ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Advanced Task Scheduling in Heterogeneous Multiprocessor Systems Using Evolutionary Algorithms

N Srinivasu[1], G.Krishna Chaitanya[2]

[1*]Professor, Dept. of CSE, K L University, Vaddeswaram, Guntur District, India

[2]Assistant Professor, Dept. of CSE, K L University, Vaddeswaram, Guntur District, India

ABSTRACT: Scheduling is the process which improves the performance of parallel and distributed systems. Multiprocessor is a powerful computing for real-time applications and their high performance is purely based on parallel and distributed systems. Number of scheduling tasks in homogeneous and heterogeneous multiprocessor systems is an important problem in computing because this problem is a NP-hard problem.The execution time for individual tasks in a network are specified in a vector and refer only to the computation time. To increase the performance, reducing the processing time of the program is the main aim of scheduling. In this process we are computing rank for all nodes starting from exit node. An advanced task scheduling in heterogeneous multiprocessor by using P-HEFT Algorithm is proposed to minimize the execution time and increase the processor utilization and load balancing for more no of tasks.

KEYWORDS: Scheduling, Tasks, Minimize, Maximize, DAG, Rank, Priority, Computation cost, successor node.

I.INTRODUCTION

The issue of scheduling a task graph of a parallel program onto a parallel and distributed processing system is an all around characterized NP-hard issue that has gotten a lot of thought, and it's viewed as a standout amongst the foremost troublesome problems in parallel computing [1]. The programming issue has been attended in an exceedingly few applications, as an example, information reworking, info frameworks, climate estimating, image getting ready, liquid stream, methodology management, monetary aspects, operation exploration and continuous high speed incitements of propellant frameworks. The digital computer assignment designing issue thought-about during this paper is in light-weight of the settled model, that is that the execution time of undertakings and therefore the info correspondence time between assignments that square measure appointed; and therefore the directed acyclic task graph (DAG) that speaks to the priority relations of the errands of a multiprocessing system [2]. the target of such a computer hardware is to portion undertakings to accessible processors specified priority requirements between assignments square measure consummated and therefore the general amount of your time required to execute the entire program, the calendar length or build compass, is decreased .

Many scheduling algorithms are represented for solving scheduling problem with the purpose of minimizing the execution time of parallel program and correspondence value between tasks within the graph. One necessary variety of these scheduling algorithms is heuristic algorithms [3]. Hereditary algorithms are one necessary variety of heuristic algorithms. As indicated by the looking out capability of these calculations, they will represent optimum scheduling in examination with differing types of scheduling algorithms [4, 5, 6].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

II SCHEDULING

Scheduling is the process which allocates the tasks to the available resources for the output at particular time. The aim of the scheduling is to link the tasks on the processor and make them more reliability. The task scheduling is divided into static and dynamic[13][14][15][16].

In static scheduling, execution process of each task is to be known in advance. This type of scheduling takes place during compile time. This is known as offline deterministic scheduling. The main problem in static scheduling is NP completes [7].

In dynamic scheduling, tasks are assigned to processors upon their arrival and scheduling decisions must be made at run time. Scheduling decisions are based on dynamic parameters that may change during run time. In dynamic scheduling tasks can be reallocated to other processors during the run time [7].Dynamic scheduling is flexible and faster than the static scheduling.

III RELATED WORK

A. *Monte Carlo Algorithmic Program* dynasty Zheng and RizosSakellariou planned a calculation referred to as Monte Carlo approach [8].This algorithmic program utilizes static programming technique. The principle style is to reduce the makespan. This algorithmic program avoids the complicated computation attached variant applicable to any random distribution. Limit qualities square measure utilized to contour the design. The probabilistic conveyance is connected to reduce the makespan.

B. *Cbhd Algorithmic Program*Abdelkader and Omara planned bunch based mostly HEFT with Duplication, used for flourishing designing and mapping for undertakings on heterogeneous frameworks, to interrupt the errands into subtasks and within the in the meantime work with distinctive assignments [9]. This algorithmic program is taken into account as mix between the Triplet bunch algorithmic program, by bunching errands into reticulate cluster, and also the HEFT computation, the assignment in each bunch is asked for in perspective of rank to upgrade the execution of booking designing with burden leveling. The rule target is to reduce the execution time, expand processor utilization and also the store dynamical between the processors.

C. *Colossal Execution And Energy Economical Task Programming Algorithmic* program Ilavarasan and Monoharan [10] planned another calculation that focuses on the diminution of timetable length and power use. This reckoning has 2 stages notably, aggregation time stage and run time stage. The amassing time stage has 3 stages, for example, level sorting, task prioritization and processor determination. Within the thick of the run time, to screen the vitality this calculation reschedules the endeavor to the best center points from the possessed assignments. This reckoning performs through and thru higher than the customary calculation.

D. *Obliged Earliest End Time (Ceft) Algorithmic Program* forced Earliest end Time was planned by Minhaj Ahmad Khan [11].This novel methodology is to grant higher booking to heterogeneous frameworks utilizing the thought of compelled basic ways that (CCPs).Once the CCP in DAG square measure discovered, the undertakings square measure reserved utilizing the finishing time of the full CCPs. this system encourages in delivering timetables with shorter build compass and works with very little elaborateness.

E. *Sorted Nodes InLeveled Dag Division (Snldd) Algorithmic Program*Nirmeenet et al [12] planned the algorithmic program referred to as high performance task programming algorithmic program. This kind of calculation partitions the DAG into distinctive levels and each level is sorted in plunging request as per their interval, that lessens the reliance between the assignments. Static endeavor booking is connected in heterogeneous frameworks with planned range of processors. This kind might deliver high caliber of trip booking. The reckoning time of all assignments in DAG is registered once; consequently the run time overhead is distributed with. The graceful time of processors square measure in addition decreased, on the grounds that DAG square measure leveled and appointed to the processors.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

IV.EXISTING SYSTEM

In this work, we consider that the individual networks consist of number of tasks whose precedences are reflectedin the communication delay matrix. The communication delay matrix for a network is a Nt*Nt matrix (N is the no.of tasks) whose (i,j) element gives us the communication delay incurred in starting a task j after completing task i,provided i precedes j and they are scheduled on different processors. In addition, our model assumes that each network arrives as a batch of tasks with a specified time period for the entire batch. The execution times for individual tasks in a network are specified in a vector and refer only to the computation time. It is also assumed allthe networks are acyclic in nature so that the communication delay matrix essentially becomes an upper triangular matrix.

V.PROPOSED SYSTEM

A. *P-Heft Algorithm*

The main objective of this algorithm to minimize the completion time of the each task and increase the performance of the P-Heft Algorithm.For inputs HEFT takes a set of tasks, represented as a directed acyclic graph, a set of workers, the times to execute each task on each worker, and the times to communicate the results from each job to each of its children between each pair of workers.
To arrange the priority to a task n_i The P-HEFT algorithm uses the upward rank of the task called rank r. The task list is generated by sorting the nodes to the decreasing order of the rank r. If two nodes to be scheduled and having same rank r, one of them is selected parallelly.

B. *P-Heft Algorithm Steps*

1. Compute rank r for all nodes by traversing graph upward, starting from the exit node.
2. Sort the nodes in a list by non increasing order of rank r values.
3. While there are unscheduled nodes in the list
4. do
5. Begin
6. Select the first task ni in the list and remove it.
7. Assign the task ni to the processor pj that minimizes the (EFT) value of ni .

VI.EXPERIMENTAL RESULTS

To evaluate the performance of the P-HEFT algorithm minimize the execution time and increase the processor utilization and load balancing for more no of tasks. , the algorithm is implemented and tested on a PC with a 2.8 GHz Pentium 4 processor, and 1 GB main memory. In the results two graphs are generated Fig-1 shows the processor fluctuations and load balancing by using the HEFT algorithm. Fig-2 shows the constant maintenance of the processor and x-axis indicates the time in sec and y-axis indicates the no of tasks.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

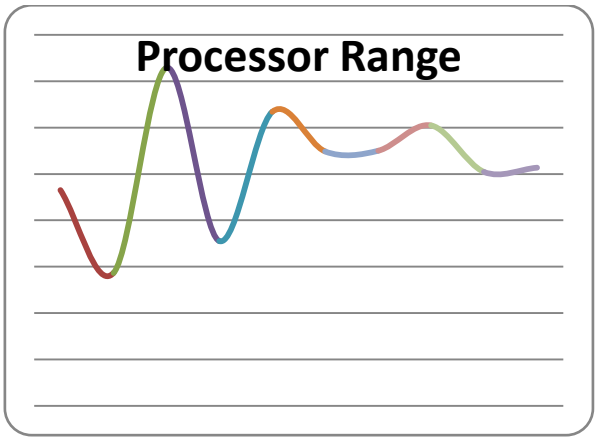

Fig-1, Processor Performance with HEFT

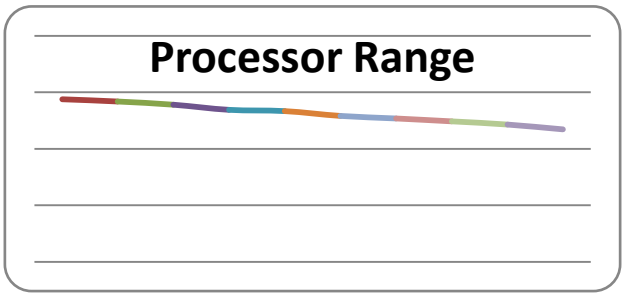

Fig-2, Processor Performance with P-HEFT

DIRECTED ACYCLIC GRAPH (DAG)

It is the collected of vertices and directed edges. Every edge haves one vertex to another. These does not contain cycles. From the P-HEFT point of view the DG is represented as G= (V, E). Where V is set of nodes and E is a set of directed graph. Source node of the edge is called parent node and sink node is called child node.The DAG forms partial order, any partial order can be represented by using reachability. Additionally, DAGs are used as a space-efficient representation with overlapping subsequences. DAGs are to represent systems of events or potential events. DAGs are used to model processes, in which data flows in a direction through a network of processors. Here N_2 is parent node for N_4, N_5. N_3 is Parent node for N_6, N_7.

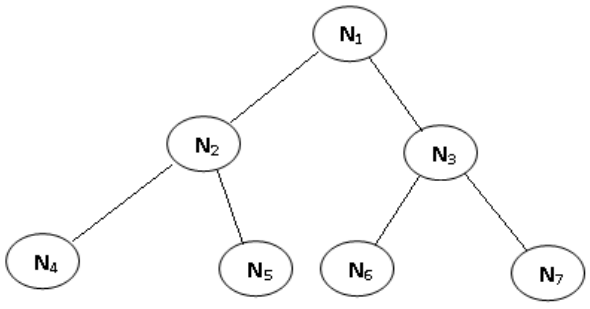

Fig: 3, DAG-Directed Acyclic Graph

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

HEFT
HEFT executes in two segments:

PRIORITIZING TASKS

In the first segment each task is given a priority. The priority of each task n_i is usually designated to be its "upward rank" which is defined recursively as follows.

$$rank_u(n_i) = \overline{w_i} + \max_{n_j \in succ(n_i)} (\overline{c_{i,j}} + rank_u(n_j))$$

where n_i represents the i^{th} task, $\overline{w_i}$ is an average computation cost of job i among all the workers, $succ(n_i)$ is the set of all jobs that immediately depend on task n_i, and $\overline{c_{i,j}}$ is the average communication cost of the variables transferred between jobs n_i and n_j between all pairs of workers. Note that the computation of $rank_u(n_i)$ depends on the computation of the rank of all its children. The upward rank is meant to represent the expected distance of any task from the end of the computation. For averaged quantities like $\overline{w_i}$ different averages may provide different results.

VII ASSIGNING TASKS TO WORKERS

In the second segment undertakings are conveyed to specialists. Since all errands are having need and we consider and arrange each one, starting with the most elevated thought. The undertaking with the most elevated need for which each ward task has finished is moved toward the specialist which will achieve the soonest finish time of that endeavor. This fulfillment time depends on upon the correspondence time to send each imperative data to the specialist, the count time of the endeavor on the worker, and the time when that processor gets the opportunity to be open.
The advantage of P-HEFT is efficient and best make span.

VIII CONCLUSION

In this paper, a new version of HEFT called P-HEFT for parallel task is proposed. This algorithm is proposed minimize the execution time and increase the processor utilization and load balancing for more no of tasks.This P-HEFT obtained the best makespan for multiple processor task scheduling.through this algorithm we can reduce NP-Hard problem.By this algorithm,we can efficiently reduce the processing time in multiprocessor systems. Here we have used two segments mainly that are Prioritizing tasks, Assigning tasks to Workers. According these two segments our proposed works better compare with previous one.

REFERENCES

[1] J. Ullman, "NP-Complete Scheduling Problems," J. Comp. Sys. Sci., 10, 1975.
[2] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Mich., 1975.
[3] R. Hwang, M. Gen and H. katayama. A comparisonof multiprocessor task scheduling algorithms withcommunication cost. Computers & Operational research, 35:976-993, 2008.
[4] R. Armstrong, D. Hensgen and T. Kidd.Therelative performance of various mapping algorthmsis independent of sizable variances in run-time predictions. 7th IEEE Heterogenouscomputingworkshop (HCW'98), 1998.
[5] R. L.Haupt, S.E. Haupt. Practical geneticalgorithms.Wiley interscience publication.Second edition, 2004.
[6] S. Parsa, H. Eizadkhah, A.Hosseinzadeh, Thecomposition of migration automata and geneticalgorithm for task graph scheduling In multiprocessor architecture. 3th internationalconference of information and knowledgetechnology. September 2007. (In Persian).
[7] Y.K. Kwok and I. Ahmad. 1999. "Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors", ACM Computing Surveys, Vol. 31, No. 4, pp. 406 – 471.
[8] Wei Zheng and RizosSakellariou. 2013."Stochastic DAG scheduling using a Monte Carlo approach", J.ParallelDistrib.Comput, pp. 1673 –1689.
[9] DoaaM.Abdelkader and FatmaOmara. 2012. "Dynamic task scheduling algorithm with load balancing for heterogeneous computing

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

system", Egyptian International Journal, pp.135 – 145.
[10] Ilavarasan and Monoharan. 2010. "High performance and energy efficient task scheduling algorithm for Heterogeneous mobile computing system", International Journal of Computer Science and Information Technology, Vol.2.
[11] Minhaj Ahmad Khan. 2012. "Scheduling for heterogeneous systems using constrained critical paths", Parallel Computing, pp. 175 – 193.
[12] Nirmeen A. Bahnasawy, Magdy A. Koutb,MervatMosa and FatmaOmara. 2011."A new algorithm for static task scheduling for heterogeneous distributed computing systems", African journal of Mathematics and Computer Science Research,Vol.4(6), pp.221 – 234.
[13] Siva NageswaraRao,G.,2014.Comparision of Round Robin CPU Scheduling Algorithm with Various Dynamic Time Quantum .Int. J.Appl.Eng.Res.,9(18):2749-2760.
[14] Siva NageswaraRao,G.,2015.A New Proposed Dynamic Dual Processor Based CPU Scheduling Algorithm.Int.J.Theoritical and Applied Info.tech. 73(2):1992-8645.
[15] Siva Nageswara Rao,G.,2014.An enhanced dynamic round robin CPU scheduling algorithm.Int. J.Appl.Eng.Res.,9(14):2749-2760.
[16] Dr.NSRinivasu,An Augmented Dynamic Round Robin CPU Scheduling Algorithm,Int,J.Theoritical and Applied Info.tech. :1992-8645.

BIOGRAPHY

Dr.N.Srinivasu is Professor of computer science department at KL University. His area of interest is Cloud computing, Bigdata, soft computing.

G.Krishna Chaitanya is Assistant Professor of computer science department at KL University. His area of interest is cloud computing..