IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

A Study of Genetic and Honey Bee Algorithms for Load Balancing in Distributed Systems

Dr. G. Suvarna Kumar[1], G. Venkata Lakshmi[2]

Associate Professor, Computer Science Engineering Department, MVGR College of Engineering, Vizianagaram, Andhra Pradesh[1]

Department of Computer Networks and Information Security, MVGR College of Engineering, Vizianagaram, Andhra Pradesh[2]

Abstract: Parallel processing in Distributed Computing refers to the concept of running several tasks simultaneously on different processors. Load balancing and scheduling are very important tasks to optimally utilize the available resources and processor utilization. These problems are NP-Complete. In this paper, we introduce two methods which are genetic algorithms and Honey bee algorithms for scheduling and load balancing in parallel heterogeneous multi-processor systems. Genetic algorithm is a meta-heuristic algorithm which provides optimization by performing genetic operations on the running tasks and Honey bee algorithm is a Bio-inspired algorithm which provides optimization based on foraging behaviour of honey bees. Finally these two types of algorithms provides better optimization than the normal CPU scheduling algorithms, it is studied by comparing the results of meta-heuristic and bio-inspired algorithms with CPU scheduling algorithms, such as Longest Processing Time (LPT) and Shortest Processing Time (SPT) [2]. The results of simulations indicate meta-heuristic and bio-inspired algorithms for scheduling and load balancing provides better total response time and system utilization. Simulations results indicate Genetic Algorithm and Honey bee algorithm reduces overall response time and also increases resource utilization in load balancing of distributed systems.

Keywords: Heterogeneous multiprocessor systems, Meta-heuristic algorithms, Bio-inspired algorithms, Genetic algorithms, Honey bee algorithm, Waggle dance, Virtual machine, System utilization, Total response time.

1. INTRODUCTION

Distributed Systems are characterized by their structure. A typical distributed system will consist of large number of interacting devices which are computers, hard drivers, printers and other network devices [4] each runs on their own programs and they are communicated with message passing. Sometimes those are affected by receiving messages, or observing shared-memory updates or the states of other devices (Yorozu, et al, 1987). Examples of distributed computing systems ranges from simple systems in which a single client talks to a single server to huge network like the Internet communicates with many clients.

As Distributed Systems get larger, it becomes harder and harder to predict or even understand their behaviour. Part of the reason for this is that we are as programmers have not yet developed the standardized tools for managing complexity of large networks as are found in sequential programming. For this reason, the large distributed systems bring with them large amounts of non-determinism and unpredictable events like delays in messages [5] and message with modifications, the sudden failure of components or in other cases of the actions of faulty machines opposed to the goals of the system as a whole. Because of the unpredictability in the large distributed systems, it can be very difficult to test or simulate them efficiently and effectively. Therefore, there is a need for tools that allows us to prove the properties of distributed systems that will provide efficient communication and working in distributed environment. The first task of any theory of distributed systems is defining a mathematical structure that describes all the relevant properties of entire distributed system.

In Distributed Computing, multiple processors work together to solve a problem. For example an online shopping website, number of processors called Virtual machines (VM's) to process the requests from the user. Load must be balanced among the processors. This application requires many processors to satisfy all requests from the user which is very economical and time consuming.
But it is very difficult to manage that requests with available number of resources with in time. In Distributed computer networking, load balancing is a technique to divide the work between two or more computers, network devices, CPUs, hard disks, or other resources, in order to get optimum resource utilization, increase the throughput and increase the response time.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

2 METHODOLOGY

Load balancing in distributed computing is managed by meta-heuristic and bio-inspired algorithm which provides better optimization with minimum processing time, maximum resource utilization and increases throughput. In meta-heuristic, genetic algorithm is used to balance and schedule the tasks, in bio-inspired Honey bee algorithm is used. Scheduling algorithms which are LPT and SPT assigns the tasks to virtual machines or computers only based on the task length. They doesn't consider the processing time of virtual machines. But, the genetic [1] and honeybee algorithms consider both the cases and assign tasks to respective VM's. Therefore, Load can be balanced in an efficient manner with available number of resources.

2.1 Genetic Algorithm

Genetic Algorithm (GA) is a meta-heuristic algorithm used to solve the classification problems, search problems and optimization problems. In my project, Genetic algorithm is used for solving load balancing problems. In load balancing, multiple processors execute different tasks at the same time. Therefore in order to utilize the resources effectively GA is used. It provides optimised solutions among the number of possible solutions. Each solution in the GA is referred as chromosome. The number of possible solutions is known as population size. A fitness function is used to evaluate the fitness of each individual. Fitness function is a function which is designed based on the requirements of user. To balance the load, the fitness function is in the form of processing time so that load can be balanced [10]. GA provides optimized solution by performing some operations which are selection, crossover and mutation [9] with some parameters like crossover probability rate, mutation probability rate and fitness function.

2.2 Honey Bee algorithm

A honey bee Algorithm is a bio-inspired algorithm. The load balancing in distributed computing is inspired by colony of honey bees with foraging behaviour. The colony of bees is search for the food sources [7] like flowers and patches etc. They harvest the nectar or pollen from the food sources. The bees find the place where they can get enough amounts of flowers to harvest the nectar, the bees check for the quality of nectar. When bees return to its hive, collect the amount of nectar they harvest and perform waggle dance [7] on the floor. Waggle dance is a representation of results of search food. It shows they found a beneficial quality food source. The duration of dance indicates the bee rating for the food source. Therefore, more bees are recruited to best rated flowers and remaining bees are recruited to lower rated patches [6]. The bees harvest the nectar until they return to its hive. The bees again perform waggle dance to get more profitable. This process repeats until the bees found beneficial flower patches. This foraging behaviour of bees is inspired by load balancing in distributed computing to provide better optimization with more system utilization, less response time and more utilization of resources.

Results of the simulations indicate Genetic Algorithm and honey bee algorithm reduces total response time in comparison with Longest Processing Time, Shortest Processing Time. In addition, the system utilization increases when we using Genetic and Honey bee Algorithms for Scheduling in parallel heterogeneous multiprocessor systems.

3. IMPLEMENTATION OF GENETIC ALGORITHM FOR LOAD BALANCING

3.1 Algorithm for Genetic Algorithm

1. Determine the number of chromosomes, number of iterations and crossover and mutation probability rates.
2. Initialize the values of genes in each chromosome.
3. Evaluate the fitness function of the chromosomes from the objective function developed by the user.
4. Chromosome selection from the population.
5. Perform crossover
6. Perform mutation.
7. New chromosomes are generated from the crossover and mutation operations.
8. Calculate the fitness function for these chromosomes. If the optimized solution is achieved, stop the process.

Otherwise repeat the process with the new generation of chromosomes.

The fitness function is evaluated first for possible number of solutions in the population. Then first generation starts with this fitness value and remaining genetic operations performed on this evaluated fitness values. The genetic operations involve selection in order to select best solution among the number of possible solutions in the population. Highest fittest values have more chance to get selected. There are different techniques used to perform selection operation. Another operation is crossover which is used to create a new solution by combining two solutions which are selected. In crossover, the values from one solution are replaced with other values in the next solution. For simplicity the values for two solutions are interchanged to make new solutions. The final operation is mutation, which generates a

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

new solution by replacing a single random value at a particular position of gene. Now, determine the fitness for new solutions which gives optimized results than the previous results. All these process comes under first generation of GA. Repeat this process until the optimized result obtained.

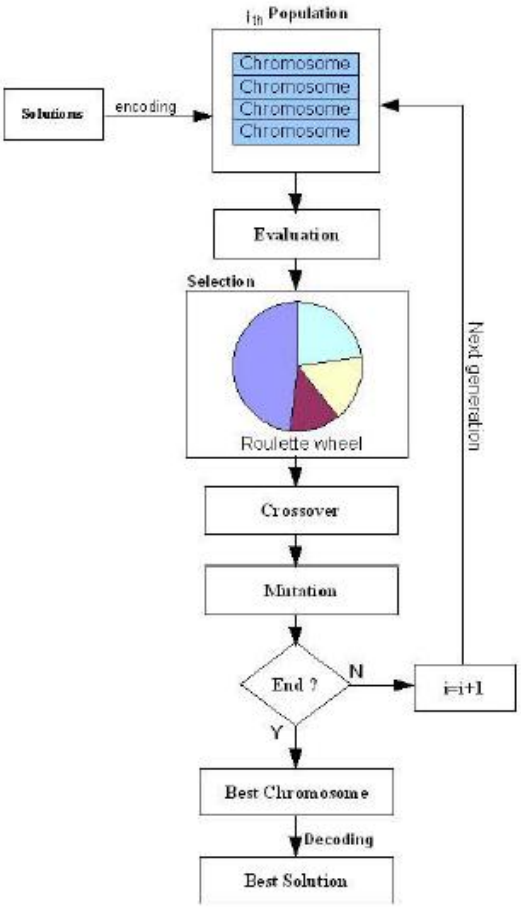

Fig3.1 flow chart for genetic algorithm

In the above figure, there are number of possible solutions which are known as chromosomes. Each chromosome undergoes the process of steps defined in the above Genetic algorithm.

3.2 Scheduling and Load balancing in Genetic Algorithm:

3.2.1 Encoding:

The main aim of this project is to find best sequence of tasks for load balancing and minimizing total response time and improve the system utilization by effectively utilize the resources in distributed systems. Thus, each chromosome is a sequence of variety of tasks which are T1, T2, T3, T4, T5, T6, T7 and T8 as shown below. Each task is considered as a gene in the chromosome. Therefore, the best way to encode chromosomes is permutation encoding. To explain how chromosomes are encoded, consider the 8 tasks, Ti represents the tasks. Table 3.1 shows two encoded chromosomes.

Table 3.1: Two encoded chromosomes

Ch 1	T1	T4	T8	T2	T7	T3	T6	T5
Ch2	T6	T1	T7	T3	T5	T8	T2	T4

3.2.2 Generate the Initial Population

First, GA generates an initial random population for enter into the first generation. For this, a random generation function, chromosomes must be in the population. In order to create an initial population, GA needs Information on the number of processors which are ready to execute the tasks, number of tasks and the size of the population. Random chromosomes generate the initial population.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

3.2.3 Fitness Function

The main part of GA is fitness function evaluation. The fitness function is defined by user based on the requirement. Fitness function measures the quality of the chromosomes. The fitness function is always designed from the problem of user. For this paper it is calculated for total response time and system utilization because the aim of this study is to maximize the total response time and system utilization on load balancing.

The fitness function is calculated according to the equation shown below:

$$F = \sum_{i=1}^{N} (TFT - Pi)$$

Where TFT is total response time obtained from chromosomes.
Pi is the number of Processors.

From the calculation of fitness function, the conclusion statement is low value of fitness function gives more optimized solution and high fitness values gives less optimization. Because total response times less means the system response is quickly.

3.2.4 Selection Operator

Selection means which chromosomes get selected among the population. It based on the fitness value evaluated in the first step. GA uses selection operator to select the superior and eliminate the inferior. Once fitness values have been evaluated for all chromosomes, we can select the chromosomes with highest probability through the selection techniques mentioned above. Selection selects the best chromosomes for next operator in the GA.

3.2.6 Crossover Operator

Select the type of crossover to be used which is either single point or multi point or uniform crossover. Then performs the crossover operation by randomly selects two parent chromosomes and randomly chooses the crossover to produce two offspring's (new chromosomes). Consider one point crossover for and perform crossover for the tasks which is shown in the below table 3.2:

Table 3.2: Single-point crossover

Ch 1	T1	T3	T4	T2	T8	T5	T6	T7
Ch 2	T5	T3	T1	T4	T7	T8	T6	T2
Offspring	T1	T3	T4	T2	T7	T8	T6	T2

3.2.7 Mutation Operator

A mutation operator works by randomly selecting two tasks and swapping them. First, select a processor, and then selects a task on that processor randomly. This is the first task to be swapped in the pair. Second, select another process randomly and select a task on that process. If the two selected tasks are the same task the search continuously for other tasks otherwise perform swapping on the two selected tasks which is shown in the below table 3.3.

Table 3.3 Mutation operator

Before	T1	T4	T8	T2	T7	T3	T6	T5
After	T1	T7	T8	T2	T4	T3	T6	T5

Now calculate the fitness function with the new chromosomes. If the result is optimized value for the given sequence of tasks, stop the generations of GA.

4 IMPLEMENTATION OF HONEY BEE ALGORITHM FOR LOAD BALANCING

4.1 Algorithm for Honey Bee Algorithm

1. Initialize the population with random solutions.
2. Evaluate the fitness function of the population.
3. Select the sites for neighbourhood search.
4. Recruit the bees for best selected sites and determine the fitness function.
5. Select the fittest bee from the each patch.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

6. Assign the remaining bees to other sites from random search and determine the fitness function.
7. If the result is optimized for the given problem, stop the iteration otherwise repeat the same procedure shown in the figure 4.1.

Foraging behaviour of honey bees is inspired by load balancing of task in the distributed [7] environment. Processors in the distributed environment are referred as Virtual machines (VM's). The tasks are search for VM's to execute their work. Once the task is assigned to VM and load is calculated [8]. The load of the VM is overloaded, the task is assigned to next VM otherwise it is executed by the current VM. Each VM has some threshold based on the computational efficiency of the VM in order to decide the amount of load each of the VM's can handle. Therefore, based on the threshold the tasks are executed on the VM's.

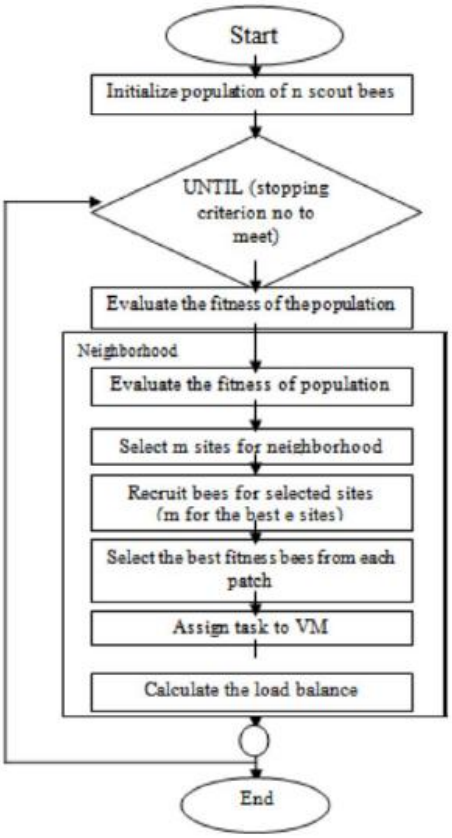

Fig 4.1: Flow chart for Honey bee implementation

5. EVALUATION AND SIMULATION RESULTS

The main aim of project is to reduce the total response time, increase the system utilization for load balancing in distributed environment. This project compares the results of CPU scheduling algorithms with Natural and Bio-inspired algorithms which are Genetic and Honey bee algorithms. All simulations are performed using MATLAB software.

The results are obtained for Honey Bee, LPT, SPT, GA and Honey bee algorithms. These are obtained by varying load on the processors that means result shows how the load balancing is performed while number of tasks are increases.

This project simulates the programs for number tasks given below:
When the number of tasks is50
When the number of tasks is 500

From the simulation results, the total response time for GA and Honey bee are lower than the normal CPU scheduling algorithms SPT, LPT. Therefore, it can observe that, as increasing the number of processors the system responds to the tasks quickly. The simulation results for 50 tasks and 500 tasks with total response time are shown in the above figure 5.1 and 5.2.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

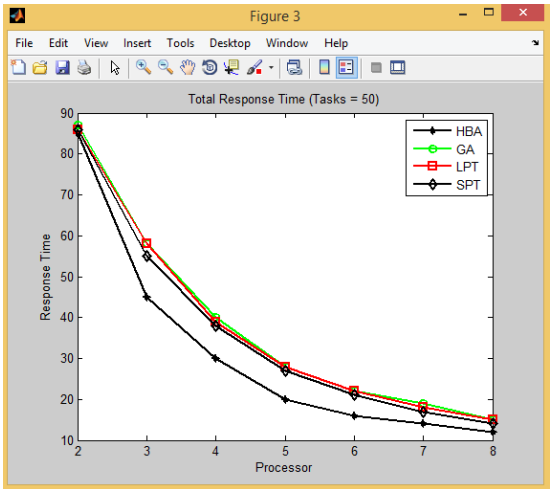

Fig 5.1: Total response time for 50 tasks

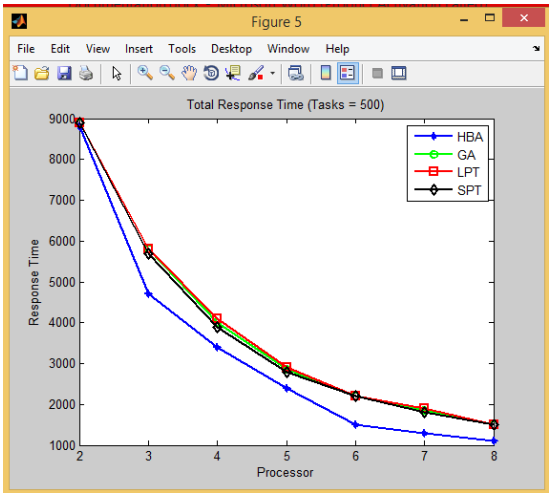

Fig 5.2: Total response time for 500 tasks

The system utilization of GA and Honey Bee are greater than the CPU scheduling algorithms SPT, LPT. The system response is very quickly even the number of tasks are changed dynamically; it will results increasing the utilization of system resources efficiently. Therefore, the system utilization for GA is greater than CPU scheduling algorithms which are shown in the below fig 5.3 and fig 5.4:

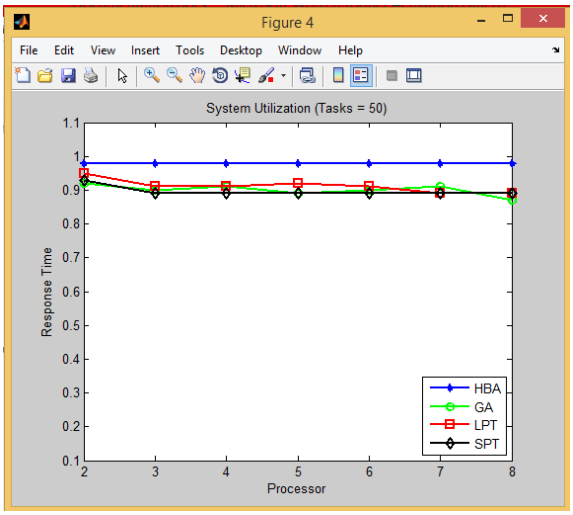

Fig 5.3: system utilization for 50 tasks

DOI10.17148/IJARCCE.2017.65143

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 5, May 2017

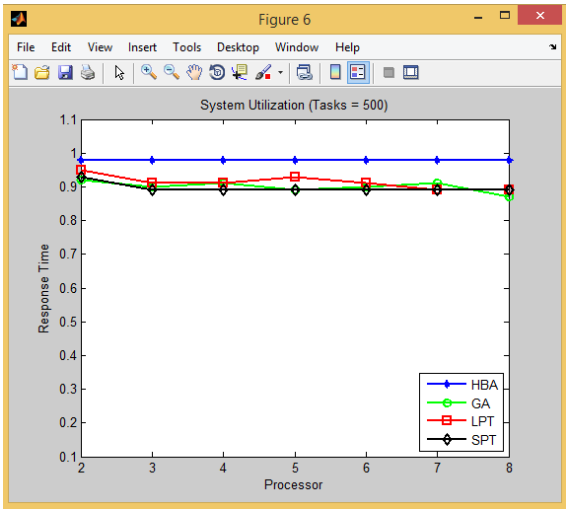

Fig 5.4: system utilization for 500 tasks

6. Conclusions and Future references:

In this paper, we proposed a best solution for load balancing in heterogeneous distributed systems with meta-heuristic and bio-inspired algorithms such as Genetic Algorithm (GA) and Honey bee algorithms. Genetic and Honey Bee algorithms performs well and provide global optimization solution for load balancing which increases the system utilization, decreases the total response time and increase throughput. The simulation results compare the performance of Honey Bee, LPT, SPT, GA algorithms. The future study for this project is different meta-heuristic and bio-inspired algorithms can be used to implement different online shopping websites in cloud to perfectly manage the load with better performance, capability, scalability in the heterogeneous environment.

REFERENCES

[1] Priyanka Gonnade, Sonali Bodkhe "An Efficient load balancing using Genetic algorithm in Hierarchical structured distributed system"
[2] Young, G.O., (1964), Synthetic structure of industrial plastics (Book style with paper title and editor, Plastics, 2nd ed., J. Peters, Ed. New York: McGraw-Hill, 3:15–64.
[3] Bibhudatta Sahoo, Sudipta Mohapatra and S. K. Jena, "A Genetic Algorithm Based Dynamic Load balancing Scheme for Hetrogenous Distributed Systems
[4] Yue-Jiao Gong, Wei-neng Chen, Zhi-Hui Zhan, Jun zhang, Yun Li, Quingfu 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, 2014.
[5] I. zelinka, "A survey on evolutionary algorithms dynamics and its complexity-Mutual relations, past, present, and future," elsevier, vol. 25, pp. 2-14, 2015.
[6] S. Jyothsna Asst.professor "Dynamic Load Balancing in Cloud Using Honey Bee Optimization"
[7] Chandrakanta korat, Piyush Gohel "A Novel Honey Bee Inspired Algorithm For Load Balancing In Cloud Environment"
[8] Monika Rathore, Sarvesh rai, Navadeep Saluj "Load Balancing Of Virtual Machine Using Honey Bee Galvanizing Algorithm in Cloud"
[9] Z. Bingual, A.S.Sekmen, S. Palaniappan, S. Sabatto "Genetic Algorithms Applied to Real time Multiobjective Optimization Problems "
[10] William A Greene "Dynamic Load Balancng via Genetic Algorithm"