



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Load Balancing With Genetic Algorithm

Kulvinder Singh Handa¹, Chetna Kukreja²,

^{1,2} Department of Computer Science & Engineering, University Institute of Engineering and Technology
Kurukshetra University, Kurukshetra, Haryana, INDIA.

E-mail (s): kshanda@rediffmail.com, chetna_kukreja2000@yahoo.com,

Abstract: In load balancing, each processor is assigned work proportional to its performance so that execution time of the program can be minimized. The gap between lightly loaded processor and heavily loaded processor is filled by load balancing. Static load balancing and dynamic load balancing are two types of load balancing. For decreasing unnecessary request messages and for increasing acceptance rate, genetic algorithm is used with load balancing. Population of solutions are maintained iteratively by genetic algorithm in which each solution represents a chromosome. Basic operation of genetic algorithm here is to determine a destination processor that can receive a task.

Keywords: Load balancing, genetic algorithm

I. Introduction

Basic operation of load balancing algorithm is to equally spread the load on processors. Load balancing is a technique to enhance resources, utilizing parallelism, exploiting throughput improvisation, and to cut response time through an appropriate distribution of the application [1]. Load balancing is essential for maximizing processor's utilization and minimizing the mean response time. For achieving efficient resource utilization, rapid response time and higher throughput, load balancing is a key factor where resources are distributed and heterogeneous servers are present in the system. The difference between time when job arrives to the system and time when job gets processed by system is called Mean response time (MRT), is improved by load balancing. By reconfiguration of network, in load balancing the load is migrated from the node which is heavily loaded to the node which is lightly loaded which will reschedule load and network reliability and peak demands are smoothed out with the help of reconfiguration. Some load balancing techniques detect load imbalance by periodically checking the average idle time of workstation, in which until the period ends, load balancing can't be performed whether at the mid of the period imbalance occurs.

Munetomo *et al.* [2] presented a dynamic load balancing with genetic algorithm for stochastic environments in distributed systems. Zaki *et al.* [3] examine the behavior of different load balancing strategies: global v/s local and centralized v/s distributed; and showed that different strategies are best for different applications under varying system parameters and programs. Maeng *et al.* [4] proposed a new threshold method for load balancing for workstations which process the jobs with relatively long execution time and this method decides a proper time for performing load balancing and after detection of load imbalance performs balancing action. Seo *et al.* [5] presented dynamic load balancing algorithm for hypercube multicomputer with faulty nodes. W. A. Greene [6] presented a relatively low cost genetic algorithm scheduling routine which finds well-balanced schedules.

Load balancing algorithm is generally characterized into two categories:

- a) Static Load Balancing Algorithm: Prior information of task is used to allocate all tasks to processors before their execution. In static load balancing algorithm, the task is allocated to the processor based on either the load at the time nodes are allocated to some task, or based on an average load of system. When estimation are made about resource requirements, at compile time decisions are made about load balancing. Static load balancing algorithm is mainly adopted because of its simplicity in terms of both implementation as well as overhead. There is no need to monitor workstation constantly for performance statistics in static load balancing scheme.
- b) Dynamic load balancing scheme: Transfer of task occurs during execution from heavily loaded processor to lightly loaded one. Changes are made at run time for distributing the work in which information about load is used that is current or recent for making distribution decision in dynamic load balancing scheme. Dynamic load balancing algorithms offer the possibility of improving load distribution at the expense of additional communication and computation overheads [7]. Significant improvement is provided by dynamic load balancing algorithm as compared to static load balancing algorithms. But additional cost of adopting this algorithm is collecting and maintaining load information and it is important to have these overheads within a reasonable limit.

II. GENETIC ALGORITHM

Genetic algorithms were introduced by John Holland after the invention of evolutionary computing in 1960s by Rechenberg. Genetic algorithms, subclass of evolutionary algorithms, are the procedures that are used to find out optimized solution to search problems. Selection, mutation, inheritance, recombination, and crossover are some biological inspired techniques which are used in genetic algorithm. Initial and feasible solutions in form of chromosomes are generated by genetic algorithm. Genetic Algorithm is adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic [8]. Genetic algorithm mimics the process of natural selection which is used as a problem solving technique. Individuals, which can biologically reproduce to a problem, are selected as potential solutions. At first level of genetic algorithm, individuals are randomly selected from the population to see the closeness between them to solve the problem. Individuals which are more fit than other members of the population, are allow to copulate to produce the next generation is the better solution of the problem. Until an optimized solution of the problem is not found, this process is repeated again and again. GA is one kind of global optimization probability search algorithm based on theory of evolution by Darwin and theory of heredity by Mendel which can simulate life evolution mechanism in biosphere and realize optimization of specific goals in manual system [9].

- a) **Search Space:** Binary string or array of other types is used to represent elements in search space. Various forms, such as human based genetic algorithms (HBGA), in genetic algorithm requires selection or evolution of solution candidate. Best solution is picked among all for solving problem. The search space is the collection (set) of all solutions in which best solution which we desire, resides.
- b) **Chromosome:** Every organism contains same set of chromosomes called a cell. Chromosomes, whose basic informational unit is genes, serve as a model for whole organism and are a string of DNA. A particular type of protein is encoded in each gene which encodes the trait, for example color of eyes. The position of chromosomes in a gene is called locus. And the set of all chromosomes (genetic material) is known as genome.
- c) **Reproduction:** Firstly crossover occurs during reproduction in which new chromosome is formed by combining the genes of parents. After that mutation takes place in which DNA elements are bit changed which is mostly caused by copying error of genes from parents. Then fitness is measured by survival of an organism.
 - (i) **Crossover:** In this type of operation, one or more pair of individuals is randomly picked as parents and swap segments of parents randomly. For example, **110110011** and **100110001** => **110110001** and **100110011**.
 - (ii) **Mutation:** Mutation increases randomness in the string representing an individual by changing one or more digits in the string. For example, the individual 4-2-6 after mutation may be changed to 4-3-6 or 4-2-3, giving two new offspring.
- d) **Evolution:** Many human inventions are inspired by nature artificial neural network and genetic algorithms are the examples. A subclass of artificial intelligence is genetic algorithm which is inspired by the way the living organism survives in nature.
- e) **Natural Selection:** Individuals which can survive for longer period of time have better survive trails in nature and there is better chance to produce offspring with its genetic material. As time grows, there remains the population in which almost genes are of superior individuals and less amount of inferior individuals. This criterion is termed as natural selection.

III. LOAD BALANCING WITH GENETIC ALGORITHM

For balancing work load on machines, machine allocations of each work center to lots of jobs, is search by genetic algorithm which is an iterative search method and used in many optimization problems. Solutions are represented by chromosomes in genetic algorithm and population of solutions is maintained iteratively. Value of objective function which is determined by the solution is used to evaluate the value of fitness of solution. Chance of survival greatly depends upon the fitness value of solution. . By applying genetic operator such as crossover and mutation, solutions are changed from generation to generation.

Genetic algorithm is a kind of random search number which is not merely a one-point search but also incorporates a multipoint search feature. Finding a suitable receiver which is ready to receive additional task, when a system becomes heavily loaded, is difficult because mostly processors are heavily loaded so many request and reject messages are send back and forth, and lots of time is also wasted before execution which causes reduction in system throughput and CPU utilization.

To get rid from this limitation, genetic algorithm is used for determining a destination processor that can receive a task. Genetic algorithm helps in finding the number of request message issued before accepting the task. Gene in the string determines the number of processors to which request message are sent off. There are various factors which are needed in load balancing with genetic algorithm such as load measure, fitness function, coding method and algorithm.

- a) Load Measure: basically 3-level measure scheme is used to represent the load state of processor. Lightly loaded processor, normally loaded processor, and heavily loaded processor. Two design parameters that are taken are T_{up} and T_{low} which are called upper and lower threshold respectively. Threshold policy is used as transfer policy which is used to made decisions based on the CPU queue length. If the CPU queue length exceeds T_{up} by originating the new task at a node then that node is identified as the sender and the node, whose CPU queue length will not cause to exceed T_{low} on acquisition of a task, is identified as the receiver.
- b) Coding Methods: There are many kinds of coding methods, binary encoding, character and real value encoding, tree encoding are some examples which are used on population, onto which genetic operations are applied. Binary coded vector $\langle v_0, v_1, \dots, v_{n-1} \rangle$, can be used to define a string in population, which represent a set of processors to which request messages are sent off. The value of $V_i=1$ if request message is sent off to processor P_i (where $0 \leq i \leq n-1$, n is the total number of processors) otherwise $V_i=0$. Every string contains its own fitness value and we select a string whose probability is proportional to its fitness value and transfer the request message to the processors which is indicated by the string. Depending upon the queue length each processor send accept message or reject message to the sender upon receiving the request message from sender. If more than two accept messages are returned then one is selected at random.
- c) Fitness Function: following formula will evaluate fitness [10]:

$$F = \frac{1}{\alpha \times TMP + \beta \times TMT + \lambda \times TTP}$$

Where α , β and γ are the weights of parameters TMP, TMT, and TTP respectively. The summation of the processing times for request messages to be transferred is called as TMP (Total Message Processing Time). TMP is defined by the following formula

$$TMP = \sum_{K \in X} (ReMN \times time - unit)$$

(where, $x=\{i | v_i=1 \text{ for } 0 \leq i \leq n-1\}$) [10]

Number of message to be transferred is defined by ReMN. By ReMN the string which has fewer numbers of messages to be transferred is selected. TMT(Total Message Transfer time) means the summation of each message transfer times(EMTT) from the sender to processors corresponding to bits set '1' in selected string. The objective of this parameter is to select a string with the shortest distance eventually. So, we define the TMT as the following formula.

$$TMT = \sum EMTT_k$$

(where $x=\{i | v_i=1 \text{ for } 0 \leq i \leq n-1\}$) [10]

TTP (Total Task Processing time) is the summation of the times needed to perform a task at each processor corresponding to bits set '1' in selected string. This parameter is defined by the following formula. The objective of this parameter is to select a string with the fewest loads. Load in parameter TTP is the volumes of CPU queue length in the processor.

$$TTP = \sum_{K \in X} (Load_k)$$

(where $x=\{i | v_i=1 \text{ for } 0 \leq i \leq n-1\}$) [10]

To achieve largest fitness value TMP, TMT, TTP must have values small as possible as.

IV. CONCLUSION

This paper deals with the problem of load balancing with the help of genetic algorithm. Genetic algorithm was used to allocate loads to processors, with the objective to minimize the difference between the loads of each processor. With the help of genetic algorithm suitable candidate receiver is decided to which request message

should be sent off. In terms of achieving the goals of maximum processor utilization and minimum total completion time, genetic based load balancing algorithm performs really well.

V. References

- [1] M. Bozyigita, "History-driven dynamic load balancing for recurring applications on network of workstations", Systems and software, vol. 2, 2000, pp. 61-72.
- [2] M. Munetomo, Y. Takai and Y. Sato, "A Stochastic Genetic Algorithm for Dynamic Load Balancing in Distributed Systems", IEEE, 1995, pp. 3795-3799.
- [3] M. J. Zaki, W. Li and S. Parthasarathy, "Customized Dynamic Load Balancing for a Network of Workstations", IEEE, 1996, pp. 282-291
- [4] H. s. Maeng, H. S. Lee, T. D. Han, S. B. Yang and S. D. Kim, "Dynamic Load Balancing of Iterative Data Parallel Problems on a Workstation Clustering", IEEE, 1997, pp. 563-567.
- [5] J. Seo, S. Lee and J kim, "Synchronous Load balancing in Hypercube Multicomputers with Faulty Nodes", IEEE, 1997, pp. 414-421.
- [6] W. A. Greene, "Dynamic Load Balancing via a Genetic Algorithm", IEEE, 2001, pp. 121-128, doi: 10.1109/ICTAI.2001.974456.
- [7] Zeng Z. and veeravalli B., "Rate based and queue based dynamic load balancing algorithm in distributed systems", IEEE, 2004, pp.349-356.
- [8] Y. F. Dong, J. H. Gu, N. N. Li, X. D. Hou, and W. L. Yan, "Combination of genetic algorithm and ant colony algorithm for distribution network planning", IEEE, 2007, pp. 999-1002.
- [9] S. Wang and D. Gu, "Intelligence Control System and its Application", Beijing: China Machine Press, 2005.
- [10] Lee S. and Hwang C., "A dynamic load balancing approach using genetic algorithm in distributed systems", IEEE, 1998, pp. 639-644.