International Journal of Technology and Engineering System (IJTES)
Vol 7. No.2 2015 Pp. 172-175
©gopalax Journals, Singapore
available at : www.ijcns.com
ISSN: 0976-1345

TASKS SCHEDULING IN A CUSTOMIZED APPROACH USING GENETIC ALGORITHM

[1]R.SUKANYAM. [2]RATHNA DEVI
Student, Department of IT , Assistant Professor,Department of IT
KCG College of Technology
Chennai,TamilNadu
[1]suks_ya@yahoo.com , [2]rathnadevi.muthiah@gmail.com
[1]9884945027 , [2]9884427048

ABSTRACT

Efficient task scheduling is of high criticality for obtaining high performance in heterogeneous multiprocessor systems. Since task scheduling is a NP-hard problem, The Genetic Algorithm, an Evolutionary Algorithm which make use of techniques inspired by evolutionary biology such as inheritance, mutation, selection and crossover that is capable of producing optimal solutions. An approach for categorizing the tasks as Hard Real-Time Tasks (critical tasks that need to be completed on time with high rates of confidentiality) and Soft Real-Time Tasks (tasks that can be completed with certain delay and still can be efficient in its own way) before they are scheduled is applied. From the results observed the efficient processor for a particular combination of the tasks is determined thus producing customized results for each of the tasks.

Keywords-Task Scheduling, NP-hard problem, Genetic Algorithm, Hard Real-Time Tasks, Soft Real –Time Tasks.

I.INTRODUCTION

Grid Computing is a set of interconnected computer systems where the machines utilize the resources collectively. There is one main computer which distributes the information and tasks to a group of networked computers to accomplish a common goal. A grid works on various tasks within a network, but is also capable of working on specialized applications. A collection of servers clustered together to work out a common problem forms a grid environment. The computers joined to form a grid may have different hardware and operating systems.

After scientists became disillusioned with classical and neo-classical attempts at modeling intelligence, they found other sources. Two prominent fields arose, neural networking and parallel processing It is the latter that this easy deals with the genetic algorithms and genetic programming.A Genetic algorithmis a search technique used in computing to find true or approximate solutions to optimization and search problems. These algorithms are categorized as global search heuristics. Evolutionary algorithms use techniques inspired by evolutionary biology such as Representation, Initialization, Fitness function, Selection, Crossoverand Mutation.Theevolution usually starts from a population of randomly generated individuals and happens in generations. In each of the generations, the fitness of every individuals in the population are evaluated, multiple individuals are selected from the current population and modified to form a new population.This population is then used in the next iteration of the algorithm. The algorithm terminates when either the maximum number of generations has been produced, or the satisfactory fitness level has been reached. If the algorithm terminates due to a maximum number of generations, a satisfactory solution may be reached.

II-RELATED WORK

There are many heuristic based methods for solving multiprocessor task scheduling approach. The best heuristic approaches are based on task the list technique [1]. In this technique, a list of descending priority ordered tasks is made. A task is selected from the top of the list and assigned to the processor.In static scheduling algorithms the list is not updated with new ordering at run time while the dynamic approaches do. Scheduling algorithms using t-level (top level) and b-level (bottom level) attributes for assigning priority to the processors have been proposed. There are several other Heuristic methods (Level-based Heuristics) such as HLFET [2] ((Highest Level First with Estimated Times), HLFNET (Highest Levels First with No Estimated Times), Random

(the assigned tasks priority are random), SCFET (Smallest Co-levels First with Estimated Times) and SCFNE (Smallest Co-levels First with No Estimated Times), CP/MISF [3] (critical path/most immediate successors first), HNF (Heavy Node First) and WL[1](Weighted Length). All of these attributes act based on level concept in the DAG and without consideration of communication cost. Moreover, DF/IHEZ [4] (Edge-zeroing) algorithmLC[5] (Linear Clustering) algorithm, DSC [6] (Dominant Sequence Clustering) algorithm, MD [7] (Mobility Directed),DCP [8] (Dynamic Critical Path), ETF (Earliest Task First) [9]and greedy heuristics [10] are other heuristic methods.
These heuristic based methods are not considered as intense as before as they do not have good result in all cases. Therefore research on the combinatorial optimization techniques such as meta-heuristics and hybrid methods are going on.

III-PROBLEM STATEMENT

There are a variety of tasks that need to be executed in the Grid Computing in the real world with efficient execution time and best utilization of the resources available in the Grid Environment for which a customized approach is followed in this paper to achieve better off results to the original grid computing. A solution that can be best utilized for providing efficient make span rate along with best use of the Grid resources is bought about to light. Thus the solution determined should fulfill both the criterion.

IV-PROPOSED SOLUTION

An approach for categorizing the tasks as Hard Real-TimeTasks (critical tasks that need to be completed on time with high rates of confidentiality) and Soft Real-Time Tasks (tasks that can be completed with certain delay and still can be efficient in its own way) before they are scheduled is applied. A combination of the tasks with their corresponding processor speed is tabulated. From the results observed the efficient processor for a particular combination of the task is determined thus producing customized results for each of the tasks.
The concept is applied to find an optimal solution for scheduling different kinds of applications namely. This is achieved by categorizing the applications and identifying the suitable processor for scheduling the applications thereby reducing their make span time. The results obtained from the existing and the proposed concepts are compared to determine the efficient scheduling of tasks.

In order to achieve a customized approach in the tasks scheduling, the tasks are initially categorized as Hard Real-Time Tasks and Soft Real –Time Tasks based on their rate of confidentiality and the also the rate at which the tasks need to be completed.

There are various kinds and various streams of engineering disciplines for which the tasks need be scheduled with high efficiency. Some of the disciplines include,

1. Image Processing
2. Marketing
3. Biometrics applications
4. Robotics
5. Media Files
6.

HARD REAL-TIME TASKS
BIOMETRICS
BANKING TRANSACTIONS
ROBOTICS
STOCK EXCHANGE
ASTRONOMICAL

SOFT REAL –TIME TASKS
NETWORK SECURITY
MEDIA FILES
MARKETING
DATA MINING
IMAGE PROCESSING

The Processors are then identified for the tasks that are to be scheduled using the customized approach in scheduling using GA.
Let us consider the most widely used processors from the Intel Family namely the I3, I5, I7 processors.

I3 PROCESSOR

- Clock rate of 2.0Ghz to 2.50GHz
- SLOWER compared to i5 and i7
- Capable of simple tasks
- Applied for SS combination
- *EXAMPLE-SS-Mediafile:Image Processing*

I5 PROCESSOR

- Clock rate of 2.67GHz to 2.87GHz
- INTERMEDIARY speed between i3 and i7
- Capable of mediocre level applications
- Applied in both SH and HS combinations
- *EXAMPLE- HS-Stock Exchange :Marketing SH-Image Processing :Security*

I7 PROCESSOR

- Clock rate of 1.7GHz to 3.7GHz
- Comparatively FASTER to i5 and i3 processors
- Capable of executing high-end applications
- Applied to HH Combination of Tasks
- *EXAMPLE –Data mining : Media Files*

Once the Processors are identified then the scheduling is done using the Genetic Algorithm and the results are tabulated for comparison later.
The next phase is performing the conventional tasks scheduling using the Genetic Algorithm from where the similar results of the execution time is obtained and tabulated for obtaining the better off technique for scheduling the tasks.

Genetic Algorithm

In a genetic algorithm, population of the candidate solutions (called individuals or phenotypes)of an optimization problem evolves towards a better solution. Each candidate solution has a set of properties (its chromosomes or genotype) which can be easily mutated. Traditionally, the solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.

The evolution usually starts from a population of randomly generated individuals which is an iterative process, with the population in each of the iteration called as a generation. In each generation, the fitness of every individual in the population is evaluated and the fitness is usually the value of the objective function in the optimization problem being solved. The more fit the individual is the probability of selecting from the current population is more and each individual's genome is modified to form a new generation. The new generation of candidate solutions is then taken in the next iteration. Genetic Algorithm is terminated when a maximum number of generations have been produced.

The pseudo code followed for calculating the GA is,

Initialization: Generate the initial population P of n individuals

Fitness: Evaluate the fitness of each individual of the population P.

while (stopping criteria not met)

{

Selection: Select a subset of individuals from P.

Crossover: Perform cross-over on each of the selected individuals with probability P_c

Mutation: With probability P_m, mutate each individual

Fitness: Evaluate the fitness of each individual in the new population.

}

*return*Best found solution

i) Representation

Representation determines the type of operators that could be used to ensure the evolution of the individuals and also impact on the feasibility of individuals. . Each individual is represented as a vector, of size equal to the number of tasks to be scheduled and entry in position i indicates the processor to which task i is assigned. The values in this vector are in the range (1, number of processors). It is to be noted that a processor number can appear more than once.

ii)Initialization

During initialization, a set of individuals are generated randomly from a uniform distribution to form a solution space of the desired population size.

iii)Fitness Function

The fitness function calculated is for the make span time/execution time. It is calculated by,

Make span time=

where$F_j$denotes the time when task j finalizes, schis the set of all possible schedules and jobs denote the set of all jobs to be scheduled.

iv)Selection

Selection operators are used to select good individuals to form mating poolsto which the crossover operators will be applied. For this problem a Binary Tournament Selection is used. The Selection is done by making two randomly selected individuals to participate in the tournament and choose the best among them.[10]

v)Crossover

Crossover is achieved by selecting individuals from the parental generation and interchanging their genes, to obtain new individuals. Depending on the representation of individuals, a single point crossover and fitness based crossover is used for optimization.In single point crossover, the crossover operation selects a random pair of individuals and chooses a random point in the first individual.Every individual is considered for crossover with a certain probability.

vi)Mutation

Mutation alters one or more gene values in an individual from its initial state resulting in new individuals helping to arrive at better solution than was previously possible. This also prevents the population from stagnating at any local optima. Mutation occurs during evolution according to a user-definable mutation probability usually set fairly low otherwise the search will turn into a primitive random

V-CONCLUSION

In this paper we have proposed a method for categorizing the tasks before they are scheduled using the Genetic Algorithm to provide a customized approach for the tasks which are scheduled in the Grid environment.

REFERENCES

[1].Shirazi, B., Wang, M., Pathak, G.: Analysis and evaluation of heuristic methods for static task scheduling. J. Parallel Distrib. Comput. 10(3), 222–232 (1990)

[2].T.L. Adam, K. Clhandy and J. Dickson, "A Comparison of List Scheduling for Parallel Processing Systems," Communications of the ACM, vol. 17, no. 12, pp. 685-690, Dec. 1974.

[3].Kasahara, H., Narita, S.: Practical multiprocessing scheduling algorithms for efficient parallel processing. IEEE Trans. Comput. 33, 1023–1029 (1984)

[4].Sarkar, V.: Partitioning and Scheduling Parallel Programs for Multiprocessors. MIT Press, Cambridge (1989)

[5].Kim, S.J., Browne, J.C.: A general approach to mapping of parallel computation upon multiprocessor architectures. In: Proceedings of International Conference on Parallel Processing, pp. 1–8 (1988)

[6].Yang, T., Gerasoulis, A.: List scheduling with and without communication delays. Parallel Comput. 19(12), 1321–1344 (1993)
[7].Wu, M.Y., Gajski, D.D.: Hypertool: a programming aid for message-passing systems. IEEE Trans. Parallel Distribut. Syst. 1(3), 330–343 (1990)
[8].Kwok, Y.-K., Ahmad, I.: Dynamic critical path scheduling: an effective technique for allocating task graphs to multiprocessors. IEEE Trans. Parallel Distrib. Syst. 7(5), 506–521 (1996)
[9].Hwang, J., Chow,Y.,Anger, A., Lee, C.: Scheduling precedence graphs in systemswithinterprocessor communication times. SIAM J. Comput. 8(2), 244–257 (1989)
[10].Kermia, O., Sorel, Y.: A Rapid Heuristic for Scheduling Non-Preemptive Dependent Periodic Tasks onto Multiprocessor. ISCA PDCS, pp.1–6 (2007)