Genetic Algorithm for Parallel Process Scheduling

Rachhpal Singh
Asstt. Prof., P.G. Department of Computer Science & Applications,
Khalsa College (KCA), City: Amritsar (State-Punjab-INDIA).

ABSTRACT
The hindrance of process scheduling in parallel and distributed system is challenging job in the field of computer science as research. Scheduling in parallel computing system plays an important role in overall for the enhancement of the performance of system in distributing the jobs. Scheduling in such type of system has a NP complete or NP hard problem. Process scheduling in parallel and distributed system allocate the processes to processor. By doing so make-span or execution time can be minimized. Further the processors utilization in the parallel computing system can be maximized. One more advantage of this one is to maximize the load balancing of jobs in the parallel processing. To tackle such type of problem a heuristic approach can be utilized that is known as Genetic algorithm. Genetic algorithm optimizes the problem in minimum time with maximum workload and is based on natural selection and natural genetics. Here with the help of GA, the problem can be made more simple and fast by considering load balancing efficiently and with the help of simulation results, the performance and efficiency of the proposed genetic algorithm can be evaluated.

Keywords
Parallel system, Directed Acyclic Graph, Genetic algorithm.

I. INTRODUCTION:
The concept of parallel processing is different from sequential processing and it refers to the concept of speeding-up the execution of a task by dividing the task into multiple fragments that can execute simultaneously, each on its own processor in order to obtain faster results. Parallel processing in the task scheduling can be effectively used for tasks that involve a large number of calculations, and have time constraints. Scheduling is the crucial step in the field of parallel programming and it deals with the optimal assignment of a set of tasks onto parallel system and orders their execution so that the total execution time is minimized. The efficient execution of the schedule on parallel system takes the structure of the application and the performance characteristics of the proposed algorithm. Many heuristics and approximation algorithms have been proposed to fulfill the scheduling task. It is well known NP-complete problem or NP hard problem. This study proposes a genetic based approach to schedule parallel tasks on a set of parallel system [1]. Here in this study, the scheduling problem has - next to search for an optimal mapping of the task and their sequence of execution and also search for an optimal configuration of the parallel system. The computational complicated process cannot be executed on the parallel computing machine in an accepted interval time. So to avid such type of hindrance, the big task must be divided into small sub-process and further the sub-process can be executed either in the expensive multiprocessor or in the parallel distributed system. A complete parallel system is best and can be utilized due to its cost and performance ratio. Scheduling in parallel environment is a critical factor in enhancing the overall system performance. Process scheduling in a parallel and distributed computing environment can be stated as allocating processes to processors so that total execution time will be minimized, utilization of processors will be maximized, and load balancing will be maximized [2]. Parallel and distributed process scheduling is further sub-divided into two parts:

1). In the first part all the processes in parallel and distributed environment are distributed on all the set computers.

2). In the second part the execution criteria and sequence on each processor must be determined according to the user requirement.

The techniques available to solve the task scheduling problem in parallel and distributed processing environment can be sub divided into three main classes as:

- A Graph theory based approach [3].
- A Mathematical models based method [4].
- Heuristic techniques [5].

An search technique or heuristic search algorithm can be categorized into three sub classes as:

- The iterative improvement algorithms [7].
- The probabilistic optimization algorithms.
- A constructive heuristics search technique.

Heuristic can obtain sub optimal solution in ordinary conditions and optimal solution in particular situations. In the first phase of process scheduling in a parallel distributed computing system, it is the process distribution on different parallel stream computer. Here the critical aspect is load balancing. Recently created processes may be overloaded heavily while the others are under loaded or idle. The major objectives in such type of environment is load balancing and are used to speared load on parallel processors equally, maximizing processors utilization and minimizing total execution time [6]. The second phase of parallel process scheduling in parallel distributed environment is process execution ordering on each processor. There is popular approach in the heuristics search is known as Genetic algorithm (GA). GA is guided random search technique that mimics the principles of evolution and natural genetics [8]. It searches optimal solution from entire solution space and from a pool of questionnaire data. Besides to obtain the reasonable solution in the pool of solutions and deals in all the difficult situations, the main disadvantage of this process is to spend much time for schedule. So to overcome such type of problem a genetic algorithm is proposed to overcome such type of drawback in realistic world according to the user's satisfactions and conditions. Here the problem can be solved easily by using the power of genetic algorithm in the field of parallel and distributed computing system and it is based on the load of the processor. Mapping was done on each schedule with a chromosome that shows the execution order of all existing process on processors. The best and fittest chromosomes are selected to reproduce offspring. Note that the parallel and the distributed system processes are in non uniform and non-pre-emptive form. This research paper is subdivided into five sections. Genetic algorithm approach and techniques with the problem description be described in the section II. The purposed genetic algorithm with the parallel and distributed system was explained in the section III. Experimental results with the performance analysis were described in the section IV. Last but not least the discussion and conclusion was done in the last section i.e. Section V. Rest are the references taken for this research paper.

II. GENETIC ALGORITHM WITH PROCESS MODEL:

Parallel system scheduling can be classified into many different classes based on the characteristics of the tasks to be scheduled, the parallel system and the availability of the information. The strategy behind the execution of the tasks on parallel system environment is to efficiently partitioning the huge task into set of tasks of appropriate gain size and an abstract model of the partitioned tasks that can be represented by Directed Acyclic Graph (DAG). The focus is on a deterministic scheduling problem in which there exist precedence relations among the tasks to be scheduled. A deterministic scheduling problem [6] is one in which all information about the tasks and the relation to each other such as execution time and precedence relation are known to the scheduling algorithm in advance and the processor environment. Heterogeneity of processors means that the processors have different speeds or processing capabilities. Here in this, a study has been done regarding the task scheduling problem as a deterministic on the parallel multiprocessor environment. The main objective is to minimize the total task finish time (execution time + waiting time or idle time). The system used for simulation is loosely coupled non-uniform system, all task are non-pre-emptive and no process migration are assumed. The process scheduling problem considered in this paper is based on the deterministic model. A distributed system with m processors, m>1 should be modelled as follows: P= {p1, p2, p3,..., pm} is the set of processors in the parallel and distributed system. Each processor can only execute one process at each moment, a processor completes current process before executing a new one, and a process cannot be moved to another processor during execution. T = {t1,t2,t3,...,tn} is the set of processes to execution. E is an n ×m matrix, where the element eij $1 \leq i \leq n$, $1 \leq j \leq m$ of E, is the execution time of process ti on processor pj. In the parallel and distributed environment the execution time of an individual process ti on all processors is equal. F is a linear matrix, where the element fi $1 \leq i \leq n$ of F, is the target processor that is selected for process ti to be executed on. C is a linear matrix, where the element ci $i \leq i \leq n$ of C, is the processor that the process ti is presented on just now. The problem of process scheduling is to assign for each process a processor so that total execution time will be minimized, utilization of processors will be maximized, and load balancing will be maximized. The processor load for each processor is the sum of process execution times allocated to that process [11]. The length of schedule T is the maximal finishing time of all processes or maximum load [11].

Length of schedule = maximum of load of each process

The processor utilization for each processor is obtained by dividing the sum of processing times by scheduling length. The Average of process utilization is obtained by dividing the sum of all utilizations by number of processors [11].

As to solve the NP complete problem or NP hard scheduling problem, the Genetic algorithm is used and it is guided random heuristic search algorithm based on the principles of evolution and natural genetics. GA deals with the past results by combining the exploitation of the past results with the exploration of new areas of the search space. Genetic algorithm is the mimic some of the innovative flair of human search and with the help of the fittest techniques and a structured yet randomized information exchange. Already there is a solution graph algorithm. As graph algorithm is randomized but not simple random walks and so cannot solve the problem in easy mode. This new heuristic technique maintains a population of candidate solutions that evolves over time and ultimately converges. As the concept of GA, Individuals in the population are represented with chromosomes and each individual is numeric fitness value that measures how well this solution solves the problem. To sole the GA approach three operators are used in a sequential manner. These three operators are selection operator, crossover operator and mutation operator. The selection operator selects the fittest individuals of the current population to serve as parents of the next generation. Next the second operator the crossover operation chooses randomly a pair of individuals and exchanges some part of the information. The third one and last one is the mutation operator and it takes an individual randomly and alters it. As natural genetics, the probability of crossover is usually high, the population evolves iteratively (in the genetic algorithm terminology, through generation) in order to improve the fitness of its individuals. The structure of genetic algorithm is a loop composed of a selection, followed by a sequence of crossovers and mutations. Probabilities of crossover and mutation are constants and fixed in the beginning [2]. Finally, genetic algorithm is executed until some termination condition achieved, such as the number of iterations, execution time, execution time, result stability, etc.

III. PROPOSED GENETIC ALGORITHM:

GAs operate through a simple cycle of stages: creation of a population strings, evaluation of each string, selection of the best strings and reproduction to create a new population. The individuals are encoded in the population string known as chromosomes. Once the chromosome has been coded, it is possible to evaluate the performance or fitness of individuals in a population. A good coding scheme [7][8] will benefit operators and make the object function easy to calculate. During selection, each individual is assigned a fitness value given by the objective function and choose the fittest individual of the current population to serve as parent of the next generation. Reproduction involves two types of operators namely crossover and mutation. The crossover operator chooses randomly a pair of individuals among those selected previously and exchange some part of the information. The mutation operator takes an individual randomly and works with node duplication heuristics, so that, the total execution time of the schedule should me minimum[9].

A. Creation of the population string:

The first step in the GAs algorithm is the creation of the initial population. Number of processors, number of tasks and population size are needed to generate initial population. The initial population is initialized with randomly generated individuals. The length of all individuals in an initial population is equal to the number of tasks in the DAG. Each task is randomly assigned to a processor[1].

B. Evaluation of the fitness function:

The fitness function used for proposed genetic algorithm is based on the total completion time for the schedule, which includes execution time and communication delay time. The fitness function separates the evaluation into two parts: Task fitness and processor fitness. The task fitness focuses on ensuring that all tasks are performed and scheduled in valid order. A valid order means that a precedence relations are

satisfied i.e. successor task cannot scheduled until predecessor has been completed.
The processor fitness component attempts to minimize the processing time. Therefore, the fitness values (task & processor) have been evaluated for all chromosomes and the probability of higher fitness is to be selected for reproduction from current generation to the next generation [1].

C. Selection operator:

The design of the fitness function is the basic of selection operation, so how to design the fitness function will directly affect the performance of genetic algorithm. GAs uses selection operator to select the superior and eliminate the inferior. The individual are selected according to their fitness value. Once fitness values have been evaluated for all chromosomes, we can select good chromosomes through rotating roulette wheel strategy. This operator generate next generationby selecting best chromosomes from parents and offspring [1].

D. Crossover operator:

Crossover operator randomly selects two parent chromosomes (chromosomes with higher values have more chance to be selected) and randomly chooses their crossover points, and mates them to produce two child (offspring) chromosomes. We examine one and two point crossover operators. In one point crossover, the segments to the right of the crossover points are exchanged to form two offspring as shown in Figure. 1(a) and in two point crossover [9][10], the middle portions of the crossover points are exchanged to form two offspring as shown in Figure. 1(b).

Randomly selects parent 1 and 22:

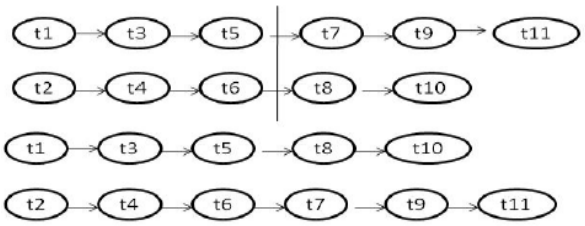

Figure. 1(a) One point crossover

Randomly selects parent 1 and 2:

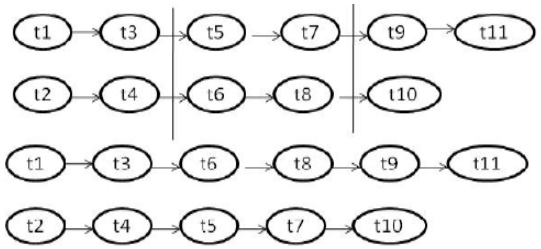

Figure. 1(b) Two point crossover

E. Mutation operator:

A mutation operation is designed to reduce the idle time of a processor waiting for the data from other processors. It works by randomly selecting two tasks and swapping them. Firstly, it randomly selects a processor, and then randomly selects a task [2] on that processor. This task is the first task of the pair to be swapped. Secondly, it randomly selects a second processor (it may be the same as the first), and randomly selects a task. If the two selected tasks are the same task the search continues on. If the two tasks are different then they are swapped over (provided that the precedence relations must satisfy).

IV. PERORMANCE ANALYSIS WITH EXPERIMENTAL RESULTS:

The proposed algorithm was simulated on Core i3 processor having 2GB RAM as memory and 500 GB as hard disk as permanent storage. After applying the different values of population size as chromosome strings and number of generations as iterations, number of options of different results occurs and can get the best solution from the set of different options of solutions by using the GA search technique [11]. After analysis, a performance evaluation was drawn and the performance of such type of GA was based on total make-span i.e. total completion time which further means scheduling length and average processor utilization. Here by setting the experiment, population size was assumed. Let it be 500 having a group size 10 and so fifty groups were maintained. Experiment was analysed on the basis of different argument values and from these values, results can be concluded [11]. So performance was measured by applying the effect of increasing number of processes on scheduling length and average processor utilization and final results were shown in the table 1 and table 2.

Table 1

Number of Processor	*Length of schedule*
10	50
20	55
30	75
40	70
50	90

Table 2

Number of Processor	*CPU utilization*
10	110
20	95
30	70
40	80
50	85

As from the table 2, it was noticed that when number of process were increased then utilization of CPU was obtained higher. Similarly by using the purposed GA, when the number of generations was increased, then it gives a better performance as compared to table 1 and table 2. The new computed results by using the proposed GA algorithm was shown in the table 3 and table 4 as shown below.

Table 3

Number of Generations	*Length of schedule*
10	95
20	90
30	98
40	110
50	120

Table 4

Number of Generations	*CPU utilization*
10	95
20	85
30	90

40	110
50	120

It was noticed that as the number of generation was increased the scheduling length was reduced and also note that the performance the generated process assignment increases after each iteration or generation. Also note that as the number of generation was increased, the CPU utilization was obtained higher. It shows the impact of increasing number of processes on scheduling length and average processor utilization and when number of process is increased, higher utilization is obtained [11].

V. CONCLUSIONS:

In parallel and distributed computing environment, scheduling and CPU utilization has a vital role in measuring the performance and throughput of any system. So by applying a new heuristic technique genetic algorithm, problem was solved more efficiently in such a fine way that it simultaneously minimizes the scheduling length as well as maximizes the average processor utilization. At end it minimizes the computation time with the help of genetic algorithm approach and so optimizes the results.

VI. REFERENCES:

[1] M. Nikravan and M.H.Kashani " A Genetic Algorithm For Process Scheduling in Distributed Operating systems considering Load Balancing", *European Conference on Modelling and simulation.*

[2] P.Y.R.Ma, E.Y.S.Lee, & J.Tsuchiya, "A Task Allocation Model for Distributed Computing Systems", *IEEE Trans.On Computers, 31*(1), 1982, 41-47.

[3] G.L.Park, "Performance Evaluation of a List Scheduling Algorithm In Distributed Memory Multiprocessor Systems", *International Journal of Future Generation Computer Systems 20*, 2004, 249-256.

[4] C.I.Park, & T.Y.Choe, "An optimal scheduling algorithm based on task duplication" , *IEEE Trans. on Computers, 51*(4), 2002, 444–448.

[5] A.Y.Zomaya, & Y.Teh, "Observations on Using Genetic Algorithms for Dynamic Load-Balancing", *IEEE Trans .On Parallel and Distributed Systems, 12*(9), 2001, 899-911.

[6] L.M.Schmitt, "Fundamental Study Theory of Genetic Algorithms" , *International Journal of Modelling and Simulation Theoretical Computer Science 259*, 2001, 1 –61.

[7] A.Y.Zomaya, C.Ward, & B.Macey, "Genetic Scheduling for Parallel Processor Systems: Comparative Studies and Performance Issues", *IEEE Trans. On Parallel and Distributed Systems*, 10(8), 1999, 795-812.

[8] D.Goldberg genetic algorithm in search optimization and machine learing reading Mass addssion wesly 1989.

[9] Sung-Ho Woo, Sung-Bong Yang, Shin-Dug Kim, Tack-Don Han, "Task scheduling in distributed computing systems with a genetic algorithm", *High-Performance Computing on the Information Superhighway*, HPC-Asia '97, 1997, p. 301.

[10] A.Y. Zomaya, C. Ward, and B.Macey "Genetic Scheduling for Parallel Process System: Comparative studies and Performance Issues", *IEEE Tansaction on Parallel and Distributed Systems*, Vol.10, No.8 pp795-812 Aug. 1999.

[11] Vinay Harsora, Dr. Apurva Shah, "A Modified Genetic Algorithm for Process Scheduling in Distributed System", IJCA Special Issue on "Artificial Intelligence Techniques - Novel Approaches & Practical Applications", AIT, 2011, pp. *36-40*.

[12] Sharma, Manik, Gurdev Singh, and Harsimran Kaur. "A Study Of BNP Parallel Task Scheduling Algorithms Metric's For Distributed Database System." *International Journal of Distributed and Parallel Systems* 3.1 (2012): 157.

[13] Sharma, Manik, Gurdev Singh, and Rajinder Virk. "Analysis of Joins and Semi Joins in a Distributed Database Queries."International Journal of Computer Applications 49.16 (2012).

[14] Sharma, Manik, et al. "A comparative study of static object oriented metrics." *International Journal of Advancements in Technology* 3.1 (2012): 25-34.