A New Algorithm by Index Parameters at Grid Environment with Static Scheduling

Arash Ghorbannia Delavar and Mohsen Khajeh naeini

***Abstract*—In this paper is introduced a new algorithm by index parameters at grid environment with static scheduled that this proposed algorithm is used as more efficient method than previous studied algorithms. RQSG algorithm can increase the efficiency rather than previous algorithms using index parameters, and also is introduced a new technique method using proposed algorithm for better optimizing with mixing of communication cost and fitness function. In this situation is used choosing the best tasks with RQSG algorithm to create a suitable method for solving the problem which this method didn't use in previous algorithms. It will be introduced an algorithm for optimized choosing of processing sources based on a technique method, that with this algorithm and also with various simulations is increased the reliability and dependability in competition of previous algorithms. Also this algorithm has decreased dramatically the certification time of final processing source with the most optimized state. Finally the mentioned algorithm has increased the servicing quality for using the task in processing sources and also has created the best state of optimized path in index parameters at grid environment.**

***Index Terms*—Genetic algorithm; scheduling; grid systems; dependentttask; RQSG**

I. Introduction

In recent years have done some studies on Genetic Algorithms at grid environment because of the problems which there are in Genetic Algorithms. The specialists have used from various methods for mixing of current algorithms for solving the problems which are created in people brain by Genetic Algorithm.

In the past Distributed systems were used from the sources and parallelize the identified process time. In information systems, the most expensive item is the information itself, if the amount of information would be high, by layering the information, you can classified the amount of information to optimize the searching time. That's why in the distributed systems because huge amount of information the speed will be decreased [1].

To achieve the objective distributed systems should be classified in two phases we studied:

The first phase of software, with previous exact algorithms optimizes a new algorithm to be presented to the previous case with the fault tolerance and dependable more done.

Proposed algorithm must be the same environmental conditions until obtained output is optimal than previous mode. But in the second phase of the hardware, network's communications environment and processors that can speed run processing time processes on independent processing than similar processes with regard to the above conditions reduce and with using the proposed algorithm can reduce allocation of time allocation process program and we can optimize with the above algorithm, faster than output program. Also, consuming time for a number of programs to implement applications with high volumes of data taken, with limiting the scheduler program and the proposed algorithm in the Process Scheduler program sweep mode with independent processing resources, we could able put it to use [2].

GA uses the bases of Darvins natural evolutionary selecting method in order to find the optimal answer and to solve the problems with the NP-Complement complex and scheduling grid resources in which includes in this group, which is very useful [3].

In this paper is introduced a new algorithm for optimized scheduling of non homogenous tasks on non homogenous sources according to Genetic Algorithm. This algorithm can minimize the makspan time with considering of different indices and parameters of fitness function and also controlling of created generations by Genetic Algorithm. Also the efficiency of grid environment has increased by this algorithm.

In this method is used from check point method for algorithm efficiency maintaining at high failure rate environments, so the used method has fault tolerant

In this paper is used from roulette wheel schema for creating the first generation and also used from Gridsim for simulations.

II. Literature Review

At the first we reviewed the related works on dynamic task scheduling at grid calculations, then we introduced own method for solving the reconstruction problem and also discoveries at grid tasks scheduling. Although most of researchers have different comments on tasks dynamic scheduling but the most common of real function at task scheduling problems is makspan function. In grid there is this possibility that makspan from non optimized scheduled has more time than interest makspan because the networks calculation power varies with time. [4]

In the past task scheduling was proposed for super computers and parallel computers [5]-[11].

Hwang and Kesselman say that grid environment isn't reliable by nature. They proposed a failure detector service and a flexible item for failure controlling as a failure tolerant mechanism on grid. In grid environment breaking the task running is done because of some reasons such as network incompetence, bad condition of sources, unreliability to

Manuscript received April 12, 2012; revised May 12, 2012.
The authors are with the Department of Computer Science, Payam Noor University, and Tehran, IRAN.

elements of needed software's. So the grid system should be able to detect and control the failure and also be able to protect of reliable running at the same time.

The scheduling of task on grids is an active area for research subjects and is focused on heuristic techniques. Using the executive model which select the sources and the allocated tasks to it prepare the limited and reliable algorithms for grid sources. Hwang and his workers proposed the failure detection service and also trade of with grid failures. Limaye and his co workers developed checkpoint/restart mechanism that the checkpoint places were obtained according to system reliability.

Dongara and his co workers used from failure protection result and running time homogenous actions for leading the independent task scheduling on non homogenous clusters. There are many sources management and scheduling models in grid calculations that their difference is in their scheduling strategy.

Scheduling strategy based on sureness mechanism:

Scheduling strategy based on sureness mechanism collects the sources information by some of the grid system tools such as MDS and NWS and then returns them. these strategies can upgrade some of items such as cost and task finishing time, they also have a success proportion from task running.

They work in local scheduling better than large scales. The structure of sureness mechanism is difficult in large scales. The other type of scheduling strategy is done by algorithm that in this strategy all of tasks are divided to smaller parts equally and then these parts are allocated to processing sources with a constant proportion. This type of scheduling works well and is suitable at non homogenous environments such as grid but because of calculations and communications of overload the efficiency of system is lower than optimized position. Other scheduling strategies: There are many classic strategies that are used in calculation grids such as PUNCH, Optimal Job Scheduling, Using Legacy Codes, [5]

In all of these strategies are determined the number of sending tasks to any processing source by scheduled module. The variations which are accrued cant effect on scheduled, really there isn't any feedback and these strategies cant scheduled without interruption.

III. RQSG Algorithm

In this algorithm is introduced a suitable method in graphs leveling that this leveling is done in 3 levels:

Zero level: In this level graphs should create index scheduling survey Using new algorithm to priority of each node. This algorithm creates a more optimized scheduling survey rather than previous algorithms

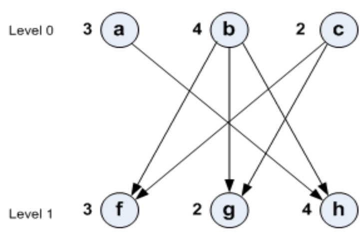

Fig. 1. 2 level initial graphs with RQSG proposed algorithm

RQSG index parameters with 2 level graphs:

For example suppose that there are 4 nodes in graph that this graph is shown in figure 1. Node b is selected as the first node according to scheduling priority with RQSG algorithm. Then will be used node a. the available problem in this figure is that its weight factors aren't used accurately and also there isn't a suitable method for creating the priority function. So this 2-level graph can change into completed 3-level one for choosing optimized priority.

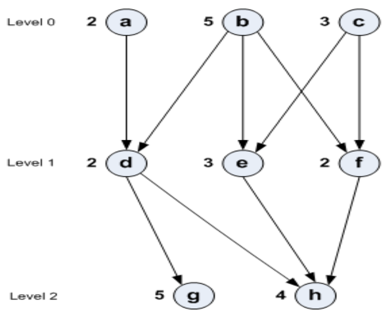

Fig. 2. 3-level graph with RQSG proposed algorithm

This completed graph that is created by index parameters is defined with these parameters:

Ei : the number of input vertex

Eo: the number of output vertex

L : level

R: needed processing power amount

1) E_i: the number of input vertex arrived each node.

In RQSG algorithm are done static tasks that these tasks create a communication between top and bottom layer node according used E_i parameter. This parameter causes to have an accurate priority choosing rather than tasks and also this parameter can be effective on priority function.

2) E_o: the number of output vertex from each node. Another index parameter which is used in RQSG algorithm is the number of output vertex that is studied by graph. This parameter determines the release amount of nodes which are finished their tasks and now is allocated a selected task to it in processing source.

3) L: the node level

Another index parameter is the number of graph leveling . For creating an optimized level is used from mixed function for covering all its states. For node level of L will create 2^L States for priority function. This parameter causes to the second level of nods doesn't start to work until the first level doesn't finished. This priority process will repeat for next levels.

4) R : the needed processing power value:

One of other index parameter is R in RQSG algorithm that introduces the needed processing power value for running the processing source. This parameter is used is certification function because of its positive effect on running the tasks in scheduling processing.

According to above definitions is used this function:

$$\alpha_j = \frac{E_o}{E_i + 2^L} \qquad j = 1,\dots,n$$

$$P_j = R_j \times \alpha_j \qquad j = 1,\dots,n$$

In RQSG algorithm data sending to processing sources and returning the obtained result to scheduled are done by message passing method and with considering of delay and band width of network. For having sureness from sending data to processing source and also arriving the obtained result to scheduled is used from a data box to scheduled chooses another source for task running if the sent data to processing node isn't delivered at certain time period.

Transmission time, including time of work transfer on processing source and result return time to Scheduler and acknowledge time are also is considered in this algorithm. Each processing source into the environment Grid, the bandwidth is calculated to Scheduler until the transition to the bandwidth obtained. When things reach Grid environment, are put in the queue not-Scheduling tasks. This queue processes as a batch and works is given allocation by the proposed algorithm to source processing suitable.

A. Chromosome Presentation

Each chromosome shows a possible solving for scheduler. The chromosomes are shown with a list of parameters (Table I)

TABLE I: SHOWING THE CHROMOSOMES IN OFFERED ALGORITHM

Node ID	R_1	R_2	R_3	…	R_{n-2}	R_{n-1}	R_n
Number of Task	9	3	5	…	1	4	7
Expected Execution Time	15	20	17	…	32	16	29

This type of chromosome representation helps us to have the needed expected execution time for doing any task that allocated to a computing node (without considering the transmission time). The first line of table.1 is the identities of computing node that the task is allocated to it. The second line is the mentioned task number and the second one is the expected time for doing the mentioned task on the relative computing node.

B. Fitness Function

For evaluating the efficiency level of selected computing node, is used from fitness function for doing the purposed task from obtained number. This function involves these parameters:

N: Number of computing node which joining the computation

NTR_j: Number of tasks which processed by computing node with ID=j

Computing capability of computing node with ID=j CRj:

WTi: Workload of task i

Primary file size of task with ID=i Di

RBj: Bandwidth between the scheduler and computing node with ID=j

The expected time for WT_i task doing on R_j computing node is calculated from followed equation:

$$EET_{i,j} = \frac{WT_i}{CR_j} + \frac{Di}{RB_j}$$

$$TR_j = \sum_{i=1}^{NTR_j} EET_{i,j}$$

$$j = 1, 2, \ldots, n$$

That Di is the transmitted data from scheduled to destination processing node, RB_j is the band with between them and TR_j is the sum of the WT_i tasks that are allocated to number j.

The maximum value of TR_j shows the makespan and the fitness function is defined as followed equation [2]:

$$FitnessValue = \left(\frac{1}{\max \{TR_j\}_{j \in N}} \right)$$

C. Procedures of the Algorithms

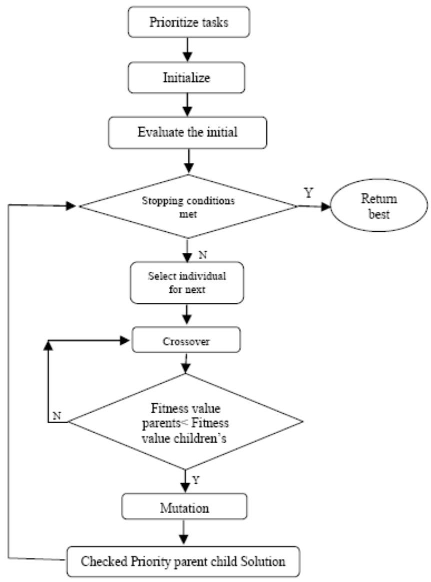

Fig. 3. Proposed Algorithm flowchart corresponding to standard GA.

1- Making the primary population: a population is a set of chromosomes which each of them represents one possible scheduling solution which is the order of mapping between source and tasks.

2- Chromosome evaluation: amount of fitness value for each chromosome will computed the goal in GA is to find chromosome with optimal fitness value.

3- Crossover operation: one pair of chromosome will randomly be selected and chooses a random point in the first chromosome and replaces each other spaces of second part of the two chromosome with each other. And calculated fitness value for avoiding to generate useless generation come up with the worst fitness value in contrast with its own parents, the function will be repeated. This function will be done in a limited cases.

4- Mutation operation: a chromosome will be chosen randomly and a task of it will be chosen randomly and is given to another node, this function causes that algorithm do not stopped in the local minimum.

5- Finally, the resulted population will be evaluated again. If one of the stopping condition will be done, algorithm will be finished otherwise, we go back to 3rd step. Conditions to end this process will be:

(1)no improvement in recent evaluations

(2)all chromosomes converges to the same mapping

(3)a cost bond is met.

II. Simulation Results

The used tool for simulations is Gridsim. In this paper is tried to select the simulation environment similar to previous studied algorithms as far as possible.

In Fig. 4 were studied a multipurpose simulation with task number=25 using the index parameter.

The proposed algorithm has had the more optimized the more optimized makspan rather than other two algorithms. Using the index parameters and also priority of static independent tasks by the proposed algorithm has been decreased the scheduling survey rather than the studied algorithms.

TABLE II: Showing The Number of Tasks 25 Using Available Nodes.

25 job	Number of Machines					
	2	3	4	5	6	7
NG	934	875	819	756	679	612
SGA	835	789	736	699	645	590
RQSG	800	760	720	685	632	582

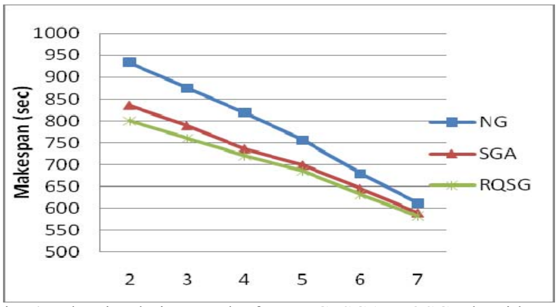

Fig. 4. The simulating results from NG, SGA, RQSG algorithms

Explanation about table 3: this table shows the number of static independent tasks using the studied various nodes.

Fig. 3 Competition between RQSG algorithm and another algorithm with the task number of 100.

Explanation about Fig. 5 in this figure RQSG algorithm will become more optimized when the number of static independent tasks is increased. Because in those static tasks is created less time delay by mentioned algorithm priority.

TABLE III: Showing the Number of Tasks 100 Using available Nodes.

100 job	Number of Machines					
	2	3	4	5	6	7
NG	3785	3552	3383	3159	2876	2476
SGA	3345	3176	2953	2788	2531	2263
RQSG	3062	2904	2714	2614	2424	2125

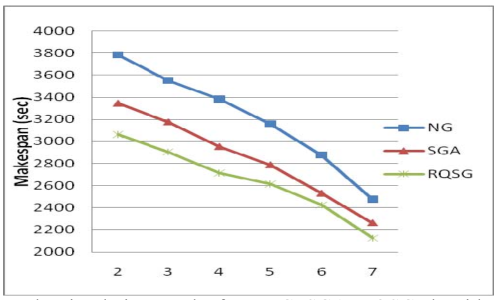

Fig. 5. The simulating results from NG, SGA, RQSG algorithms

III. Conclusion

With using of index parameters and static independent task priorities for allocation the tasks to processing source, the proposed algorithm RQSG is tested between the previous algorithms. In the state with fewer tasks this used algorithm has more optimized return time rather than similar previous algorithms, when the number of tasks is increased, in this algorithm static independent tasks coming and returning time and also its final certification time are decreased using various simulations. At the end are done tasks priorities to decrease the losses time of processing source.

References

[1] A. G. Delavar, V. Aghazarian, and S. Sadighi, ERPSD: "A New Model for Developing Distributed," *Secure and Dependable Organizational Softwares, CSIT*, 2009.

[2] A. G. Delavar, M. Nejadkheirallah, and M. Motalleb, "A New Scheduling Algorithm for Dynamic Task and Fault Tolerant in Heterogeneous Grid Systems Using Genetic Algorithm," *presented at IEEE ICCSIT*, 2010.

[3] A. G. Delavar and M. Rahmany and M. Nejadkheirallah, "GQSD: The new algorithm for optimizing the Quality of service heterogeneous possessing sources at distributed systems," *ICSTE*, 2010.

[4] N. Fujimoto and K. Hagihara, "Near-Optimal Dynamic Task Scheduling of Independent Coarse-Grained Tasks onto a Computational Grid," in *Proc. Int'l Conf. Parallel Processing*, 2003.

[5] K. Krauter, R. Buyya, and M. Maheswaran, "A Taxonomy of Grid Resource Management Systems for Distributed Computing," *Software-Practice and Experience*, vol. 32, no. 2, pp. 135-164, 2005.

[6] Y. K. Kwok and I. Ahmad, "Link-Constrained Scheduling and Mapping of Tasks and Messages to a Network of Heterogeneous Processors," *Cluster Computing*, vol. 3, no. 2, pp. 113-124, 2003

[7] M. Maheswaran, S. Ali, and H.J. Sigel, "Dynamic Mapping and Scheduling of Independent Tasks onto Heterogeneous Computing Systems," *J. Parallel and Distributed Computing*, pp. 107-131, 2001.

[8] S. Hwang and C. Kesselman, "A Flexible Framework for Fault Tolerance in the Grid," *J. Grid Computing*, vol. 1, no. 3, pp. 251-272, 2003.

[9] I. Foster and C. Kesselman, *the Grid: Blueprint for a Future* Computing *Infrastructure*, Morgan Kaufmann Publishers, USA, 1999.

[10] A. J. Page and Thomas J. Naughton, *Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing*, 2005.

[11] S. B. Priya and M. Prakash and K. K. Dhawan, Fault Tolerance-Genetic Algorithm for Grid Task Scheduling using Check Point, 2007.

Arash Ghorbannia Delavar received the MSc and Ph.D. degrees in computer engineering from Sciences and Research University, Tehran, IRAN, in 2002 and 2007. He obtained the top student award in Ph.D. course. He is currently an assistant professor in the Department of Computer Science, Payam Noor University, Tehran, IRAN. He is also the Director of Virtual University and Multimedia Training Department of Payam Noor University in IRAN. Dr. Arash Ghorbannia Delavar is currently editor of many computer science journals in IRAN. His research interests are in the areas of computer networks, microprocessors, data mining, Information Technology, and E-Learning.

Mohsen Khajeh Naeini was born in 1981. He received the MSc. degrees in computer engineering from Payame Noor University, Tehran, IRAN, in 2011.