IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. II (May – Jun. 2015), PP 73-78
www.iosrjournals.org

Scheduling Using Multi Objective Genetic Algorithm

Anu Dogra[1], Kritika Dhiman[2]
(Computer Science, Guru Nanak Dev University , Amritsar , India)

Abstract : *Multiprocessor task scheduling is considered to be the most important and very difficult issue. Task scheduling is performed to match the resource requirement of the job with the available resources resulting in effective utilization of multiprocessor systems. In this paper, a Multi Objective Genetic algorithm (MOGA) is proposed for static, non- pre-emptive scheduling problem in homogeneous fully connected multiprocessor systems with the objective of minimizing the job completion time. The proposed GA is used to determine suitable priorities that lead to a sub-optimal solution. Our proposed GA for a given job scheduling problem proves that GA results in better sub-optimal solutions.*

Keywords: *Directed Acyclic Graph, Genetic algorithm, multiprocessing environment, parallel computing, Task scheduling.*

I. Introduction

Today's applications are made up of several of tasks that is to processed by multiple processors to achieve a specific goal. So they deal with the allocation of individual tasks to suitable processors and assignment in proper order of execution of tasks. Parallel scheduling increase the performance by properly scheduling tasks to processors and reduce the completion time. Well-organized Task scheduling is a necessary part for appropriate functioning of parallel processing [7]. Efficient task scheduling is one of key factors for providing high performance on heterogeneous computing. The field of parallel task scheduling is one of the most advanced and rapidly evolving fields in computer science. High performance computing is expected to bring a breakthrough in the increase of computing speed [1]. Optimal solution for scheduling is not feasible for Such Problems which are NP -complete. A large number of algorithms has been proposed which attempt to bring a sub optimal solution. Mainly four algorithms are proposed in parallel computing for task scheduling
BNP (Bounded Number of Processors) scheduling
UNC (Unbounded Number of Clusters) scheduling
TDB (Task Duplication Based) scheduling
APN (Arbitrary Processors Network) scheduling
Multiprocessor scheduling processor involves mapping a Directed Acyclic Graph (DAG) for a collection of computational tasks and data precedence onto parallel processing system. The objective of scheduling is to minimize the overall program finish time by proper allocation of jobs to processors and arrangement of execution sequencing of tasks [2].

1.1 APN Scheduling

It is one of the scheduling algorithm from the four above parallel scheduling algorithms. Our main focus is on this algorithm and we want to make such an algorithm which can find better results than one of the further algorithms of APN that is MH (mapping heuristics). Algorithms in this class take into account specific features such as the number of processors and their interconnection topology. These algorithms can schedule tasks on processors and messages over a network communication links. Scheduling of tasks is dependent on the routing strategy used by the original network [9]. The mapping including the temporal dependencies, is therefore implicit-without going through a separate clustering phase. In the following, we discuss four such algorithms.
MH (Mapping Heuristic) algorithm
DLS (Dynamic Level Scheduling) algorithm
BU (Bottom Up) algorithm
BSA (Bubble Scheduling and Allocation) algorithm [5]
To estimate performance for these algorithms we have to analyse the various parameters are

1.1.1 Make span: It is defined as the completion time of the algorithm and it is calculated by the extent of the finishing time of the algorithm.

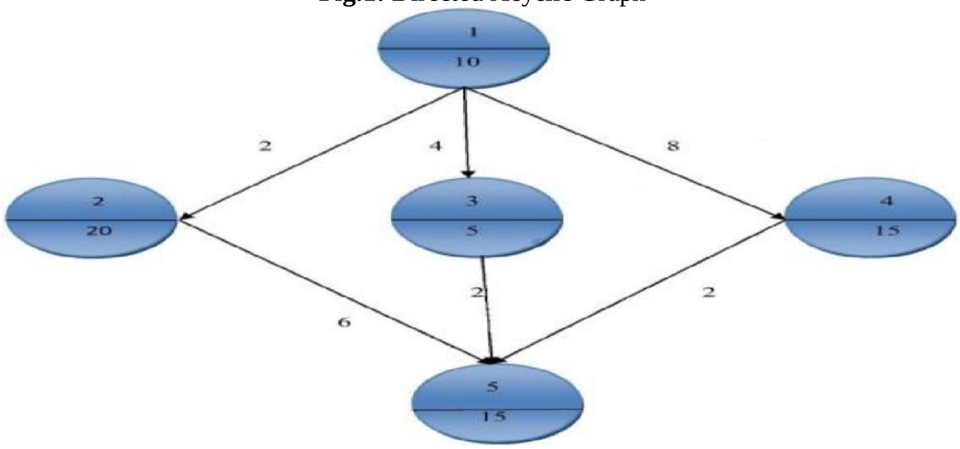
1.1.2 Speed up: Speed up is calculated by computing the execution time for the processors running sequential and parallel. Dividing the finishing time of processors running sequentially by the execution time of processors running parallel is termed as speed up.

1.1.3 Scheduled Length Ratio: It is the proportion of the make span of algorithm to the critical path values of DAG [10].

1.2 Dag Model

DAG (Directed acyclic Graphs) shows the tasks that are allocated to the uniform processors .It is a task graph in which various tasks assigned to the processors are represented and set with the purpose of minimizing the all-purpose finish time by proper share of the tasks such as the maximum throughput can be achieved. DAG is regular model of parallel program consisting of set of processes amongst which there are dependencies. Each process is an inseparable unit of execution, articulated by a node. A node has one or more inputs and can have one or more output to different nodes. When all inputs are available the node is triggered to perform. After its implementation it generates its output. In this model a set of nodes $(n_{1,}n_{2,}\ldots\ldots n_n)$ are related by a set of directed edges, which are represented by $(n_{i\ ,}n_j)$ wheren_i is called the parent node and n_j is called the child node. A node with no parent is called an Entry node and node without child node called an Exit node. The weight of a node, denoted by w (n_j) represents the process finishing time of a process. Since each edge correspond to a message transfer from one process to another, the weight of an edge denoted by c $(n_i\ ,n_j)$ is equal to the message transmission time from node $n_{i\ ,}$to n_j. Thus c $(n_{i\ ,}n_j)$ becomes zero when n_i and n_j are scheduled to same processor because intra processor communication time is negligible compared with the inter processor communication time. The node and edge weights are usually obtained by estimations. Some variations in the generic DAG model are described below in fig.1 [3]

Fig.1: Directed Acyclic Graph



1.3 Scheduling Attributes

The main scheduling attributes used in DAG for transfer priority while evaluating the algorithms are as follows:

(1) T- Level: T-level of a node n_i in DAG is the length of the greatest path from entry node to n_i not including n_i. It is addition of all the nodes computational costs and edges weights all along the path.

$$T-level\ (n_i) = max\ (T-level\ (n_m) + w_m + c_m)$$

where n_m € predecessors of n_i , w_m stands for computational charge, c_{mi} stands for the message cost and t-level $(n_{entry}) = 0$

(2) B- Level: B-level of node n_i in DAG is the length of the greatest path from n_i to the exit node. It is the addition of all the nodes computational costs and edges weights all along the path.

$$b-level\ (n_i) = w_i + max\ (b-level\ (n_m) + c_{mi})$$

Where n_m € successors of n_i , w_m stand for computational cost, c_{mi} stands for the message cost and b- level $(n_{exit}) = $ w (v_{exit}).

(3) SL (Static Level): If the edges weights are not taken while considering the b-level , it is called static level.

$$SL\ (n_i) = w_i + max\ (SL\ (n_m))$$

Where n_m € successors of n_i and SL $(n_{exit}) = $ w (v_{exit})

(4) CP (Critical Path): It is the length of the longest path from station node to the exit node in DAG.

(5) EST (Earliest Start Time): It is same as the t-level.

$$EST\ (n_i) = max\ (EST\ (n_m) + w_m + c_{mi})$$

Where n_m € predecessors of n_i , w_m stands for computational cost, c_{mi} stands for the computational cost and EST (n_{entry}) = 0
(6) LST (Latest Starting Time): Latest Starting Time of node is computed by follow the path starting from exit node upwards till the preferred node is reached.

$$LST\ (n_i) = min\ (LST\ (n_m) - c_{m,i}) - w_m$$

Where n_m € predecessors of n_i , w_m stands for computational cost, $c_{m,i}$ stands for the computational cost and LST (n_{exit}) = EST (n_{exit}).
(7) DL (Dynamic level): Dynamic Level of the node is considered by subtracting the earliest start time from the static level.

$$DL = SL - EST$$

Where SL stands for Static Level and EST stands for Early Start Time [4].

1.4 Genetic Algorithms

GAs motivated by DARWINs theory about continued existence of the fittest. Since in nature , competition among individuals for resources consequences in fittest individuals dominating over the weaker ones. Planned and developed in 1960s by John Holland. These algorithms are search algorithms based on natural selection and natural genetics [6]. It works under population of solution rather than a particular solution. Here investigate begins by initializing a population of individuals. Individual solutions are chosen from the population and after mating them a new children is generated. The mating is done by crossing over genetic material from two parents and as a result transfers the data from one production to next. This sometimes promotes the diversity. Fitness function is used to measures the quality of each candidate.
The three main steps applied in genetic algorithm are [7]

1.4.1 Selection: Selection improves the superiority of population by generous chromosomes a better chance to get derivative in the next generation. It gives first choice to better individuals, allow them to pass on their genes to the next generation. The integrity of each individual depends on its fitness. Fitness is resolute by a fitness function which may be one-sided, objective or judgement.

1.4.2 Crossover: The two new offspring is generated from mating in excess of the two individuals that are selected from population by means of selection method and set them into the next generation. By recombining portions of good quality individuals, this process is expected to create even better individuals.

1.4.3 Mutation: Mutate a chromosome to form new chromosomes. Figure a new chromosome by randomly pick chromosome. This operator modify one or more genetic material values in an individual. The result of this modification can be completely new gene values being added to the same group. The genetic algorithm may be accomplished to reach at better solution with these new gene values [8].

II. Literature Survey

S.Sharma et al. (2014) Author described various parallel scheduling algorithms and their drawbacks. Out of static algorithms the DCP (Dynamic Critical Path) is the best algorithm having admissible time complexity and cheap in terms of number of processors used. But multiprocessors problem is NP-complete in nature and becomes more complex under calm assumptions. Therefore a genetic approach based was proposed to get together the goals of high performance, scalability and fast running time called parallel genetic scheduling algorithm [8].

A.sharma et al. (2013) developed Parallel processing is a field in which diverse systems run together to save the time of processing and to increase the presentation of the system and to balance load in this paper we have combined HLFET, MCP, DLS , ETF with fuzzy logics to check out effects on parameters like speedup, process utilization , make span. So it has established that fuzzy logic execute better than single algorithms [3].

A.Kaur et al. (2013) Author designed Mapping Heuristic method where list is prepared according to maximum priority node. Here routing table is maintained with each processor straight path between processing element is also maintained in this table. Route from source to destination gets busy when message is sent and become free when received by the receiver and accordingly updation in routing table is complete. Also applying genetic in this approach may find an optimal result and can perform better than simple mapping heuristic [6].

P.Kaur (2013) Author developed Scheduling and mapping of task graphs to the processors which is one of the most critical problems in parallel computing. Due to NP- completeness of any problem, the optimal solution can not be find in a sensible time. The existing heuristics is that they can assess the problem size which are very small. Author implements APN Dynamic Level Scheduling algorithm by using genetic operators for task scheduling in parallel multiprocessor system with communication delays to reduce the completion time and to boost the throughput of the system. The parameters used are make span time, processor operation and scheduled length ratio. The graphs show better results of dynamic level scheduling with genetic operators as compared to simple dynamic level scheduling algorithm [9].

P.Sharma et al.(2013) Author proposed genetic algorithms for a superior solution in parallel processing. Adaptive parameter approach is applied to enlarge the performance of genetic algorithm. Paper presents GA implementation. Results shows with amplify in number of nodes, speedup increases but communication below also increases [7].

P.Gupta et al. (2012) Discussed one of the algorithm of APN scheduling which is to be explained in detail called as mapping heuristic (MH) APN scheduling algorithm. It is implemented for task processing in parallel multiprocessor system including the communication delays to reduce the completion time and throughput of the system. [4]

N.Arora (2012) Author proposed Directed acyclic Graphs which schedule the tasks to reduce the completion time. Various algorithms of scheduling tasks are analyzed which are categorized in four groups. The performance is the important factor in every algorithm. Also the performance is measured using these algorithms by manipulative some parameters. [10].

W.Nasri (2012) Author explained efficient task scheduling which is one of the key factors to provide high presentation in heterogeneous systems. Directed Acyclic Graphs are explained for the scheduling in heterogeneous systems. In this work, author has addressed the problem of DAG scheduling on heterogeneous platforms made of clusters of clusters. To be sure, author has developed a new algorithm called SMC (Scheduling on Multi Clusters) based on two principal phases: the organization of tasks and the project of tasks to the most appropriate processors. SMC algorithm gives an improvement in the first phase which is a very important leading to an perfection of performance [1].

R.Kaur (2012) Author claims the designed Genetic Algorithm as the more efficient and finds more optimal solution than the parallel computing algorithms. To evaluate the performance the future GA is experienced by mapping the tasks into the Directed Acyclic Graph. Performance study of static algorithms of BNP and future GAs for a given difficulty shows that GAs are better than BNP scheduling [2].

P.Kaur (2011) Author surveys algorithms that assign parallel tasks represented by an edge-directed acyclic graph (DAG). Taking purpose to minimize the execution time, assess and evaluate the performance of the individual algorithms to find the best algorithm. Dissimilar algorithms are analyzed and classified into four groups. BNP algorithm is study and discussed, and to measure the performance of BNP algorithm and evaluate the best algorithm. So it is concluded from above results that DLS is one of the efficient algorithms [12].

E.S.C et al. (2003) Author shows problem in allocating non identical tasks in multiprocessor model assumes identical processors and at a time only one processor may execute one task. Here GA approach is proposed to finding optimal solution for arbitrary task graph.GA provides set of optimal solutions. GA is free of List Scheduling Algorithms anomalies [11].

I.Ahmed et al. (1995) Author Compared various algorithms for scheduling and also compare one of the class of the algorithms called APN scheduling algorithms. He discussed the designs ,philosophies and principles behind these algorithms and access their merits and deficiencies. Also APN algorithms can be complicated so further research is required in this area [5].

III. Nature of the problem

- The Mapping Heuristic problem of parallel Job Scheduling is NP-complete.
- The MH-problem may take super polynomial time to solve task scheduling. This is because they are not scalable in nature.
- Accurate methods such as Branch and Bound method and dynamic programming take considerable computing time if an optimal solution exists.

IV. Purpose

APN scheduling: These algorithms execute scheduling of tasks amongst various processors in which the processors are connected via a network of arbitrary topology. The algorithm in this class take into account some specific features such as number of processors and interconnection topology. They also list messages on the network message links.

MH scheduling: It is one of the class of APN scheduling. It performs scheduling on the basis of priorities assigned to the nodes. It assigns priorities by computing static B-level of the nodes. Nodes with highest priority will be assigned processor first and accordingly scheduling is done.

The mapping Heuristic problem of parallel Job Scheduling is NP-complete. The MH-problem may take super polynomial time to solve task scheduling. This is because they are not scalable in nature. Exact methods such as Branch and Bound method and dynamic programming take considerable computing time if an optimal solution exists

This paper deals with the use of Genetic Algorithm in field of Arbitrary Processors Network scheduling algorithm. Our purpose is to overcome the limitations of earlier techniques, it is more sensible to obtain a good solution near the optimal one. So this work will use genetic algorithm to obtain good solution near the optimal one. The overall objective is to reduce the make span time along with the reduction of execution time of genetic algorithm.

V. Objectives

1. To reduce the computational complexity of Mapping Heuristic in APN scheduling algorithm, the multi objective genetic programming will be used.
2. To establish a practical relationship between the development of Genetic algorithm and parallel scheduling in MH of APN scheduling .
3. To evaluate the effect of job size on the proposed algorithm.
4.To verify the proposed algorithm following parameters will be used

Speed up
SLR
Makespan

VI. Methodology

Proposed Mapping Heuristic scheduling using multiple objective Genetic algorithm . It searches optimal solution from entire solution space. We will overcome the drawbacks of previously explained algorithms of APN scheduling and try to attain our objectives using multiple objective Genetic Algorithm .

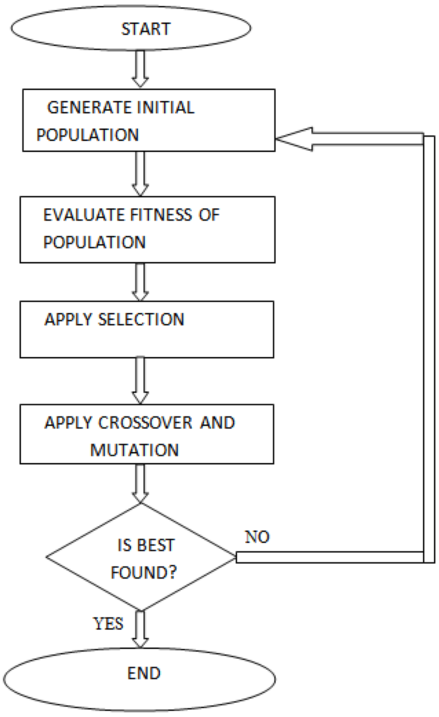
Genetic algorithm is a method that works on the chromosomes. Mainly three operators are used in it:

Selection operator
Crossover operator
Mutation operator

Genetic Algorithm works in the following steps:

STEP 1[Start]: Create random population of chromosomes, to be exact suitable solutions for the problem.
STEP 2[Fitness]: Calculate the fitness of each chromosome by resources of fitness function.
STEP 3[New population]: Create a new population by repeating following steps until new population is complete.
[SELECTION] Select two parent chromosomes from a population according to their fitness. Better the fitness, the bigger chance to be selected to be the parent.
[CROSSOVER] Crossover the parents to form the new offspring, that is children. If no crossover was performed then off spring will be the exact copy of the parents.
[MUTATION] Using mutation mutate new offspring at each locus.
[ACCEPTING] Then new offspring is placed in the new population.
STEP 4[Replace]: Replace with new generated population for further execution of the process.
STEP 5[Test]: Finally if the end condition is satisfied then stop and return the final result to the in the current population.
STEP 6[Loop]: Go to step 2.

Below the flow chart will explain the working of the genetic algorithm in fig. 2 (Directed Acyclic Graph)



VII. Conclusion

Scheduling system makes everyone's job easier. It provides a means of holding down costs through better use of personnel and equipment. it makes better use of multiprogramming capabilities. It may predict the effects of an increased workload, future equipment and personnel needs. It gives the full advantage of computational power provided by multiprocessors. It also increases throughput and reduces complexity. As planning out tasks is an important part of good business. So to achieve high performance, an efficient scheduling with an optimal solution is to be developed.

References

[1]. Nasri, Wahid, and Wafa Nafti. "Dept. of Comput. Sci., Higher Sch. of Sci. & Tech. of Tunis, Tunis, Tunisia." In Parallel Distributed and Grid Computing (PDGC), 2012 2nd IEEE International Conference on, pp. 114-119. IEEE, 2012.

[2]. Ravreet Kaur, Dr.Gurvinder Singh "Deptt of Computer Science & Engineering, Guru Nanak Dev University , Amritsar (Punjab), India" .Genetic Algorithm Solution for Scheduling Jobs in Multiprocessor Environment, 978-1-4673-2272 .IEEE,2012.

[3]. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No.5, 2013 QOS,Comparison of BNP Scheduling Algorithms with Expanded Fuzzy System.A.sharma

[4]. Gupta, Preeti. "A Mapping Heuristic Approach for Scheduling the Tasks on Parallel Multiprocessors." International Journal of Innovative Research and Development 1, no. 8 (2012): 198-206.

[5]. Ahmad, Ishfaq, Yu-Kwong Kwok, and Min-You Wu. "Performance comparison of algorithms for static scheduling of DAGs to multiprocessors." In Second Australasian Conference on Parallel and Real-time Systems, pp. 185-192. 1995.

[6]. Kaur, Amanpreet, and Prabhjot Kaur. "Implementation of Mapping Heuristic Genetic Algorithm." International Journal of Advanced Research in Computer and Communication Engineering (2013): 3224-3229.

[7]. Sharma, Prashant, and Gurvinder Singh. "Genetic Approach to Parallel Scheduling." International Journal of Engineering Mathematics and Computer Sciences 1, no. 2 (2013).

[8]. Sharma, Shreya. "Critical Analysis of Various Parallel Scheduling Algorithms." International Journal of Engineering and Computer Science.Volume 3 issue ,6 june,(2014)

[9]. Kaur, Prabhjot, and Amanpreet Kaur. "Implementation of Dynamic Level Scheduling Algorithm using Genetic Operators." International Journal of Application or Innovation in Engineering & Management (IJAIEM) Volume 2, Issue 7, July 2013

[10]. Arora, Nidhi. "Analysis and performance comparison of algorithms for scheduling directed task graphs to parallel processors." ANALYSIS 4, no. 2 (2012).

[11]. ESQUIVEL S.C., GATICA C. R., GALLARD R.H. Departamento de Informática Universidad Nacional de San Luis (UNSL) Ejército de los Andes 950 - Local 106 5700 - San Luis, Argentina. Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology), 2003

[12]. Kaur, Parneet, Dheerendra Singh, Gurvinder Singh, and Navneet Singh. "Analysis, Comparison and Performance Evaluation of BNP Scheduling Algorithms in Parallel Processing." International Journal of Information Technology and Knowledge Management 4, no. 1 (2011): 279-284.