USE OF GENETIC ALGORITHM FOR BALANCING THE GRID LOAD

Saad Masood Butt[1], Moaz Masood Butt[2] and Azura Onn[3]

[12]Computer and Software Engineering Department, Bahria University Islamabad, Pakistan
saadmasoodbutt668@yahoo.com
moazbutt786@hotmail.com

[3]Department of Management and Human Resource, Universiti Tenaga National, Malaysia
azura@uniten.edu.my

ABSTRACT

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. 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. 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 algorithm 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.

KEYWORDS

Network Protocols, Grid Computing, Load Balancing, Distributed Computing, Bubble Sort, 2-element Insertion Sort

1. 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 an optimized results without any time delay and with maximum throughput we need enhanced algorithms across the networks. Hence this 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 presents a method of using the computational resources optimally within an association involving different computing resources. Its 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 (Neil Y, 2012).

Multiple sharing resources are accumulated across the grid network. These resources involves 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 [1]. 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. 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 from long time for finding optimized solutions. These algorithms incorporates the mechanism inspired from biological evolution [2]. 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, and dual species genetic algorithm

2. RELATED WORK

There is 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 [3].

Distributed systems 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 [4]. 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 has been focused and the majority of genetic algorithm operators for each of these schemes are implemented and logically studied. 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 neighbourhood property and focusing on the tree based techniques generated optimized results in for managing load balancing across the grid networks [5]. 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 capitalize the processors efficiency and utilization [6]. 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 [7].

[8] proposed approach to solves 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 o HYGAGA and genetic algorithm emphasizes major improvement in the recital of search algorithm[9][10].

2.1. Proposed Methodology

Grid Model:

We have taken in account the grid information system model for solving the issue o 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 o task. Let us consider that set of task come to the scheduler in GIS model of grid. These tasks are selected on the basis o criteria we define based on the total completion time of tasks. We select those tasks first whose completion time is approaching to zero

$$T_{ct} \rightarrow 0$$

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 focuses 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 as slave schedulers. These slave schedulers are selected on the basis of state information present at the master scheduler. Let us consider the figure 1 for the illustration of this kind o grid model.
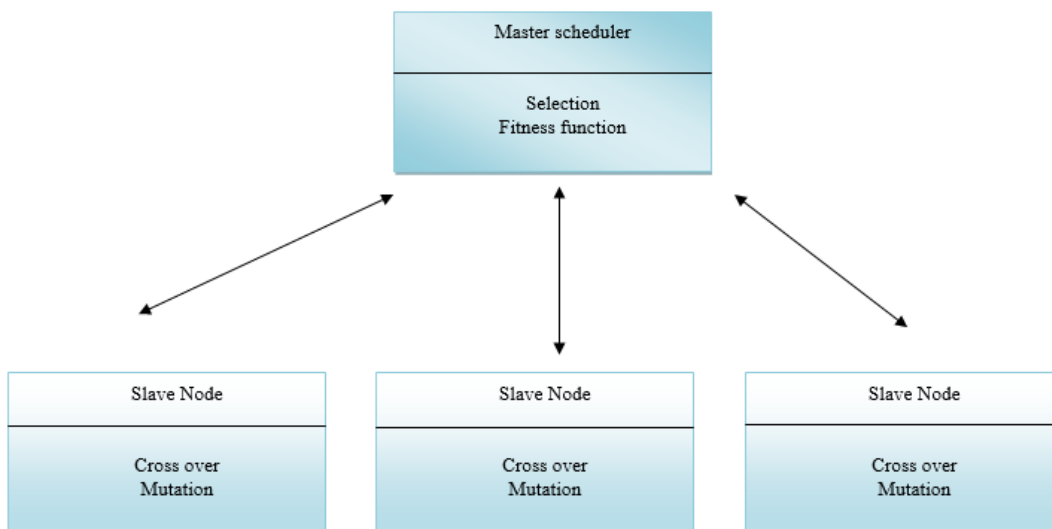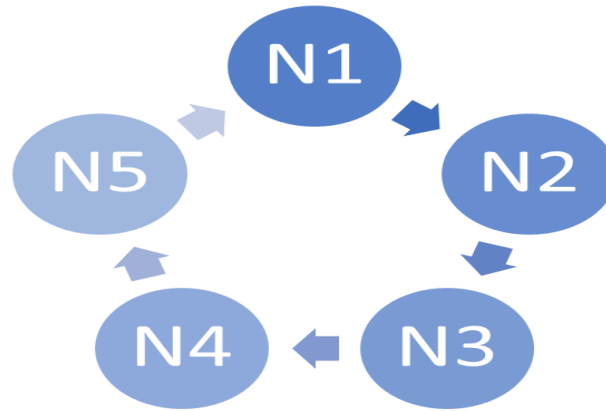
Figure 1. Master and slave nodes in scheduler

Figure 2: Group of nodes connected to grid

Suppose 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.

2.2. SELECTION OF OPTIMIZED SET O JOBS FOR ASSIGNING TO RESOURCES

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.

Initial Population

A Initial population based on the completion time of each task:
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.

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 makes pan. 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.

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

- 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

- LS = 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. is 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.

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

- 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 are based on the total completion needed by each task. The result of fitness function is categorized in two ways.

$F(pop) > 0$

$F(pop) <= 0$

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 shows the selection of strings.

Table 1: Fitness values of selected strings

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

2.4. 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 1

pi = Fi / SFi

Where pi is the probability of each string to be selected
F is the fitness value of each chromosome within the string
Where SFi 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.

Crossover 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 $\rightarrow 0$

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 this
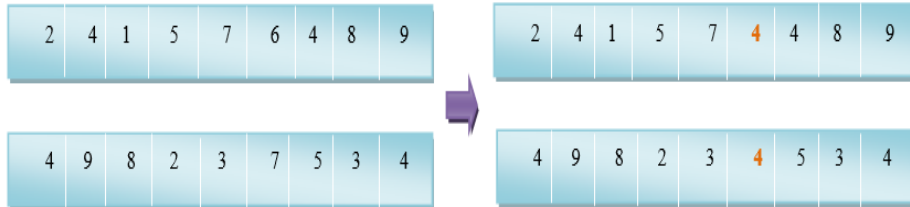Suppose we have two parents: and <y1,…,yn> and <z1,…,zn>
We will select gene (k) at random after that some selected point and mix values. Then the child become

$$\left\langle x_1, …, x_k, \alpha \cdot y_k + (1-\alpha) \cdot x_k, …, x_n \right\rangle$$

(y1,… , yk,α.zk +(1-α).yk,…., yn)

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

Suppose we have α=4

| 2 | 4 | 1 | 5 | 7 | 6 | 4 | 8 | 9 |

| 4 | 9 | 8 | 2 | 3 | 7 | 5 | 3 | 4 |

➤

| 2 | 4 | 1 | 5 | 7 | 4 | 4 | 8 | 9 |

| 4 | 9 | 8 | 2 | 3 | 4 | 5 | 3 | 4 |

After crossover the above mentioned results come out. Now the mutation that is also performed by the slave nodes is done after crossover operation

3. CONCLUSIONS

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.

REFERENCES

[1]   Yen, N. Y., & Kuo, S. Y. (2012). An intergrated approach for internet resources mining and searching. J. Converg, 3(2), 37-44.

[2]   Pyshkin, E., & Kuznetsov, A. (2010). Approaches for web search user interfaces. FTRA Journal of Convergence, 1(1).

[3]   Ling, A. P. A., & Masao, M. (2011, May). Selection of model in developing information security criteria on smart grid security system. In Parallel and Distributed Processing with Applications Workshops (ISPAW), 2011 Ninth IEEE International Symposium on (pp. 91-98). IEEE.

[4]   Schmid, O., & Hirsbrunner, B. (2012, March). Middleware for distributed collaborative ad-hoc environments. In Pervasive Computing and Communications Workshops (PERCOM Workshops),

        2012 IEEE International Conference on (pp. 435-438). IEEE.

[5]   Aikebaier, A., Enokido, T., & Takizawa, M. (2011). Trustworthy group making algorithm in distributed systems. Human-centric computing and information sciences, 1(1), 1-15.

[6]   Maheshwari, P. (1996, January). A dynamic load balancing algorithm for a heterogeneous computing environment. In System Sciences, 1996, Proceedings of the Twenty-Ninth Hawaii International Conference on, (Vol. 1, pp. 338-346). IEEE.

[7]   Li, Y., Yang, Y., Ma, M., & Zhou, L. (2009). A hybrid load balancing strategy of sequential tasks For grid computing environments. Future Generation Computer Systems, 25(8), 819-828.

[8]   Zomaya, A. Y., & Teh, Y. H. (2001). Observations on using genetic algorithms for dynamic load-balancing. Parallel and Distributed Systems, IEEE Transactions on, 12(9), 899-911.

[9]   Nikravan, M., & Kashani, M. H. (2007, June). A genetic algorithm for process scheduling in distributed operating systems considering load balancing. InProceedings 21st European Conference on Modelling and Simulation Ivan Zelinka, Zuzana Oplatkova, Alessandra Orsoni, ECMS.

[10]   Naseri, N. K., & Jula, A. (2012, August). A hybrid genetic algorithm-gravitational attraction search algorithm (HYGAGA) to solve grid task scheduling problem. In International Conference on Soft Computing and its Applications (ICSCA'2012).

Authors

Engineer. Saad Masood Butt received his BS (Software Engineering) degree from Bahria University Islamabad, Pakistan in 2008. He completed his MS (Software Engineering) degree in 2010 from Bahria University Islamabad, Pakistan. He is the recognized Engineer of Pakistan approved by Higher Education Commission and Pakistan Engineering Council (PEC). He has got more than 7 years international work experience and was associated with various organizations and universities in Pakistan and Malaysia.