the instructions from LGRB. In practice, client and resource node can be the same computer. GIS collects the information regarding resources from all LGRBs, and transfers it to GGRB. This GGRB is responsible for scheduling. It possesses all necessary information about the tasks and resources and acts like a database of the grid.

Each instance of the User entity represents a Grid user. Each user may differ from the rest of the users with respect to different characteristics such as types of job created, job execution time and scheduling optimization strategy. Each resource may differ from the rest of resources with respect to number of processors, cost of processing, Speed of processing and internal process scheduling policy.

Grid provides services with high reliability and lowest cost for large volumes of users and support group work and the most important issues in grid computing are resource management and control, reliability and security. To increase the efficiency of grid a proper and useful scheduling is needed. Grid computing solves high performance and high-throughput computing problems through sharing resources ranging from personal computers to supercomputers distributed around the world. One of the major problems is task scheduling, i.e., allocating tasks to resources. There are several algorithms such as simulated annealing, genetic algorithm, ant colony optimization, particle swarm optimization, threshold accepting etc., for solving this problem.

II. SCHEDULING ALGORITHMS

Scheduling refers to the mapping of tasks to resources that may be distributed in various administrative domains. Scheduling can also be defined as the method by which processes or tasks or jobs are given access to system resources. This is usually done to balance a system effectively and to achieve the target.

A. Simulated Annealing (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 eventually decreased, particles of the melted solid arrange themselves, 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. Simulated Annealing algorithm [4] for task scheduling in the grid environment starts by generating an initial solution. For each iteration, SA generates a new solution randomly in the neighborhood of the present solution, and it will be accepted if it is better, or accepted with a probability controlled by a temperature parameter. As the temperature gradually drops, the ability to jump out of local optima decreases and finally moves to the global optimum. The most important part in the application of SA is generation of the initial solution and creating a set of neighbors.

The details of the initial solution generation and creation of the neighbor set for grid scheduling algorithm is as follows. Let the number of the tasks in the set of tasks is greater than the number of machines in the grid. The result will be triples (task, machine, starting time). To generate initial solution greedy heuristic can be used. The first task in the set will be executed on the first free machine, and the same method is used for the second task in the set and so on. So the initial solution is a feasible solution. After the generation of a feasible solution the set of neighbors will be created. As is written above the solution is triple (task, machine, starting time). The solution can be thought of like a matrix with three columns, the first column contains the tasks, the second column is the corresponding machines and the third is its corresponding starting times. The order in the columns is based on the starting time. Thus the tasks with early starting time are before tasks with later starting time. To create new solution two of the tasks will swapped. It changes the starting times and reorder succeeding tasks. The performance of the achieved result is highly dependent on the right choice of both specific and generic choices. This algorithm statistically guarantees finding an optimal solution. It is relatively simple to code, even for complicated problems and generally gives a good solution. In SA technique it is very important how the set of neighbors is created. This algorithm may consume more time to find the good solution.

B. Genetic Algorithm (GA)

A Genetic Algorithm is an evolutionary technique for large space search. The general procedure of GA search includes Population generation, Chromosome evaluation and Crossover and mutation. A population is a set of chromosomes. Each chromosome represents a possible solution that is a mapping sequence between tasks and machines. Each chromosome is associated with a fitness value, which is the total completion time of the task-machine. The goal of GA search is to find the chromosome with optimal fitness value. First it generates the initial population [9] randomly. The initial population may be generated by any other heuristic algorithm; if the population is generated by Min-Min then it is called seeding the population with Min-Min. This genetic algorithm randomly selects chromosomes. Then crossover and mutate the chromosomes selected based on selection rules or can be 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. This crossover and mutation are done randomly. After crossover and mutation is performed, 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, no improvement in recent evaluations or all chromosomes converge to the same mapping; 3) a cost bound is met.GA works with a population of points instead of a single point. GA uses the previously obtained information more efficiently. GA is the most popular nature's heuristic used in algorithms for optimization problems. GA heuristic has the overall best performance however with most expensive search time cost. Additionally the convergence time for GA is more.

C. Ant Colony Optimization (ACO)

The Ant algorithm is introduced by Dorigo M. in the year 1996. It is based on the real ants and it is derived from the social behavior of ants [3]. Ants work together to find the shortest path between their nest and food source. When an ant looks for food, it deposits some amount of chemical substance called pheromone on the path. The shortest path is found using this pheromone. The ant's moves consist of transitions from nodes to nodes. If an ant tries to move from one place to another then it encounters an already laid trail. The ant can detect that pheromone trail and decide with high probability to follow it. This ant also reinforces the trail with its own pheromone. When more ants are following the same trail, then the pheromone on

shorter path will increase quickly. The quantity of pheromone on every path will affect the possibility of other ants to select path. At last all the ants can opt for the shortest path. The same concept is used in grid computing to assign the jobs to resources.

When a job is assigned and it completes its pheromone value will be added each time. If a resource fails to finish a job, less pheromone value will be given to that resource. The main issue to be considered here is the stagnation, which means there is a possibility for the jobs being submitted to the same resources having high pheromone value. The load balancing method is proposed to solve the issue of stagnation. The user will send request to process a job. The grid resource broker will find a resource for the job. The resource will be selected by the resource broker based on the largest value in the pheromone [7] value matrix. The pheromone trails are updated in two ways. The local pheromone update is done when a job is assigned to a resource. The global pheromone update is done when a resource completes a job. And finally the execution result will be sent to the user. In this algorithm the Initial decisions on which path to choose are made at random. ACO is more applicable to problems where source and destination are predefined and specific and also to problems that require crisp results.

D. Particle Swarm Optimization (PSO)

The Particle swarm optimization is a population-based swarm intelligence algorithm. It is modeled on swarm intelligence, like bird flocking and fish schooling. It is starts with a group of particles known as the swarm. A PSO algorithm [5] contains a swarm of particles in which each particle includes a potential solution.PSO algorithm is an adaptive method that can be used to solve optimization problem. Conducting search uses a population of particles. Each particle corresponds to individual in evolutionary algorithm. A flock or swarm of particles is randomly generated initially with each particle's position representing a possible solution point in the problem space. Each particle has an updating position vector X i and updating velocity vector V i by moving through the problem space. Each particle is aware of its own best position pbest and the best position so far among the entire group of particles gbest. The pbest of a particle is that the best result (fitness value) reached so far by the particle, whereas gbest is that the best particle in terms of fitness in the complete population.

The algorithm starts with random initialization [10] of particle's position and velocity. The particles are the task to be assigned and the dimension of the particles is the number of tasks in a

workflow. The value assigned to each dimensions of a particles are the computing resources indices. Thus the particles represent a mapping of resource to the task. The evaluation of each particle is done by the fitness function. The particles calculate their velocity and update their position. The evaluation is carried out until the specified number of iterations (user-specified stopping criteria). PSO algorithm provides a mapping of all the tasks to a set of given resources based on the processing capability of the available resources. In PSO, the population is the number of particles in a problem space. Particles are initialized randomly. This approach aims to generate an optimal schedule so as to get the minimum completion time while completing the tasks. Each particle is a function of design variables and it is improved through the algorithm by changing the position of particle on the search space. Particles, similar to individuals, not only remember their own local best positions (solutions), but also communicate with each other and record the globally best position. PSO have no overlapping and mutation calculation and hence the convergence time for PSO is less. But this method easily suffers from the partial optimism.

E. Threshold Accepting (TA)

Threshold Accepting is similar to SA but with a different acceptance rule. Every new solution would be accepted as long as the difference is smaller than a threshold. TA algorithm [2] begins with an initial solution *S* and an initial threshold value *T*1. A neighborhood solution *S** to the current solution is generated by using the perturbation scheme. There

Table 1: Comparison of various scheduling algorithms

Algorithms	Advantages	Disadvantages
Simulated Annealing (SA)	Provides good solution and easy to code even for complex problems	Consumes more time to find the good solution

Genetic Algorithm (GA)	Provides good solutions	Consumes more time and performance degrades as the number of jobs and resources increases
Ant Colony Optimization (ACO)	Provides good solution	Applicable only where the source and destination are predefined
Particle Swarm Optimization (PSO)	Better convergence speed	Performance degrades as the number of jobs and resources are increased
Threshold Accepting (TA)	Greater simplicity	It involves several iterations and high convergence time

are three perturbation schemes, namely, the Pair wise Exchange, the Insertion technique, and the Random Insertion technique. Consider that the workflow sequence 1, 2, 3, 4, 5 is a seed sequence and that the integers i, j (i, j <= Wn : where Wn = number of workflow) are randomly generated. Suppose in the first instance i=1 and j=4, the pair wise exchange technique will generate the new sequence 4,2,3,1,5; the insertion technique will generate a new sequence 2, 3, 4, 1, 5 and in the random perturbation scheme the digit in the first position can be inserted at any position to its right.

The current and the candidate solution are evaluated and their objective function value is obtained. If the candidate solution is acceptable, it becomes the current solution and this completes an iteration of the TA procedure. After the completion of individual iterations, the threshold value is reduced by *r*, known as threshold reduction step size and the iteration is repeated. The algorithm is terminated when a final threshold *T*2 is reached. An apparent advantage of the TA is its greater simplicity. It is not necessary to compute probabilities or make random decisions. Since it involves several iterations the convergence time is more.

F. Chemical Reaction Optimization (CRO)

Chemical Reaction Optimization is a population-based metaheuristic, and it can be used for solving many problems. CRO mimics the interactions of molecules in chemical reactions to search for global optimum [11]. A chemical system undergoes a chemical reaction when it is unstable, in the sense that it possesses excessive energy. It manipulates itself to release the excessive energy in order to stabilize itself. This manipulation is called chemical reactions. In a chemical reaction molecules interact with each other aiming to reach the minimum state of free energy. Through a sequence of intermediate reactions, the resultant molecules (i.e. the products in a chemical reaction) tend to stay at the most stable state with the lowest free energy.

When looking at the chemical substances at the microscopic level, a chemical system consists of molecules, which are the smallest particles of a compound that retain the chemical properties of the compound. Molecules are classified into different species based on the underlying chemical properties. A chemical reaction always results in more stable products with minimum energy and it is a step-wise process of searching for the optimal point. A chemical change of a molecule is triggered by a collision. There are two types of collisions: uni-molecular and inter-molecular collisions. The former describes the situation when the molecule hits on some external substances (e.g. a wall of the container) while the latter represents the cases where the molecule collides with other molecules. The corresponding reaction change is called an elementary reaction. An ineffective elementary reaction is one which results in a subtle change of molecular structure.

There are three stages [1] in CRO: initialization, iteration and the final stage. The computer implements CRO by following these three stages sequentially. Each run starts with the initialization, performs a certain number of iterations, and terminates at the final stage. There are four major operations called elementary reactions in CRO: on-wall ineffective collision,

decomposition, inter-molecular ineffective collision, and synthesis. In CRO, on-wall ineffective collision and inter-molecular ineffective collision correspond to local search, while decomposition and synthesis correspond to remote search. Through a sequence of intermediate reactions, the resultant molecules (i.e. the products in a chemical reaction) tend to stay at the most stable state with the lowest free energy. An on-wall ineffective collision occurs when a molecule hits the wall and then bounces back. Some molecular attributes change in this collision, and thus, the molecular structure varies accordingly. An inter-molecular ineffective collision describes the situation when two molecules collide with each other and then bounce away. A decomposition means that a molecule hits the wall and then decomposes into two or more (assume two in this framework) pieces. A synthesis depicts more than one molecule (assume two molecules) which collide and combine together.

Chemical Reaction Optimization is a variable population based metaheuristic [6] where the total number of solutions kept simultaneously by the algorithm may change from time to time. Decomposition and synthesis increases and decreases the number of molecules, respectively. Several CRO programs corresponding to the different modules can be implemented simultaneously. CRO is best suited to those types of problems which will benefit from parallel processing rather than sequential processing.

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

III. CONCLUSION

An analysis on various scheduling algorithms for grid environment is done. Each algorithm has its own advantages and disadvantages. From this analysis it is clear that the Chemical Reaction Optimization algorithm has superior performance. It is an efficient solution for the grid scheduling problem.

REFERENCES

[1] Albert Y.S. Lam and Victor O.K. Li, "Chemical Reaction Optimization: A Tutorial (Invited paper)," *Memetic Computing*, vol. 4, no. 1, pp. 3-17, Mar. 2012.

[2] S. Benedict, R.S. Rejitha, and V. Vasudevan, "Threshold Accepting Scheduling algorithm for scientific workflows in Wireless Grids", *Proc. IEEE Fourth Int'l Conf. Networked Computing and Advanced Information Management (NCM '08)*, 686-691, 2008.

[3] W. Chen, and J. Zhang, "An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements", *IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews*, 2009.

[4] S. Fidanova, "Simulated Annealing for Grid Scheduling Problem," *Proc. IEEE John Vincent Atanasoff Int'l Symp. Modern Computing (JVA '06)*, 41-45, 2009.

[5] H. Izakian,B.T. Ladani, K. Zamanifar, and A. Abraham, "A Novel Particle Swarm Optimization Approach for Grid Job Scheduling," *Proc. Third Int'l Conf. Information Systems, Technology and Management*, 2009.

[6] A.Y.S. Lam and V.O.K. Li, "Chemical-Reaction-Inspired Metaheuristic for Optimization," *IEEE Trans. Evolutionary Computation*, vol. 14, 381-399, 2010.

[7] S. Lorpunmanee, M.N. Sap, A.H. Abdullah and C. Chompoo-inwai, "An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment," *World Academy of Science, Engineering and Technology*, 2007.

[8] F. Pop, and V. Cristea, "Optimization of Schedulig Process in Grid Environments," *U.P.B. Sci. Bull., Series C*, 2009.

[9] S. Prabhu, and V. Naveen Kumar, "Multi-Objective Optimization Based on Genetic Algorithm in Grid Scheduling," *International Journal of Advanced Research and Technology(IJART)*, 54-58, 2011.

[10] L. Zhang, Y. Chen, R. Sun, S. Jing, and B. Yang, "A Task Scheduling Algorithm Based on PSO for Grid Computing," *International Journal of Computational Intelligence Research*, 37–43, 2008.

[11] Jin Xu, Albert Y.S. Lam and Victor O.K. Li, "Chemical reaction optimization for task scheduling in grid computing*" IEEE Transactions on Parallel and Distributed systems*, vol. 22, No. 10, 2011.