International Journal of Engineering Research
Volume No.5, Issue No.12, pp : 901-904
ISSN:2319-6890)(online),2347-5013(print)
1 December 2016

Load Balancing Approach in Heterogeneous Distributed Systems - A Review

Palakodeti Venkata Sai Sudheer
CNIS, MVGR College of Engineering, Vizianagaram
Corresponding Email: venkat.sudheer9999@gmail.com

Abstract: In computer science scenario, Distributes Systems (DS) can be categorized by system transparency & resource mutation. There are various procedures for task scheduling in DS and one of the type is Load Balancing (LB). It is a way to spit the work between nodes for resource utilization. To keep the processor busy LB is a best way to improve system utilization and total response time. There are many methods in existence to optimize system utilization like Simulated Annealing (SA), Particle Swarm Optimization (PSO), FIFO, Shortest Processing Time (SPT), Largest Processing Time (LPT), Ant Colony Optimization (ACO). In this paper we focus on Genetic Algorithm to resolve our problem.

Keywords: Load Balancing, Genetic Algorithm, Distributed Systems, Optimisation.

I Introduction

Load Balancing's major functionality is to split the work among different nodes in a network. This is acquired by dividing work traffic among network interfaces among pool of nodes. So, for this a best algorithm is needed to attain transparency in dividing the work among systems and to maximize overall system performance. These days Heterogeneous Distributed systems are interconnected to achieve flexibility and resource sharing. They mainly comprises of the nodes with variance in their functions. It means each node have its own capabilities, processing speed and functionality. They allow high flexibility, reliability, modularity and high computational power.

Distributing the tasks among nodes is a tough work due to the inter processor communication overhead. Breaking a large process into small tasks and distributing among nodes also need a good communication about the functioning and potentiality of the node. Also, we must ensure that in a network no host should be idle or overloaded with process request. So, Load Balancing on the whole classified into two types they are Static Load Balancing (SLB), Dynamic Load Balancing (DLB) on the basis of work assigned to nodes at compile time or run time.

SLB is a method in which performance of each node is recorded at the beginning and work is assigned to each node. A master node is maintained to split the work based on the performance of the nodes. The task assigned to a particular node cannot be changed as it is a static process. DLB is a method quite opposite to SLB where a master node divides the work load among the nodes at runtime. A queue is maintained at the master node to manage dynamically the request and process allocation to the nodes.

II Related Work

LB is an emerging and most important topic of research in DS. A model view based clustering technique was proposed by Rao and Kumari [i]. An adaptive strategy of load balancing using the Quality of Service (QoS) parameters was studied in [ii] Sliding window method was employed for load balancing of a distributed cluster system that can adaptively learn and balance the load balancer at its master node [iii]. Different Schemes for DLB for Heterogeneous Distributed Systems are clearly explained in [iv].

Study on Distributed Evolutionary Algorithms (DEA) [v] [vi] provides a survey for optimization techniques for solving complex problems. Evolutionary Algorithm (EA) such as Genetic Algorithm (GA), PSO, SA, ACO, have their own scope in solving combinatorial NP hard models. Each method has its own advantage and disadvantages over one another. For example ACO was used in feature selection of intrusion detection and firewall optimization in [vii], [viii], and [ix].

III Performance Analysis of Load Balancing

In SLB and DLB a master node need certain algorithms to perform its action perfectly. They mainly work on the principle that in which situation workload is assigned whether during run time or compile time [x]. When SLB and DLB are compared SLB algorithms are more stable when compared to DLB. It is because SLB algorithms are ease to predict the behavior due to its static nature where as DLB are more faster and flexible when compared to DLB.

Static Load Balancing Algorithm

In SLB, the first major step is to collect all the relevant information regarding the nodes and its computational factors. But in reality whole information about the system may not be readily available at mater node to schedule the tasks in such case SLB is broadly classified into two categories Optimal SLB where all the information about state of the system is known and Sub-Optimal SLB where some of the computational information about the systems does not exist. Some of the SLB algorithms used are Round Robin and Randomized Algorithm, Central Manager Algorithm, Threshold Algorithm.

Dynamic Load Balancing Algorithms

To overcome the disadvantages with SLB and to achieve high computational speed and optimality we come with DLB. A buffer is maintained at the master node to allocate tasks dynamically upon host request. For a faster decision making of job transfer among nodes we must consider Load estimation policy, Processor transfer policy, State

ISSN:2319-6890)(online),2347-5013(print)
1 December 2016

information exchange policy, Priority assignment policy and Migration limiting policy. Some of the DLB algorithms are Central Queue Algorithm and Local Queue Algorithm.

Parameters

The performance of various load balancing algorithms is measured by Response time, Bandwidth, Resource utilization, Waiting time, Fault tolerance, Scalability, Performance, Overhead, Throughput, Stability, Centralized or Decentralized, Nature of load balancing algorithm etc.

IV. Optimization Techniques for Load Balancing

To find optimal value for a load balancer is an important task because it decides the working condition of the entire systems. With the available evolutionary algorithms we can find an optimal value. But, the problem is which algorithm suits best for a given problem? The answer varies according to the problem for SLB GA, SA, Graph theoretic approach, randomized algorithms, round robin algorithm etc. are the optimization techniques. For DLB analytic modeling and simulation, GA, heuristic algorithms are suitable.

SA is a probabilistic based approach for finding an optimal value [xi]. Each processors work are framed as a matrix representation which yields a local optima min and local optima max for each successive iteration.GA is a meta heuristic process which yields a optimal value based on the fitness function. It is more flexible then SA because it can be used for large search space and less selection output for the obtained fitness score [xii] [xiii]. In this paper, a review about Genetic Algorithm and its scope for optimization of System Utilization and Total Response Time (TFT) is clearly explained.

V. Genetic Algorithm Approach for Load Balancing

GA mimic the process of natural selection that generate natural solution to optimization and search problems. The concept of memorization helps to exploitation of past results with new one for a large search space.

GA requires:

- A genetic representation of the solution domain.

This is representation of the search space of GA. Each representation is solution to problem given. Each individual is represented as a finite length vector, variables, alphabets or binary alphabets {0,1}. In this genetic analogy each individual is considered to be a Chromosome and the variables are treated as Genes. Thus for every problem solution is obtained by adding a Fitness function.

- A fitness function to evaluate the solution domain.

GA aims to produce a better offspring from parents. This is achieved by using Fitness function. Based on this function optimal solution is obtained and called as Fitness Score .This score decides the best offspring (Solution). So, parents are mate with this fitness function to obtain a best fitness score and this steps are repeated for different iterations to obtain a best fitness score. Each successive iteration yields a best solution from the previous one. Newer generations are produced on an average, good or optimal from the previous one and yields a typical solution from the previous generations.

Based on the Natural selection after an initial population is randomly generated, the algorithm evolves through three operators. They are Selection, Crossover and Mutation. The following flowchart explains the implementation of Genetic Algorithm with its operators and parameters

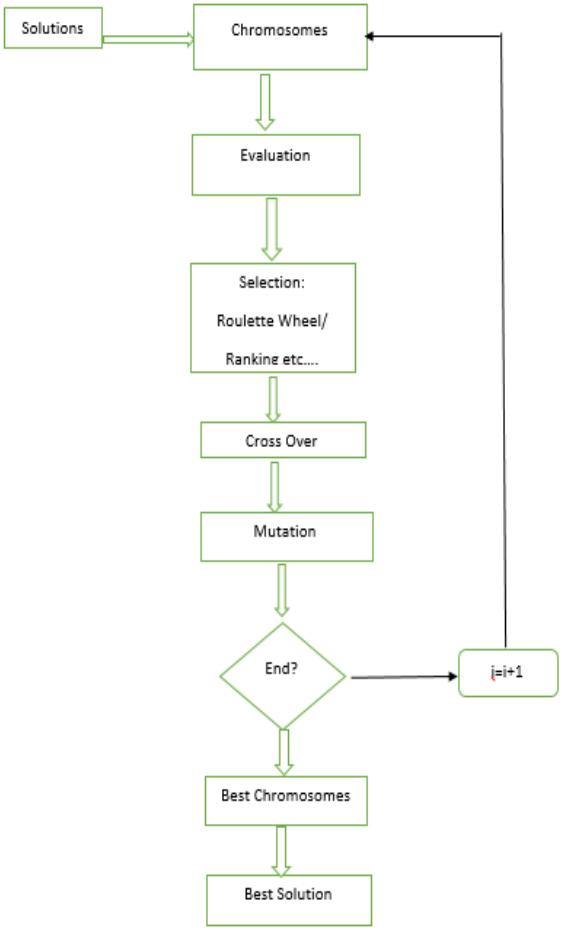

Figure.1 Flow Chart of Genetic Algorithm

VI. Evaluation and Results

For the current scenario how to encode the solution of the problem into GA input chromosomes is the key issues. Even though, we have various types of encoding strategies for the type of constraint optimization problems & combinatorial optimization problems we choose permutation encoding. In this process of finding sequence of tasks scheduling in load balancing, we treat each chromosome (input) as a sequence of tasks and each task is treated as a gene. Here we represent each chromosome is divided in tasks (T1, T2, T3…)

Initial Population

Initial process of GA is to start an initial random population to get first generation by utilizing a random generation function of chromosomes. For this we state number of

ISSN:2319-6890)(online),2347-5013(print)
1 December 2016

processors, size of the population and number of tasks so random chromosomes generate initial population.

Fitness Function

This is the critical stage in GA. This measures quality of the chromosome. This always depend on the problem and the variables involved in it. For a better optimized output the fitness function should be accurately designed. In our problem Pi is Processor N and i is total number of processors and TFT obtained. Thus a least obtained value of f is a best fitted chromosome.

$$f = \sum_{i=1}^{N} TFT - Pi$$

Selection Operator

The fitness function is a key step for the entire selection process which effects the performance of the GA. This selection operator in GA finalizes the selection of the inferior or the superior. The best selected chromosomes are elected through roulette wheel strategy.

Crossover Operator

Crossover operator selects two parents' chromosomes and randomly chooses their crossover points to produce better offspring from them. Here we consider single point crossover.

Mutation Operator

It randomly selects the tasks in the offspring chromosome and replaces them for the further search process for the best optimized output. In general swapping of tasks in a chromosome is done until a best output occurs.

All the simulations are performed in MATLAB. Here to show our output is best search strategy we compare our results with LPT, FIFO, SPT in parallel multi-processor system. Thus obtained results show that GA shows less total response time when compared to above algorithms. In the below graphs x axis plot number of processors and y axis plots Response time.

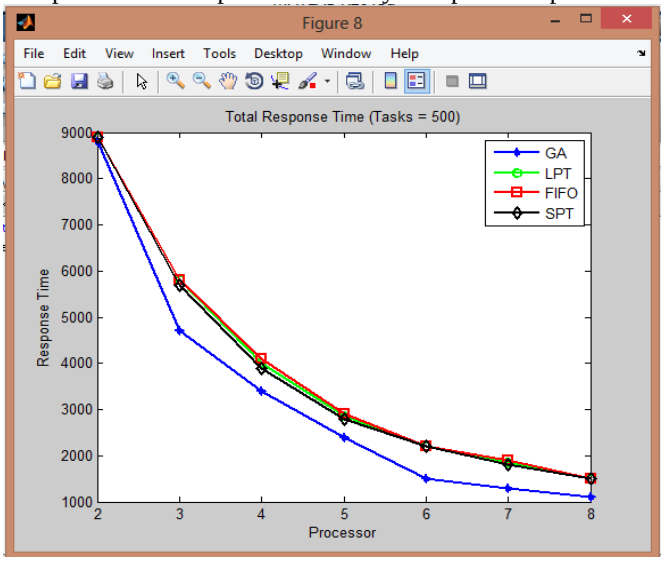

Figure.2 Total Response Time of GA

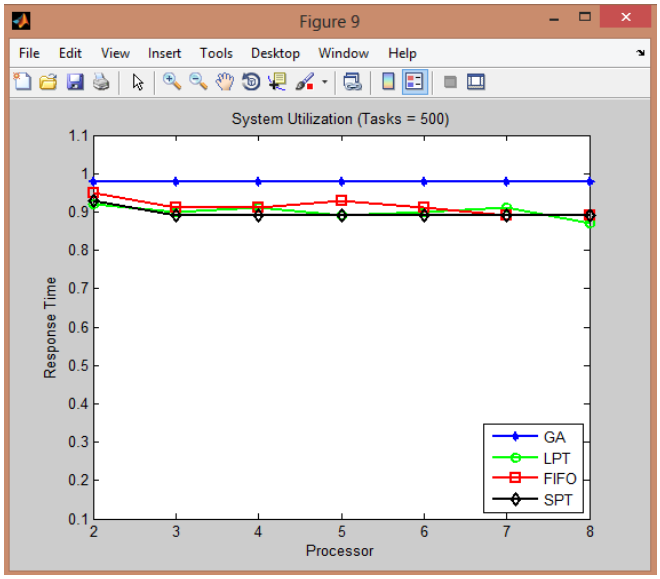

Figure.3 System Utilization of GA

VII. Conclusion

In the case of task scheduling & load balancing GA helps in reducing the TFT in heterogeneous parallel multi-processor systems. The proposed method yields more best results when compared to SPT, FIFO, LPT algorithms. The results obtained are based on a limited number of reproduction and genetic operators. Certainly, to gain better results we can work with Artificial Neural Networks or DEA algorithms. In addition to optimization along with security ACO can be applied.

References

i. Veerabhadra Rao Chandakanna and Valli Kumari Vatsavayi, "A model view controller based Self-Adjusting Clustering Framework," Journal of Systems and Software, vol. 89, pp. 193-206, 2014.

ii. Veerabhadra Rao Chandakanna and Valli Kumari Vatsavayi, "A QoS-aware self-correcting observation based load balancer," Journal of Systems and Software, vol. 115, pp. 111-129, 2016.

iii. Veerabhadra R Chandakanna and Valli K Vatsavayi, "A sliding window based Self-Learning and Adaptive Load Balancer," Journal of Network and Computer Applications, vol. 56, pp. 188-206, 2015.

iv. Bidhuatta Sahoo, Sudipta Mohapatra and S. K. Jena, "A Genetic Algorithm Based Dynamic Load Balancing Scheme for Hetrgenous Distributed Systems," in International Conference on Parallel and Distributed Processing techniques and Applications, Las Vegas, 2008.

v. Yue-Jiao Gong, Wei-Neng Chen, Zhi-Hui Zhan, Yun Li, Qingfu Zhang and J.-J. Li, "Distributed evolutionary algorithms and their models: A survey of the state-of-the-art," Elsevier, vol. 34, pp. 286-300, 2015.

vi. I.Zelinka, "A survey on evolutionary algorithms dynamics and its complexity-Mutual relations, past, present, future," Elsevier, vol. 25, pp. 2-14, 2015.

ISSN:2319-6890)(online),2347-5013(print)
1 December 2016

vii. Ravi Kiran Varma P, Valli Kumari V and Srinivas Kumar S, "A Novel Rough Set Attribute Reduction based on Rough Sets and Ant Colony Optimization," International Journal Intelligent Systems Technologies and Applications, vol. 14, no. 3/4, pp. 330-353, 2015.

viii. Ravi Kiran Varma P, Valli Kumari V and Srinivas Kumar S, "Ant Colony Optimization-based Firewall Anomaly Mitigation Engine," Springerplus, vol. 5, no. 1, pp. 1-32, 2016.

ix. Purnima Shah and S. Shah, "Load Balancing in Distributed System using Genetic Algorithm," IJCA, Ghandinagar, 2010.

x. Zongquin Fan, Hing Shen, Yanbo Wu and Y. Li, "Simulated Anealing load balancing for resource allocation in Cloud enviornment," in International Conference on PDCAT, 2013.

xi. M. Effatparvar and M. S. Garshasbi, "A Genetic Algorithm for Static Load Balancing in Parallel Hetrogenous Systems," in International Conference in Innovation, Managment and technology research , Malaysia, 2013.

xii. Anuradha Sharma and S. Verma, "A survey on Load Balancing algorithm in Grid Computing," International Journal of Advanced Engineering reaserach and Studies, 2015.