Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 2 (2017) pp. 171-183
© Research India Publications
http://www.ripublication.com

Dual objective Dynamic Scheduling Algorithm (DoDySA) for Heterogeneous Environments

Dr. D.I.George Amalarathinam[1] and A.Maria Josphin[2]

[1]Dean of Science & Director –MCA, Jamal Mohamed College, Tiruchirappalli, Tamilnadu,India.

[2]Research Scholar, Jamal Mohamed College, Tiruchirappalli, Tamilnadu, India.

Abstract

The performance of the multiprocessor system depends on how the tasks are scheduled in multiprocessors. The primary goal of tasks scheduling in multiprocessor system is to minimize makespan and to maximize processor utilization. The application program is represented by Directed Acyclic Graph (DAG). Scheduling is a process that maps and manages the execution of dependency tasks on the parallel processing. The proposed algorithm for task scheduling in multiprocessors system has Dual objectives. The proposed algorithm has been implemented with arbitrary graph and application graph. The performance of the DoDySA algorithm is measured based on scheduling length and processor utilization.

Keywords: Task Scheduling, Directed Acyclic Graph, Multiprocessor, makespan and Heterogeneity

INTRODUCTION

Multiprocessor have emerged as a powerful computing means for running real time applications, especially where a uni-processor system would not be sufficient enough to execute all the tasks by their deadlines. The high performance and reliability of multiprocessors have made them a powerful computing means in time-critical applications [1]. In heterogeneous computing, an important research problem is how to assign processor to the tasks and to order the tasks for execution on the processor. The task scheduling in multiprocessor system is called multiprocessor scheduling.

Scheduling of multiple tasks on the parallel system is proved to be NP-Complete problem [2]. This problem is considered as one of the challenging problem in the parallel computing. Here the application program is represented by Directed Acyclic Graph (DAG). The main objective of this proposed system is to minimize the overall schedule length and to maximize processor utilization.

The main objective of scheduling is to minimize the completion time of a parallel application by properly allocating the tasks to the processor and maximum processor utilization. Scheduling allocates suitable processor to task graph so that the execution can be completed to satisfy objective functions essential by users.

This paper presents a task scheduling algorithm using DoDySA method at each level of a given DAG. Allocation of tasks on the processor is based on Earliest Stating Time (EST) and Earliest Finishing Time (EFT). The results show that DoDySA algorithm performs better in comparison to DLS, TDSA and DyDupSA Algorithms. This proposed scheduling algorithm is dynamic in nature because it is focused on inter process communication, task dependency and run time allocation.

Most of the scheduling algorithms are mainly focusing on optimizing criteria namely schedule length or makespan (i.e total completion time of exit task) and advance reservation of resources. The proposed algorithm concentrates on dual objective functions.

The result part of the paper is organized as follows: Section 2 gives a related work and proposed algorithm is described in section 3.Evaluation and Results are analyzed in section 4 and finally conclusion is given in section 5.

RELATED WORK :

Ranjit Rajak, et al [8] proposed an algorithm which is used in multiprocessor environment. They discussed about task scheduling, components and types of task scheduling. The main objective is to get minimum scheduling length.

Guoqi Xie, et al [9] proposed an algorithm called a novel and high-performance DAG task scheduling algorithm called HSV (Heterogeneous Select Value) algorithm for heterogeneous networked automobile electronic systems. They proposed the concept of heterogeneous upward rank value, heterogeneous priority rank value and heterogeneous select value for task priorities ordering and processor selection respectively. The main purpose of this algorithm is to reduce scheduling length and produce minimum Schedule Length Ratio (SLR).

Chi-Yeh Chen et al [10] proposed a work called HAAS (Heterogeneous allotment aware scheduling algorithm) which used two-phase algorithm. In first phase, a favourable allotment is determined using a linear program formulation and a rounding procedure. In second phase, a scheduling method that is based on the expected

execution time and the communication time is implemented.
Sukhjit Singh et al [11] proposed an algorithm called A Heterogeneous Static Hierarchical Expected Completion Time Based Scheduling Algorithm in Multiprocessor System. This paper presents a new scheduling technique for the non-preemptive heterogeneous systems. The objective of the algorithm is to minimize the schedule length, speedup and efficiency.

ShuliWang et al [12] proposed a new reliability - aware task scheduling algorithm called RMSR in heterogeneous computing systems. The aim of this algorithm is to maximize the system reliability based on replication. A task reliability threshold is used when the replication executes.

Jiyeon Lee et al [13] proposed an algorithm called Thread- level priority assignment in global multiprocessor scheduling for DAG tasks. It is a real-time scheduling algorithm, according to the unit of priority assignment, from task-level to thread-level. This is the first approach to the problem of assigning task-wide, thread-level fixed-priorities for global parallel task scheduling on multi processors. The proposed thread-level priority assignment can improve schedulability, significantly, compared to its task-level counterpart.

Kun Cao et al [14] proposed an algorithm called Static Thermal-Aware Task Assignment and Scheduling for Makespan Minimization in Heterogeneous Real-time MPSoCs. This paper, proposed a task assignment and scheduling scheme that minimizes the execution time and smoothes the chip temperature. The proposed scheme first assigns the tasks to processors in a way that leads to reduced processors, schedule length and improved processor temperature profiles. It classifies the tasks into hot and cool task category. The cool tasks are executed at the maximal frequencies supported by processors while the hot tasks are executed at scaled frequencies.

PROPOSED DYDUPSA ALGORITHM :
A Parallel program is represented by directed acyclic graph (DAG).The DAG model [3] [4] [5] consists of two tuples G = (V,E) where V = $\{T_1,T_2,...,T_n\}$ finite set of tasks and E = $\{e_{ij}\}$ edges that connects two tasks Ti and Tj. The communication cost between two tasks is zero if they are scheduling on same processor. Precedence constraints are maintained. The task Tj executes if and only if all its predecessor are executed. A source node is called parent node and the sink node is called child node. A task with no parent is called entry node and with no child is called exit node [6].

Task scheduling is classified into deterministic scheduling also known as static scheduling and non-deterministic scheduling which is also known as dynamic scheduling. In deterministic scheduling, all the information about tasks that is communication time, execution time and their precedence constraint are well known in advance. In case of non deterministic scheduling, all the information are changed

during run time. Scheduling decision are based on the dynamic parameters that may change during run time.

Selection of processors is based on the Earliest starting Time (EST) and Earliest Finishing Time (EFT). Scheduling is done based on the EST (ni, pj) and EFT (ni, pj), the earliest execution start time and the earliest execution finish time of task ni on processor pj respectively. The EST and EFT values are computed recursively, starting from the entry task as shown in Equations (1) and (2). In order to compute the EFT of a task all immediate predecessor task ni must have been scheduled [7].

$EST(n_i,p_j) = max\{avail(j), max_{nm \in pred(ni)} AFT(nm)+Cm,i)\}$ Eq.(1)

$EFT(n_i,p_j) = w_{i,j}\ EST(n_i,p_j)$ Eq. (2)

For the entry task $EST(n_{entry},p_j) = 0$ Eq. (3)

where avail (j) in the same processor pj is free and it is ready to execute task n_i.

The inner max block in equation (1) returns the ready time i.e., the time when all data needed by task has arrived at processor. After all tasks in a graph are scheduled, the schedule length (overall completion time) will be the AFT of the exit task (nexit)). If there are multiple exit tasks and the convention of inserting a pseudo exit task is not applied, the schedule length called makespan, is defined as

$makespan = max\ [AFT\ (n_{exit})]$ Eq.(4)

In the proposed algorithm two objective functions are considered, namely maximizing the processor utilization and minimizing the total completion time (makespan). The execution time and idle time of a processor are known from the scheduled list and it is used to calculate the utilization of processor. Processor utilization ((PU (P_j)) of resource P_j is calculated as

$$PU\ (P_j) = \sum_{j=1}^{k} \frac{w(ni,Pj)}{Processor - Speed}$$

where ∑_(j=1)^k (w(ni,Pj)) is the sum of all allotted task's computation time of resourcePj, Processor – speed is the computation speed of the processor. Peudocode of the proposed algorithm of is shown below.

PeudoCode: Dual objective Dynamic Scheduling Algorithm (DoDySA)

1. *Start*
2. *N ← DAG*
3. *NP ← DAG*

4. *MMKSPN←0*
5. *For i←1 to N*
   i. *Tid← random(Parray())*
   ii. *Pid← random(Parray())*
   iii. *CMCST ← random(Parray())*
   iv. *COCST ← random(Parray())*
   v. *PS← random(Parray())*
   vi. *CCT←calculatePS/COCST*
   vii. *EST←derived from CCT*
   viii. *EFT←derived from CCT*
   ix. *MEFTPRid← Choose Minimum EFT processor Id*
   x. *max* [*AFT* (n_{exit})]← Minimization of makespan
   xi. *PU (Pi)* = $\sum_{j=1}^{k} \frac{w(tj, Pi)}{Processor - Speed}$
   xii. *MKSPN←get Makespan by* Actual Finishing time of last tasks
   xiii. *Migrate Processor and reallocates the Task*
   xiv. *MMSPN←get Makespan by* Actual Finishing time of last tasks after *migration*
   xv. *If MMSPN<MKSPN then*
      a. *MMSPN←MKSPN*
      b. *MinMST[Tid] ←MMSPN*
   xvi. *else*
      a. *MinMST[Tid] ←MKSPN*
   xvii. *end if*
6. *next i*
7. *stop*

DoDySA scheduling algorithm has two metrics such as makespan, and processor utilization. These are the parameters taken for comparison with DLS, TDSA, DyDupSA algorithms with arbitrary graph and application graph.

EXPERIMENTAL RESULTS :

The sample DAG with 10 tasks is shown in Figure 1 and the computation cost table is shown in Table 1. Figures 4 shows the comparative study of the makespan and Figure 5 shows processor utilization with various number of tasks and the processors. The main aim of this proposal is to reduce the makespan and maximum utilization of the processors. The processor are reserved in advance therefore all processors are available from the task starting time to finishing time. The processor is a basic computational device or service where tasks are scheduled, allocated or processed accordingly. Processor has their own characteristics such as CPU characteristics, memory capacity, etc. One of the CPU characteristics is the speed of the processor considered for this result. The speed is varied as 1, 1.25, 1.50 and 1.75.

The experiment is conducted for each set of task and compared with DLS, TDSA and DyDupSA algorithms. The possible combination of experiments is conducted with various number of tasks with the processors and results are tabulated after conducting the experiments.

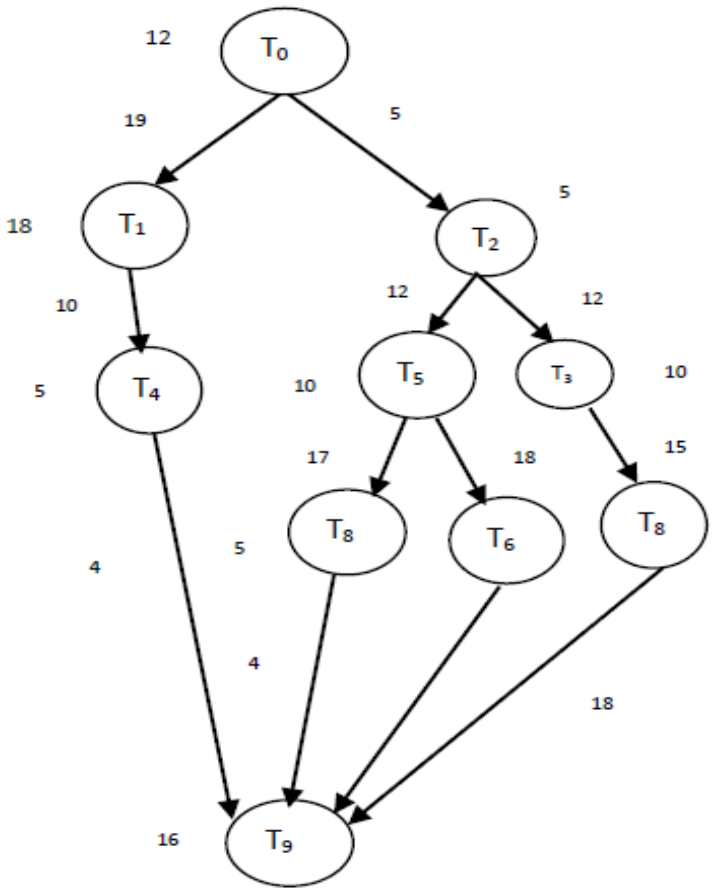


Figure 1: Example DAG with 10 Tasks using arbitrary graph

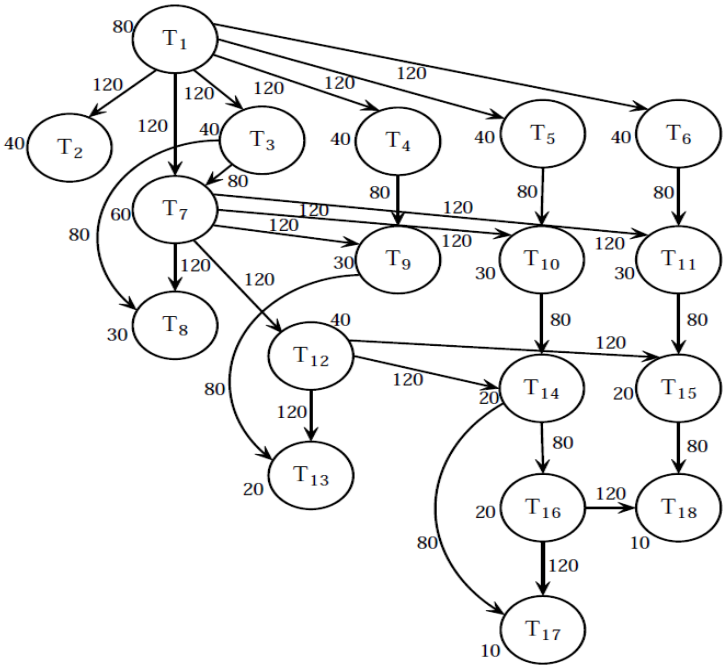


Figure 2: Gaussian Elimination Graph

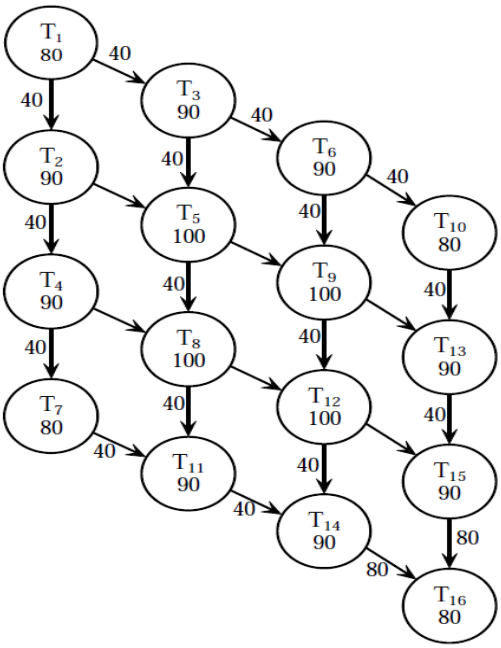


Figure 3: Laplace Equation Graph

A. Makespan

The main performance of a scheduling algorithm is the total execution time of exit task called as makespan. Figure 4 shows the comparison study of makespan with DLS, TDSA and DyDupSA algorithms. In all cases makespan is minimized.

B. Processor Utilization

The processors which have been retained in advance are available till the completion of exit task. Table 1 shows the computation cost table of each task and the processors. Figure 5 shows the comparison study with DLS, TDSA and DyDupSA algorithms and in all cases processor utilization is maximized.

C. Application Graph

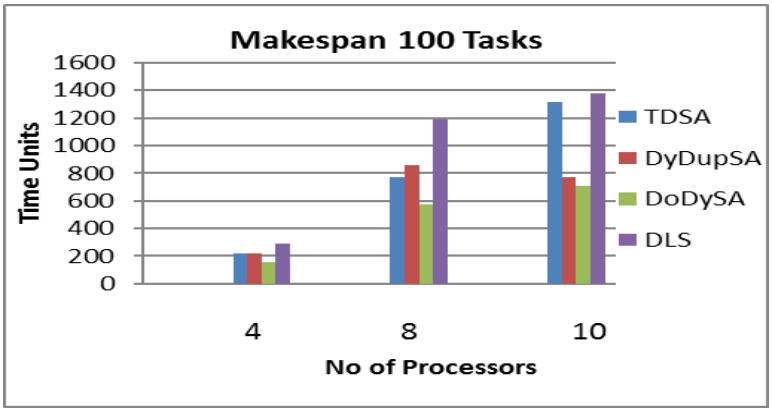
Gaussian Elimination and Laplace Equation graph are the reprsentation of communication intensive problem. It is able to execute in a sequential manner. The communication intensive problems are well suited to adopt duplication based algorithm. The DoDupSA algorithm applied to Gaussian Elimination and Laplace Equation graph which minimizes the makespan and maximizes the processor utilization when compared with DLS, TDSA and DyDupSA algorithms.

Table 1: Computation Cost

Task Id	Processors	EST	EFT
T0	P0	0	8
T2	P0	8	11.33
T1	P1	27	39
T5	P0	11.33	18
T3	P2	23.33	30
T6	P0	18	27.33
T4	P0	49	52.33
T7	P1	45	54.33
T8	P2	35	38.33
T9	P1	54.33	65

Table 2 : Makespan For Arbitrary Graph

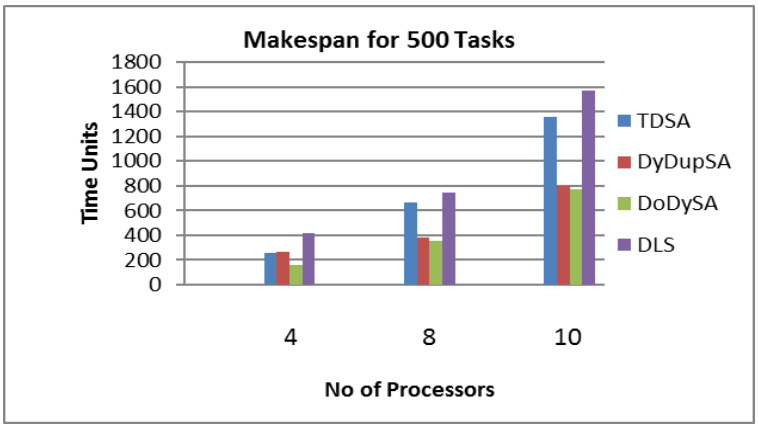
Tasks	Processors	TDSA	Dy DupSA	Do DySA	DLS
100	4	257	266	156	414
500	8	663	385	357	748
1000	10	1356	801	772	1574



a.Arbitrary Graph

Table 3. Makespan For Gaussian Elimination Graph

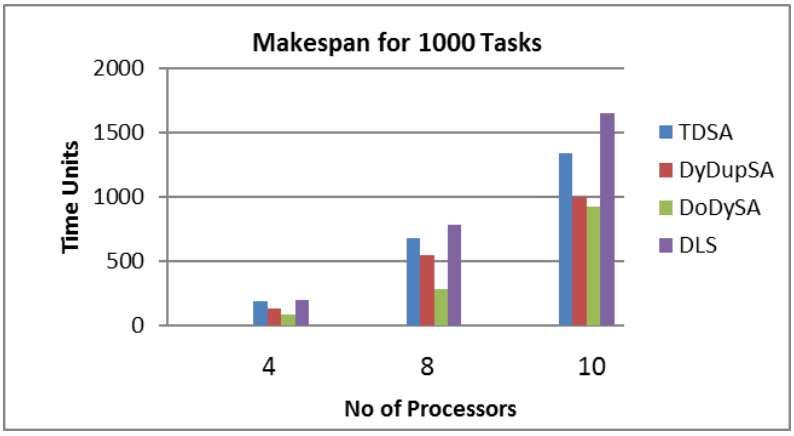
Tasks	Processors	TDSA	Dy DupSA	Do DySA	DLS
100	4	217	218	156	287
500	8	771	856	571	1191
1000	10	1315	771	705	1376



b. Gaussian Elimination Graph

Table 4. Makespan For Laplace Eqaution Graph

Tasks	Processors	TDSA	Dy DupSA	Do DySA	DLS
100	4	191	132	84	203
500	8	684	549	282	789
1000	10	1339	1004	929	1654

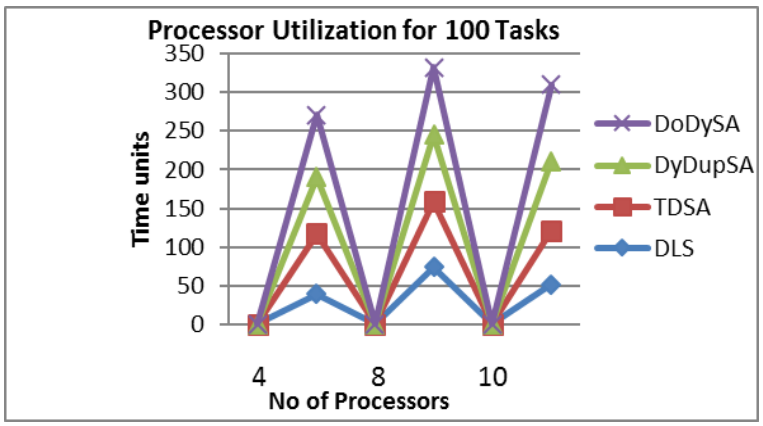


c. Laplace Equation Graph

Figure 4: Comparison study of makespan with various algorithms using Arbitrary and Regular Graph

Table 5: Processor Utilization For Arbitrary Graph

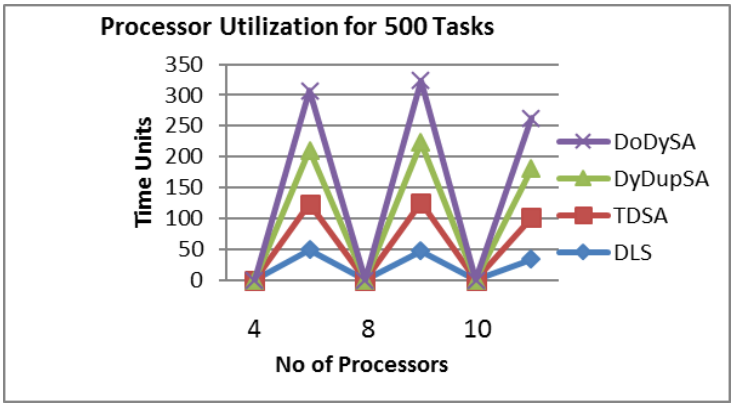
Tasks	Processors	TDSA	Dy DupSA	Do DySA	DLS
100	4	39	78	73	81
500	8	74	85	86	87
1000	10	52	69	90	92



a. Arbitrary graph

Table 6 : Processor Utilization For Gaussian Elimination Graph

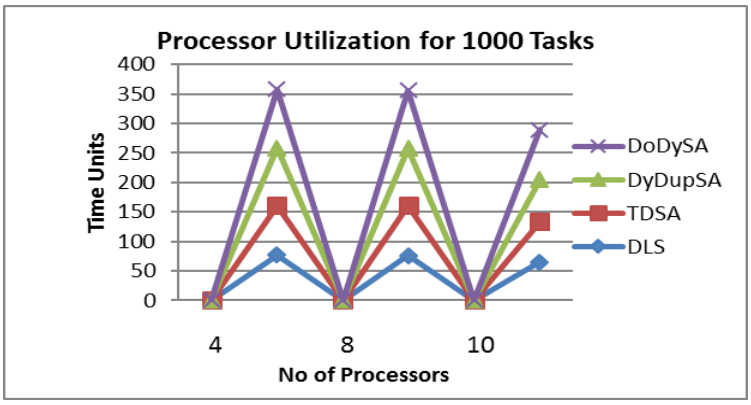
Tasks	Processors	DLS	TDSA	Dy DupSA	Do DySA
100	4	49	74	87	97
500	8	47	78	98	98
1000	10	34	67	79	83



b.Gaussian Elimination Graph

Table 7 : Processor Utilization For Laplace Equation Graph

Tasks	Processors	DLS	TDSA	Dy DupSA	Do DySA
100	4	77	84	73	98
500	8	76	85	86	97
1000	10	65	69	90	85



c. Laplace Equation Graph

Figure 5: Comparison study of Processor Utilization with various algorithms using Arbitrary and Regular Graph

CONCLUSION

Multiprocessor scheduling is one of the most important and challenging in research area. Scheduling is the decision process by which application mechanism are assigned to available processors to optimize various performance metrics. This paper presents a new approach of Dual objective dynamic task scheduling in heterogeneous environment. This Proposed algorithm gives minimum schedule length and maximum utilization of the processors. Comparison study with DLS, TDSA and DyDupSA algorithms in all cases shows that makespan is minimized and processor utilization is maximized. This algorithm has been tested with more number of tasks and shown to produce better result.

REFERENCES

[1] Kwok, Yu-Kwong and Ahmad, Ishfaq: Dynamic Critical Path Scheduling: An effective Techniques for Allocating Tasks Graph onto Multiprocessor, IEEE Transaction on Parallel and Distributed System, Vol. 7, No 5, pp 506-621,1996

[2] Kwok, Yu-Kwong and Ahmad, Ishfaq: Static scheduling Algorithm for allocating Directed Task Graph to Multiprocessors, ACM Computing Surveys, Vol.31, No.4, 1999.

[3] S.R.Vjayalakshmi and Dr.G.Padmavathi, "Multiprocessor Scheduling for Tasks with Priority Using GA", International Journal of Computer Science and Information Security(IJCSIS), Vol .6, No.3, 2009.

[4] Ranjit Rajak , "A Novel Approach for Task scheduling in Multiprocessor System," International Journal of Computer Applications(IJNCAA) Vol.44, No.11, pp. 12- 16, 2012.

[5] F.Kurus Malai Selvi and Dr.D.I.Geroge Amalarethinam, "A Task duplication based efficient Multi- Objective Grid Workflow Scheduling Algroithm", International Jouranl of Advanced Research in Computer Science,Vol.3, No.2, 2012.

[6] Ranjit Rajak & C.P.Katti, "Task scheduling in Multiprocessor System using Fork-Join Method (TSFJ)," International Journal of New Computer Architectures and Their Applications(IJNCAA) Vol.3, No.3, pp. 47- 53, 2013.

[7] Ehsan Ullah Munir, Sajjad Mohsin, Altaf , Muhammad Wasif Nisar, Shoukat Ali ", SDBATS: A Novel Algorithm for Task Scheduling in Heterogenous Computing Systems', IEEE Computer Society, 2013.

[8] Ranjit Rajak, C.P.Katti, Nidhi Rajak " A study of task scheduling in multiprocessor environment", Computer Science and Telecommunications Vol.1, No.41, 2014.

[9] Guoqi Xie, Renfa Li,Xiongren Xiao,Yuekun Chen," A High-performance DAG Task Scheduling Algorithm for Heterogeneous Networked Embedded Systems", IEEE 28th International Conference on Advanced Information Networking and Applications, 2014.

[10] Chi-Yeh Chen," Task Scheduling for Maximizing Performance and Reliability

Considering Fault Recovery in Heterogeneous Distributed Systems", IEEE Transactions on Parallel and Distributed Systems, Journal of latex class files, Vol. 6, No. 1, 2015.

[11] Sukhjit Singh, Nirmal Kaur ,"A Heterogeneous Static Hierarchical Expected Completion Time Based Scheduling Algorithm in Multiprocessor System ",International Research Journal of Engineering and Technology (IRJET) Vol. 03 No. 02, 2016.

[12] ShuliWang, Kenli Li,Jing Mei and Guoqing Xiao, "A Reliability-aware Task Scheduling Algorithm Based on Replication on Heterogeneous Computing Systems ", Journal of Grid Computing, 2016.

[13] JiyeonLeea, HoonSung, Chwaa, JinkyuLeeb, InsikShi,"Thread- level priority assignment in global multiprocessor scheduling for DAG tasks, The Journal of Systems and Software, Vol.113, pp.246–256, 2016.

[14] Kun Cao, Junlong Zhou, Min Yin, Tongquan Wei, and Mingsong Chen,"Static Thermal-Aware Task Assignment and Scheduling for Makespan Minimization in Heterogeneous Real-time MPSoCs", International Symposium on System and Software Reliability, 2016.