Volume 2, Issue 5, May 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Multiprocessor Environment Using Genetic Algorithm

Sandeep Jain[1]
Assistant Professor
Department of Computer Science and Engineering , Doon Vally Engineering College Karnal

Shweta Makkar
Research Scholar
Department of Computer Science and Engineering, Doon Vally Engineering College Karnal sandeepjain4891@gmail.com
veenu.makker63@gmail.com

ABSTRACT*- Task Scheduling [1] is the allocation of resources over time to perform a collection of tasks. Real-time systems make use of scheduling algorithms to maximize the number of real-time tasks that can be processed without violating timing constraints. A scheduling algorithm provides a schedule for a task set that assigns tasks to processors and provides an ordered list of tasks. The schedule is said to be feasible if the timing constraints of all the tasks are met. Scheduling approaches can be classified according to the arrival time of tasks into static and dynamic and deterministic or stochastic scheduling. Task Scheduling in Multiprocessor is a term that can be stated as finding a schedule for a general task graph to be executed on a multiprocessor system so that the schedule length can be minimized. Task scheduling in multiprocessor systems also known as multiprocessor scheduling. Multiprocessor scheduling problems can be classified into many different categories based on characteristics of the program and tasks to be scheduled, the multiprocessor system, and the availability of information. Multiprocessor scheduling [2] problems may be divided in two categories: Static and dynamic task scheduling. In this paper we discuss the Multiprocessor Environment using genetic Algorithm.*

Keywords*: Scheduling, Multiprocessor, Task Secluding, Genetic Algorithm.*

I. INTRODUCTION

Task Scheduling [1] is the allocation of resources over time to perform a collection of tasks. Real-time systems make use of scheduling algorithms to maximize the number of real-time tasks that can be processed without violating timing constraints. A static or deterministic task scheduling is one in which precedence constraints and the relationships among the task are known well in advance While non-deterministic or dynamic scheduling [3] is one in which these information is not known in advance or not known till run time. A major factor in the efficient utilization of multiprocessor system is the proper assignment and scheduling of computational tasks among processors.

The problem can have many variations:

(i) The scheduling algorithm can be deterministic – also known as static – or nondeterministic: A deterministic task scheduling problem is defined as one in which the knowledge related to tasks, their relations towards each other, timing and the number of processors used are all a-priori knowledge. In a nondeterministic problem on the other hand, all or some of these factors can be input-dependent and vary according to run time conditions.

(ii) The tasks can be preemptive or non-preemptive: A preemptive task scheduling problem allows the tasks to be cut off from execution and another task to begin or continue its execution cycle [operating system example]. A non-preemptive problem in which task execution must be completely done before another task takes control of the processor.

(iii) The processors can be either homogenous or heterogeneous: Heterogeneity of processors means that the processors have different speeds or processing capabilities. In a homogenous environment on the other hand, all processors are assumed to have equal capabilities.

Efficient scheduling [8] of application tasks is critical to achieving high performance in parallel multiprocessor [9] systems. The objective of scheduling is to map the tasks onto the processors and order their execution. So that task precedence requirements are satisfied and minimum schedule length (or Makespan). The most common heuristic methods are List Heuristics, such as Earliest Task First (ETF) algorithm, Critical Path/Most Immediate Successor First (CPMISF) algorithm, and Dynamic Critical Path (DCP) algorithm etc. Another heuristic method is genetic algorithm. A genetic algorithm [7] is a domain-independent global search technique where elements (called individuals) in a given set of solutions (called population) are randomly combined until some termination condition is achieved.

II. GENETIC ALGORITHMS

Genetic algorithms operate on finite-sized populations of candidate schedules. At each iteration of the algorithm, relatively poor schedules are removed from the population and are replaced with new candidate schedules generated by: (1) applying mutations to individual schedules in the population; (2) applying cross-over operations to pairs of schedules in the population.

Genetic algorithms [10] [11] as powerful and broadly applicable stochastic search and optimization techniques, are the most widely known types of evolutionary computation [16] [11] methods today. The father of the original Genetic Algorithm was John Holland [13] who invented it in the early 1970's. In general, a genetic algorithm has five basic components as follows:

(i) An encoding method that is a genetic representation [11] (genotype) of solutions to the program.
(ii) A way to create an initial population of individuals [Davis, 1991].
(iii) An evaluation function, rating solutions in terms of their fitness, and a selection mechanism.
(iv) The genetic operators (crossover and mutation) that alter the genetic composition of offspring during reproduction.
(v) Values for the parameters of genetic algorithm

III. PROBLEM STATEMENT AND OBJECTIVE

A static scheduling problem consists of three main components: A multiprocessor system, an application and an objective for scheduling. The multiprocessor system consists of a limited number of fully connected processors (P1, P2... Pm). All the processors are heterogeneous meaning thereby a task may take different execution time on each processor. An application comprises tasks and their dependencies on each other.

dependencies **on each other**

It can be represented as a directed acyclic graph (DAG) [10, 11], G= (V, E, W), where the vertices set V consists of v non preemptive tasks, and v_i denotes the ith task. The edge set E represents the precedence relationships among tasks. A directed edge eij in E indicated that vj can not begin its execution before receiving data from vi. W is a matrix of vxm, and wij in W represents the estimated execution time of vi on jth processor.

OBJECTIVE

The main objective of the task scheduling is to determine the assignment of tasks of a given application to a given parallel system such that the execution time (or schedule length) of this application is minimized satisfying all precedence constraints.

ROLE OF GENETIC ALGORITHMS IN TASK SCHEDULING OPTIMIZATION

A multiprocessor scheduling [26] problem is defined as the assignment of a given set of tasks to a set of processors. These tasks should be assigned in a way such that the total execution time is minimized and certain criteria are met. A wide range of solutions and heuristics have been proposed to solve this important system optimization problem.

Task scheduling [26] is essential for the suitable operation of multiprocessor systems. The aim of task scheduling is to determine an assignment of tasks to processors for shortening the length of schedules. The problem of task scheduling [15] on multiprocessor systems is known to be NP-complete (Garey and Johnson, 1979) in general. Solving this problem using by conventional techniques needs reasonable amounts of time. Therefore, many heuristic techniques were introduced for solving it. A new heuristic algorithm for task scheduling, based on evolutionary method [11] is Genetic Algorithm [27]. Genetic algorithms have received much awareness as they are robust and guarantee for a good solution. Genetic algorithms (GA) are stochastic search techniques that can perform optimization based on natural selection [27] and evolution theories without relying on gradient information or becoming trapped in local minima [25].

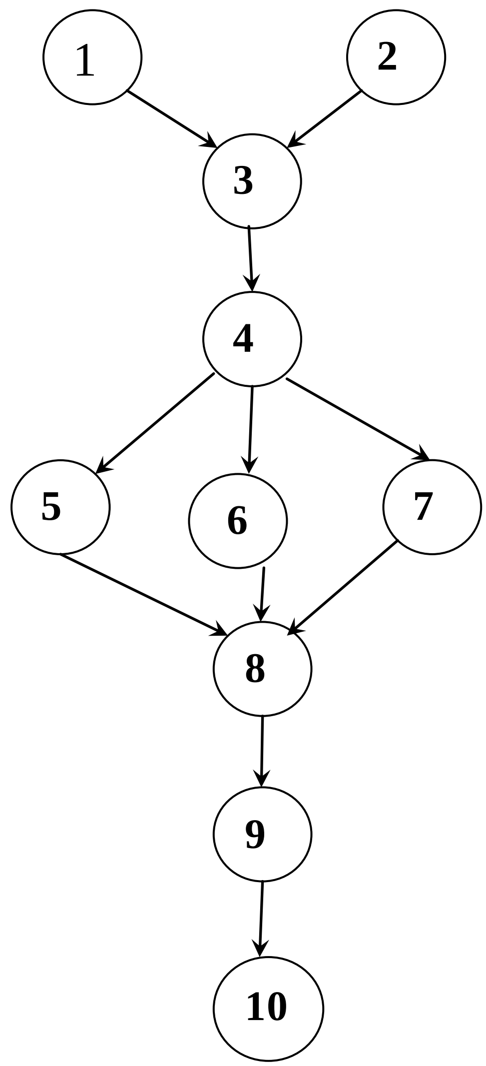

Fig 1: Directed Acyclic Graph of 10 nodes and their METHODOLOGY

IV. CONCLUSION

The problem of scheduling of tasks to be executed on a multiprocessor system is one of the most challenging problems computing. Genetic algorithms are well adapted to multiprocessor scheduling problems. As the resources are increased available to the GA, it is able to find better solutions. GA performs better as compared to other traditional methods. Overall, the GA appears to be the most flexible algorithm for problems using multiple processors. It also indicates that the GA is able to adapt automatically to changes in the problem to be solved. The advantages of the GA approach are that it is simple to use, requires minimal problem specific information, and is able to effectively adapt in dynamically changing environments.

REFERENCES

[1] G. Syswerda and J. Palmucci, "The application of genetic algorithms to resource scheduling", Proceedings of the Fourth International Conference on Genetic Algorithms and Their Applications, pages 502-508, San Mateo, CA, July 1991.

[2] G. A. Cleveland and S. F. Smith, "Using genetic algorithms to schedule flow shop releases", Proceedings of the Third International Conference on Genetic Algorithms and Their Applications, pages 160-169, San Mateo, CA, June 1989.

[3] J. H. Holland, "Adaptation in Natural and Artificial Systems", The University of Michigan Press, Ann Arbor, MI, 1975.

[4] Cottet, F., Delacroix, J, Kaiser, C., Mammeri, Z., "Scheduling in Real-time Systems", John Wiley & Sons Ltd, England, 2002.

[5] Goldberg, David E, "Genetic Algorithms in Search, Optimization and Machine Learning", Kluwer Academic Publishers, Boston, 1989.

[6] Mitchell, Melanie, "An Introduction to Genetic Algorithms", MIT Press, Cambridge, MA. 1996.

[7] L.M.Schmitt, "Fundamental Study Theory of Genetic Algorithms", International Journal of Modelling and Simulation Theoretical Computer Science. 2001.

[8] C. V. Ramamoorthy, "Optimal scheduling strategies in a multiprocessor system", IEEE Trans. Computers, vol. C-2I.,Feb. 1972.

[9] I. H. Kasahara and S. Narita, "Practical multiprocessing scheduling algorithms for efficient parallel processing", IEEE Transactions on Computers, 1998.

[10] E. S. H. Hou, R. Hong, and N. Ansari, "Efficient multiprocessor scheduling based on genetic algorithms", IEEE 1990.

[11] Muhhamad K. Dhodhi, Imtiaz Ahmad, Ishfaq ahmad, "A multiprocessor scheduling scheme using Problem-space genetic algorithms", IEEE 1995.

[12] Michael Bohler, Frank Moore, Yi Pan, "Multiprocessor Scheduling Using Genetic Algorithms", Twelfth International FLAIRS Conference, 1999.

[13] Yi-Wen Zhongiz, Jian-Gang Yang, "A Genetic algorithm for tasks scheduling in parallel multiprocessor systems", Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003

[14] Michael Rinehart, Vida Kianzad, and Shuvra S. Bhattacharyya, "A Modular Genetic Algorithm for Scheduling Task Graphs", 2003.

[15] Andrew J. Page and Thomas J. Naughton, "Framework for task scheduling in heterogeneous distributed computing using genetic algorithms", pp. 1-14, 2005.

[16] Andrew J. Page and Thomas J. Naughton, "Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing", 2005.

[17] Faezeh Montazeri, Mehdi Salmani-Jelodar, S. Najmeh Fakhraie and S. Mehdi Fakhraie, "Evolutionary Multiprocessor Task Scheduling", Proceedings of the International Symposium on Parallel Computing in Electrical Engineering (PARELEC'06), IEEE 2006.

[18] M. Salmani Jelodar, S. N. Fakhraie, F. Montazeri, S. M. Fakhraie, M. Nili Ahmadabadi, "A Representation for Genetic-Algorithm-Based Multiprocessor Task Scheduling", Congress on Evolutionary Computation, Vancouver, BC, Canada, IEEE July 16-21, 2006.

[19] M. Nikravan and M. H. Kashani, "A Genetic algorithm for process scheduling in distributed operating system considering load balancing ", Proceedings 21st European Conference on Modelling and Simulation, 2007.

[20] Keshav Dahal, Alamgir Hossain, Benzy Varghese, "Scheduling in Multiprocessor System Using Genetic Algorithms", 7th Computer Information Systems and Industrial Management Applications, IEEE 2008.

[21] Sachi Gupta, Gaurav Agarwal, "Task Scheduling in Multiprocessor System Using Genetic Algorithm", Second International Conference on Machine Learning and Computing, IEEE 2010.

[22]Adel Manaa and Chengbin Chu, "Scheduling multiprocessor tasks to minimize the makespan on two dedicated processors", European Journal of Industrial Engineering, pp. 265 – 279, Volume 4, Number 3 / 2010.

[23] Amir Masoud Rahmani and Mojtaba Rezvani, "A Novel Genetic Algorithm for Static Task Scheduling in Distributed Systems", 2009.

[24] Carnegie-Mellon, "Genetic Algorithms and Their Applications", Proc. of the First Int. Conference, July 24-26, 1985.

[25] Dr. Franz Rathlauf, "Representations for Genetic and Evolutionary Algorithms", 2nd edition, @ Springer. 2006.

[26] E. Hou, R. Hong, and N. Ansari, "Multiprocessor scheduling based on genetic algorithms", Dept of ECE, New Jersey Institute of Technology, Technical Report, Aug. 1990.

[27] Forrest, Stephanie. "Genetic algorithms: principles of natural selection applied to computation", Science, vol.261, 1993.

