Grid Load Balancing Using Parallel Genetic Algorithm

Nadra Tabassam Inam
COMSATS Institute of IT, Wah Cantt, Pakistan
Email: nadrainam@hotmail.com

Daud Awan
Preston University, Islamabad, Pakistan
Email: drdaudawan@hotmail.com

Hameed Ur Rehman
Faculty of Information Science and Technology, University Kebangsaan Malaysia
Email: rhameedur@gmail.com

***Abstract*—Grid computing is a form of distributed computing but different from conventional distributed computing in a manner that it tends to be heterogeneous, more loosely coupled and dispersed geographically. Grid computing can involve lot of computational tasks which requires trustworthy computational nodes. Load balancing in grid computing is a technique which overall optimizes the whole process of assigning computational tasks to processing nodes. Optimization of this process must contains the overall maximization of resources utilization with balance load on each processing unit and also by decreasing the overall time or output. Evolutionary algorithms like genetic algorithms have studied so far for the implementation of load balancing across the grid networks. But problem with these genetic algorithms is that they are quite slow in cases where large number of tasks needs to be processed. In this paper we give a novel approach of parallel genetic algorithms for enhancing the overall performance and optimization of managing the whole process of load balancing across the grid nodes**

***Index Terms*—distributed computing, grid computing, parallel genetic algorithm, fitness function.**

I. Introduction

With the advent of distributed computing we encounter large number issues regarding processing powers and load balancing across the distributed networks specially like grid computing. In order to get optimized results without any time delay and with maximum throughput we need enhanced algorithms across the networks.

Hence these faster processing results can be obtained by introducing a single processing unit or it can be inflated to diversity of processing types like multiple processing nodes, geographically distributed processing units and parallel processing. Grid computing is a modern and diverse form of distributed computing. It focused on the ability to support computation across multiple administrative domains that sets it apart from traditional distributed computing. Grids present a method of using the computational resources optimally within an association involving different computing resources. It supports multiple administrative domains and security authentication and other organizational mechanisms that capable it to be distributed locally or dispersed on the multiple geographic location in form of wide area network. Within the network connected nodes can share their resources over a network domain. These resources can be utilized by any other node that is connects to the grid network. I we ponder over this fact then we consider resource sharing across the grid computing network is an important factor over the internet networks [1], [2].

Multiple sharing resources are accumulated across the grid network. These resources involve number of dimensions including computational powers, processing capabilities. In grid computing environment all computational nodes are connected in such a manner that one node across the network is used for the admittance point for the whole set of resources which are physically dispersed across the network. So on the whole it seems for every user that their computer has powers analogous to super computers.

This assorted nature of gird has numerous challenges like resource management of multiple resources like storage capacity and processing powers, hardware and software based heterogeneity, multiple security issues while connecting to different administrative domain [3] when connecting to the grid. So on the whole the multi folded nature of grid has generated some stern concerns needed to be considered. Load balancing across the network manages the concept of load across the network in a manner that no node is over loaded or under loaded while the resources are used. These load balancing algorithms deal with the overall optimization of grid network in terms of resource sharing.

Evolutionary algorithms like genetic algorithms are from the family of heuristic techniques which are in used

Manuscript received November 15, 2014; revised March 4, 2015.

doi: 10.12720/ijeee.3.6.451-456

from long time for finding optimized solutions. These algorithms incorporate the mechanism inspired from biological evolution. There are many variants of genetic algorithm including different mutation and cross over procedure. Parallel genetic algorithms are quite new in this perspective but different from conventional genetic algorithm methodologies as they are faster and efficient in processing and gave more optimized results. There are many variants of parallel genetic algorithm including master slave model, coarse grain genetic algorithm, dual species genetic algorithm

II. Related Work

There are lot of examples in literature which deals with parallel processing in distributed environment.

In order to design well managed two-way applications writer has proposed the infrastructure named as middleware. This paper gives idea on a middleware concept and figure out the problems of service discovery, organization of communication between devices, harmonization, data-security and minimization of communication between the devices within distributed and collaborative environments [4].

Distributed system along with the concept of P2P systems is also a form of computer networks. By considering the strategies of mutual understanding and trust between the peers in P2P systems writer has proposed that by using different protocols set of reliable peers can be generated [5].

Schedulers based on genetic algorithms are quite efficient for jobs allocation to different computing resources in a grid environment. Author proposes a widespread study on the utilization of genetic algorithm for designing resourceful grid based schedulers for overall minimization of total completion time. Two schemes based on encoding have been focused and the majority of genetic algorithm operators for each of these schemes are implemented and logically studied [6].

Dealing with grid infrastructure at its service level generates two important issues which needs to be managed, these issues includes management of workloads and resource management in an optimized manner. Using neighborhood property and focusing on the tree based techniques generated optimized results in for managing load balancing across the grid networks [7].

Genetic algorithms to resolve the predicament of load balancing resourcefully. Proposed algorithm considers multi purposes in its resolution of assessment and resolves the scheduling problem in a way that concomitantly minimizes maxspan and overall decreases communication cost, and capitalizes the processors efficiency and utilization [8].

Proposed genetic algorithm make use of dynamic load-balancing methodology for solving issue of resource management across the networks Algorithm generated high-quality results when size of tasks to be scheduled become bigger. Implementation of central scheduler in grid environment overall optimized the scheduling process by considering the concept of least communication for making load balancing decisions along the grid networks [9].

Proposed approach to solve the task scheduling for grid. Local search algorithm based on the concept of gravitational attraction search is incorporated with genetic algorithm in order to enhance its capacity to search more sharply in problem based search space and attain more precise response in optimized time. Comparison between the results of HYGAGA and genetic algorithm emphasizes major improvement in the recital of search algorithm [10].

III. Proposed Methodology

A. Used Grid Model

We have taken in account the grid information system model for solving the issue of load balancing across the grid nodes. GIA is a grid model provides the information regarding the number of resources, number of tasks and state information of resources along with the information that which task is being served by which resource across the grid network. This model is good enough to use in the place where some kind of heuristics is being implemented. We enhance this model by adding the concept of parallel genetic algorithm.

We consider that there is a scheduler present in the grid. This scheduler according to GIS model has all the information related to the number of task. Let us consider that set of task come to the scheduler in GIS model of grid. These tasks are selected on the basis of criteria we define based on the total completion time of tasks.

We select those tasks first whose completion time is approaching to zero

$$Tct \rightarrow 0 \quad (1)$$

Suppose a new array of tasks based on their completion time reaches to scheduler for scheduling. The central scheduler has all the information about the current state of the system. At this point of time we mainly focus on the selection of optimized set of tasks for assigning to multiple resources for processing. In order to find the optimized set of tasks we use the concept of parallel genetic algorithm.

Parallel genetic algorithm is quite efficient as compare to conventional genetic algorithm techniques for finding optimized solutions. As the number of tasks are growing continuously so the concept of processing by only single processor is not optimized solution. The main scheduler selects three more schedulers for finding the set of optimized tasks. The main scheduler is called master scheduler while the rest of the three schedulers are known slave schedulers. These slave schedulers are selected on the basis of state information present at the master scheduler.

Let us consider the Fig. 1 for the illustration of this used grid model.

Fig. 1 shows the hierarchy of master and slave nodes in our proposed methodology.

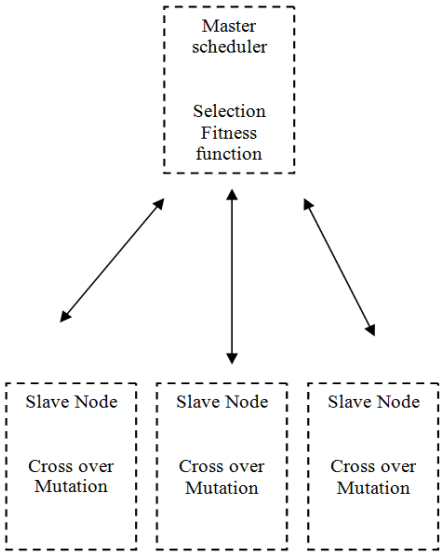

Figure 1. Master and slave nodes in scheduler

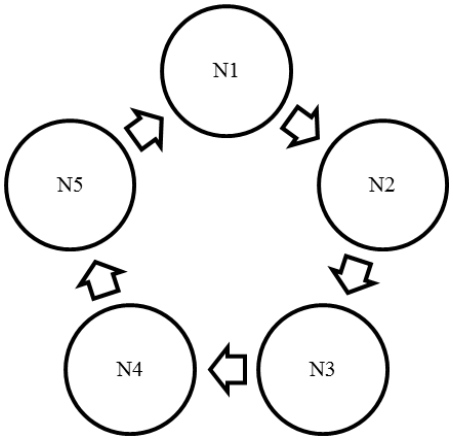

Figure 2. Group of nodes connected to grid

In Fig. 2 we suppose that we have currently five nodes connecting to the grid. Then the grid node N1 is consider to be the master node for assigning the tasks to the slave node and let's say we have N2, N3 and N5 are slave nodes depending upon the state of nodes and the current load on the node. These three nodes are selected for the processing of genetic algorithm and they return the optimized set of tasks that can be assigned to respective nodes for processing. The master node selects the node N2, N3 and N5 for the processing and implementation of genetic algorithm as these are the nodes with less loads. Now we see that how genetic algorithm is being practically implemented on these grid nodes.

IV. Selection of Optimized Set of Jobs for Assignment to Resources

A. Genetic Algorithm

The master node N1 implements the first two steps of genetic algorithm. Let's assume that node N1 gets the set of tasks or in other words an array of tasks which are selected on the basis of heuristic names as first come first serve. On the basis of FCFS basis the node N1perofrms the next step named as fitness function over the first set of population which are actually the tasks and we present these tasks in the form of numbers. We consider the tasks completion time as the population which passes through the fitness function.

We consider that at first on the basis of first come first serve we have the following set of tasks in the form of an array.

B. Initial Population

Initial population based on the completion time of each task: [3, 4, 2, 8, 9, 7, 11, 12, 4, 5, 4, 5, 7, 8, 9, 12, 13, 12, 14]. This is actually the completion time of each task in milliseconds. Genetic algorithm has initial population Most of the genetic algorithm selects the initial population without any define criteria. But we introduce the heuristic that is first come first serve for the selection of initial population. Now at next step the population passes through the fitness function.

C. Fitness Function

Fitness unction actually approves that which set of population is capable of reaching to the next step of cross over and mutations in genetic algorithm. Hence fitness unction is responsible for doing most of the task in genetic algorithm. Hence we tried that we consider all those factors which can overall increases the efficiency of load balancing procedure and can minimize the makespan.

We have taken in account the following factors for fitness function

Fitness function based on our algorithm comprise of important factors. The following equation gives our fitness function

$$F(pop)=\sum_{k=1}^{L_S}\{1/TPTQ_k*1/WT_k\}*CLP_k+DF \quad (2)$$

WT=Waiting time for each task before it is selected for processing
TPT=Total processing time needed by a task to complete
CLP=Current load on processor
L_S=Length of population string (which is selected on the basis of FCFS)
DF=Delay factor (if some kind of delay is occurred from any processing node)

Current load on the processing node can be measure by taking in consideration the following factors, the tasks that are currently being executed by processor as well as size of all tasks that are waiting in processors queue. This load on overall processor can find by using following equation.

$$CLP=\sum_{i=1}^{j}(CLi+RT_i)*1/TS \quad (3)$$

CL=Current load present on particular processor
RT=Time left for task that is currently processing by the processing node
TS=Total size of task in terms of time (time taken by each task to complete)
j=Number of processing nodes which are capable of processing

Our input string that passes through the fitness function is based on the total completion needed by each task. The result of fitness function is categorized in two ways.

$$F(pop)>0$$
$$F(pop)<=0 \quad (4)$$

The matrix defined below shows the fitness function of each population and its probability of selection. The highest probability strings are selected for mutation and cross over and for generating new child population to be used in next generation. Each string has to pass through this fitness function in order to select for the future population. The following Table I shows the selection of strings.

TABLE I. PROBABILITY OF SELECTION

Population	F(pop)	Probability of selection for each population
1	.0101	.31
2	.00345	.010
3	.00159	.049
4	.0155	.486
5	.00818	.256
Total	.03183	1.00

D. Selection

For selection the population we have used the proportionate selection method. This kind of selection methodology is similar to the roulette wheel selection methodology in which selection is done on the basis of slice of wheel. The proportionate selection methodology uses the fitness value for the selection of individuals.

Considering the equation based on the Table I

$$p_i = F_i/\Sigma F_i \quad (5)$$

p_i is the probability of each string to be selected

F is the fitness value of each chromosome within the string

ΣF_i is the sum of all the fitness values of each individual within the string.

So all out of population strings, the ones with lesser fitness value that is approaching to zero are selected as the parents.

E. Slave Nodes

Hence slave nodes are selected by master scheduler for finding the optimized set of tasks that can be assigned to under loaded or not loaded nodes over the grid. In our case we have consider N2, N3 and N5 as the slave nodes for the completion of overall genetic algorithm. The resultant set of parent strings that are generated as a result of fitness and selection steps are assigned to now processor N2 for application of the mutation and crossover procedures.

F. Cross over and Mutation

Crossover and mutation are the two important steps to be followed after the completion of fitness function and selection procedures based on genetic algorithm. Parallel genetic algorithm breaks down the procedure of GA into two parts and hence the crossover and mutation procedure is performed by slave nodes selected by master node. These slave nodes are selected on the basis of following conditions.

Total work load on each node →0

Hence all those nodes whose total work load approaches to zero can be selected as a slave node. Suppose in our case there are three nodes N2, N3 and N5 that are selected for performing the crossover and mutation procedures.

We have used the single arithmetic crossover for implementing the rest of PGA. It works like thisSuppose we have two parents: and $\langle y_1, \ldots, y_n \rangle$ and $\langle z_1, \ldots, z_n \rangle$. We will select gene ($k$) at random after that some selected point and mix values, then the child become

$$(y_1, \ldots, y_k, \alpha.z_k + (1-\alpha).y_k, \ldots, y_n) \quad (6)$$

Suppose we have these set of parents generated from master node and sent to slave node for cross over.

Suppose we have α=4

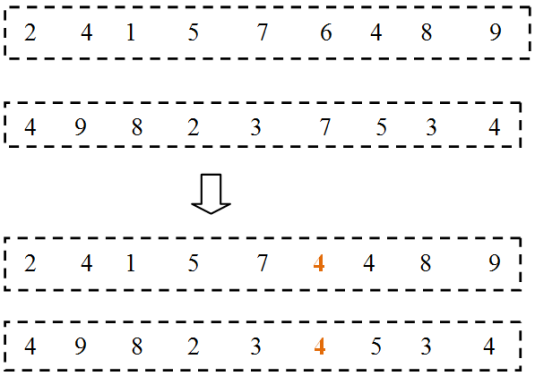

Figure 3. Single arithmetic crossover

Fig. 3 shows the results of arithmetic crossover. Now the mutation that is also performed by the slave node. Finally the mutation enables the genetic algorithm to explore the search space in a manner that is not accessible by crossover. We have applied simple mutation technique named as random mutation. In this random mutation we select the random element and exchange it. The child we get from crossover will become like this after mutation.

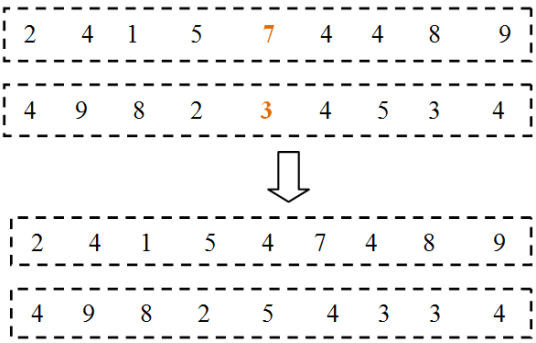

Figure 4. Random mutation

Fig. 4 presents the results of mutation. So after applying the crossover and mutation results are sending to master node so that it can make decision about

assignment of tasks to particular processor for processing. We will run PGA for almost ten times and do not wait for its convergence.

V. Results and Discussions

Hence our proposed strategy defines the results on the basis of two aspects. One is the selection of optimized set of jobs or assigning to processing nodes and other is the selection of the node which is neither under loaded nor over loaded throughout the whole process.

For presenting the efficiency of our proposed algorithm we have consider different factors and effects of our strategy on these factors. These proposed factors are time based and are total completion time, waiting time and total execution time for particular set of tasks. Hence all these time actors are dramatically enhanced when proposed strategy is implemented on them. The graphical representation of these results shows that how overall efficiency of process increases.

First we consider the effect of parallel genetic algorithm on the waiting time of tasks.

Hence our proposed strategy defines the results on the basis of two aspects. One is the selection of optimized set of jobs or assigning to processing nodes and other is the selection of the node which is neither under loaded nor over loaded throughout the whole process.

For presenting the efficiency of our proposed algorithm we have consider different factors and effects of our strategy on these factors. These proposed factors are time based and are total completion time, waiting time and total execution time for particular set of tasks. Hence all these time actors are dramatically enhanced when proposed strategy is implemented on them. The graphical representation of these results shows that how overall efficiency of process increases.

First we consider the effect of parallel genetic algorithm on the waiting time of tasks.

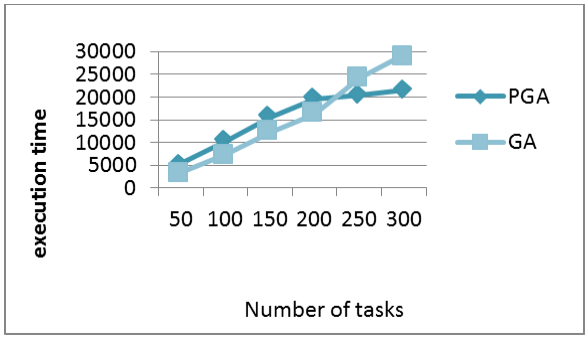

Figure 5. Effects of genetic algorithm and parallel genetic algorithm on waiting time of tasks

The above presented graph in Fig. 5 shows that the PGA based strategies give better results in terms of waiting time when numbers of tasks are continuously increasing.

The presented graph below in Fig. 6 shows that the PGA based strategies give better results in terms of execution when numbers of tasks are continuously increasing.

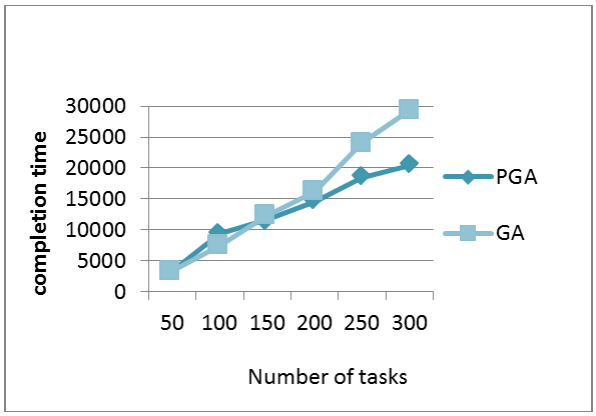

Figure 6. Effects of genetic algorithm and parallel genetic algorithm on completion time of tasks.

VI. Conclusion

Hence our algorithm that is based on the PGA based strategy is resourceful in manner that it optimizes the overall procedure by decreasing the total execution time, total completion time and total waiting time for all tasks. Hence as the time decreases the overall efficiency of the parallel genetic algorithm increases. So in case of grid computing the implementation of parallel genetic algorithm gives optimal results in selection of tasks to be assigned to decreases the overall execution time for tasks in case where the numbers of tasks are continuously increasing.

The selection of minimum loaded node in grid computing overall increases the efficiency of proposed algorithm. So in case of grid computing where the resources have to exploit and nodes are quite heterogeneous this strategy gives the optimal results and of great use. Both priorities and sizes of tasks are of uneven length. This algorithm is implemented on the local grid created during this research work.java based software gridsim is use to implement on the grid infrastructure for checking the load balancing issues and results in terms of time. As a future work some new techniques for load balancing, like Ant colony optimization and particle swarm optimization will be taken into consideration.

Acknowledgment

I would like to acknowledge my professor Dr. Daud Awan and Dr. Stauseef Ur Rehman for their support and help for completion of this task. This work is supported in part by the Preston University, Department of Computer Science.

References

[1] N. Y. Yen and S. Y. F. Kuo, "An intergrated approach for internet resources mining and searching," *Journal of Convergence*, vol. 3, pp. 37-44, 2012.

[2] E. Pyshkin and A. Kuznetsov, "Approaches for web search user interfaces: How to improve the search quality for various types of information," *Journal of Convergence*, vol. 1, pp. 1-8, 2010.

[3] A. P. A. Ling and M. Masao, "Selection of model in developing information security criteria for smart grid security system," *Journal of Convergence*, vol. 20, pp. 39-46, 2011.
[4] O. Schmid and B. Hirsbrunner, "Middleware for distributed collaborative ad-hoc environments," in *Proc. IEEE International Conference on Pervasive Computing and Communications Workshops*, 2012, pp. 435-438.
[5] A. Aikebaier, T. Enokido, and M. Takizawa, "Trustworthy group making algorithm in distributed systems," *Human-Centric Computing and Information Sciences*, vol. 1, no. 1, pp. 1-15, 2011.
[6] J. Carretero, F. Xhafa, and A. Abraham, "Genetic algorithm based schedulers for grid computing systems," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 6, pp. 1-19, 2007.
[7] B. Yagoubi and Y. Slimani, "Task load balancing strategy for grid computing," *Journal of Computer Science*, vol. 3, no. 3, pp. 186-194, 2007.
[8] A. Y. Zomaya and Y. H. The, "Observations on using genetic algorithms for dynamic load-balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 9, pp. 899-911, 2001.
[9] M. Nikravan and M. H. Kashani, "A genetic algorithm for process scheduling in distributed operating systems considering load balancing," *Journal of Parallel and Distributed Computing*, vol. 70, no. 1, 2010.
[10] A. Jula and N. K. Naseri, "A hybrid genetic algorithm-gravitational attraction search algorithm (HYGAGA) to solve grid task scheduling problem," in *Proc. International Conference on Soft Computing and its Applications (ICSCA'2012)*, 2012.

N. Tabassam has completed her Masters of computer science from the COMSATS Institute of Information Technology and is currently working as programmer. She has many papers in the field of grid computing and software engineering. Her major field of study is computer networks.

D. Awan is PhD scientist and currently working as a dean in the Preston University Islamabad, Pakistan. He has been involved in the lot of activities related to education and is a renowned educationist. Previously Dr. Daud Awan has worked at different educational management positions.

H. Rehman is a student of Masters, faculty of science and technology, at National University of Malaysia. He is a bright student and his field of expertise is augmented reality. Previously he was working at IBM as Java programmer.