THINK INDIA JOURNAL

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Breadthfirst Search Based Task Duplication Using Parallel Computation

Varun Singla
Assistant Professor
Lovely Professional University
varun.17705@lpu.co.in

Amritpal Singh
Assistant Professor
Lovely Professional University
amritpal.17673@lpu.co.in

Ravinder Singh
Assistant Professor
Lovely Professional University
Ravinder.17750@lpu.co.in

***Abstract*—** Scheduling and Execution of parallel tasks of a single job on different processors at the same time in parallel computing environment is very challenging. Various scheduling algorithms have been developed in this field. Generally scheduling algorithms are of two types: Scheduling with task duplication and Scheduling without task duplication. As the time grows, new algorithms have been developed so as to reduce the total time of execution of tasks known as schedule length. We propose Breadth first search based task duplication algorithm for compile-time scheduling of parallel programs on parallel processing systems.Proposed algorithm traverses the Directed Acyclic Graph (DAG) of tasks known as task graph, using breadth first search mechanism, taking advantage of task duplication and task insertion on different processors. The main objective of this scheduling algorithm is to minimize schedule length and time complexity of algorithm with the realistic assumptions. Proposed algorithm has been tried on various task graphs and it provides good results in terms of schedule length and time complexity. To compare its results with other algorithms, sample task graph has been shown in this paper. On comparing the performance of this algorithm with existing algorithms i.e. Papadimitriou &Yannakakis (PY)[1][4] and genetic algorithm i.e. Task Duplication Genetic Algorithm (TDGA) [2], it is found that proposed algorithm gives better performance in terms of schedule length and time complexity.

Index Terms—Parallel Task Execution, Task Duplication, Task Execution in Parallel Environment

I. INTRODUCTION

With the improvement of the multiprocessor systems, communication overhead among processors and tasks is the biggest problem for parallel computing environment. This problem involves execution of parallel tasks of Directed Acyclic Graph (DAG) on parallel processors. Various algorithms have been proposed for this task allocation. Scheduling algorithms are classified into two types: Static scheduling and Dynamic scheduling. In static scheduling, communication cost, computation cost and task dependencies should be known before task execution and in dynamic scheduling, it is not necessarily to be known but some assumptions are made for task execution at run time. Task Duplication means repeteadly allocating some tasks to various processors on which other tasks of that processor depends. Scheduling with task duplication perform better than without task duplication. Now adays, Genetic algorithms [2] are also used to provide better solution for task scheduling problem. The organization of this paper is as follows: section 2 specifies related work extending to section 3 elaborating about Directed Acyclic Graph Model. Starting with section 4 description of proposed algorithm, section 5 continues the output of proposed algorithm concluding with results shown in section 6. Finally section 7 gives conclsions with proposed future work mentioned in section 8.

II. RELATED WORK

A lot of research has been done for efficiently scheduling DAGs on homogeneous environments.

According to Liou [9] scheduling algorithms can be of two types: Static and Dynamic scheduling. In static scheduling, communication cost, computation cost and task dependencies should be known before task execution and in dynamic scheduling, it is not necessarily to be known but some assumptions are made for task execution at run time. This paper work solely depends upon static scheduling.

A general taxonomy for static scheduling algorithms has been reviewed and discussed by Kwok and Ahmad [3]. According to Kwok, scheduling algorithms may be divided into two types: Greedy algorihm and Non Greedy (Iterative) algorithm. Greedy algorithm focuses on minimizing the start time of task on processor. But in case of iterative algorithm, it first finds a solution to the problem and then tries to improve it. Greedy algorithm can be of two types: Scheduling with task duplication and scheduling without task duplication. In case of scheduling with task duplication, various methods are employed to find which ancestor node is to be assigned on the processor of a particular task.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Jefferey proposed PY algorithm [1] (named after Papaadimitriou and Yannakakis) is task duplication based scheduling algorithm, which finds the start time of a task on processor known as e value. This value is calculated for each task on each processor, after that task is allocated to processors after applying some conditions. . The time-complexity of the PY algorithm is $O(V^2 (E + VlogV))$ and its schedule length was 21 for the graph shown in Fig. 1 where E is the edge and V is node i.e. task node.

III. DIRECTED ACYCLIC GRAPH (DAG) MODEL

DAG is a model of parallel program, which consists of set of tasks connected to each other to form a graph such that if task A depends upon task B then there is an edge from B to A and there is no cycle in that graph. Each task is represented by a node. All the nodes are connected by the edges which show dependencies among them. Scheduling is done in this way that the precedence constraints of tasks are preserved. The model consists of set of nodes t1,t2,t3….tv connected with the set of edges E where each edge is denoted by (ti, tj) where ti is the parent node and tj is the child node. The aim of the Scheduling is to minimizing the schedule length. The example of DAG is shown in Fig. 1

IV. BREADTH FIRST SEARCH BASED TASK DUPLICATION ALGORITHM (BFSTD)

To further reduce schedule length, we have proposed a less time complexity (as compared to other algorithms) BFSTD algorithm which is based upon Breadth First Search (BFS) with some modifications. The Breadth First Search based task duplication algorithm concerned with the static scheduling with task duplication. In this algorithm graph will be traversed according to breadth first search. The modification is that in BFS, once a child is traversed through any parent, it is marked as visited so that it will not be visited again but in this algorithm, a task has to be accessed by every parent in the task graph to check the parent node having the highest communication cost for that task. During traversal of nodes, our algorithm adds nodes to the processors simultaneously depending upon some conditions as discussed in Algorithm 1. After entire task is traversed, the schedule length will be calculated. Proposed algorithm outperforms the previous existing algorithms i.e. PY (Papadimitriou &Yannakakis), TDGA (Task Duplication Genetic Algorithm) etc. This algorithm gives less schedule length with lower time complexity.

Algorithm 1. Calculation of Number of processors required for a particular task graph, tasks allocated to different processors and schedule length.

ALGORITHM 1. PROCESSOR NUMBER COMPUTATION
Input: Directed Acyclic Task Graph showing nodes and edges.
Output: 1) No. of processors showing different tasks for parallel execution.
2) Schedule length (SL).

```
Let ti be the current node to be traversed in BFS;
repeat
    Duplicate all nodes from root to ti on ci number of processors depending upon two conditions:
        a) If any child can be reached through longer path then do not duplicate it;
        b) If any of child c has more than one parent then calculate total computation cost w from root to c
           and calculate communication cost cij of c to its current parent
            If w > cij then
                If ti is the parent having highest communication cost from c among all parents of c then
                        Add c to current processor;
                else
                        Leave the child node;
            else
                Add all nodes from root to parent of c having least computation time, to current processor;

            end
until ti is the last leaf node in BFS;
```

V. OUTPUT OF BFSTD

We have applied our algorithm on lots of DAG's to prove its efficiency and it proves to be better than other algorithms. For better comparison of results, we have taken a sample graph shown in Fig. 1 whose schedule length is known for every other algorithm. Fig. 2 shows the output of the algorithm implemented on task graph as shown in Fig. 1, where t1 to t9 are the tasks of DAG and when it is implemented then it requires five processors i.e. p0 to p4. Here is the description of above algorithm applied on task graph of Fig. 1:

a. Task t1 is the parent of the t2, t3, t4, t5 and t7. According to the algorithm t1 is duplicated to four processors namely P0, P1, P2 and P3 and t2, t3, t4, t5 is copied on different processors respectively and Left the task t7, as per the condition 'a' of Algorithm 1, t7 has the longer path when coming from t2.
b. Now according to BFS, t6 will be allocated. Therefore, it is allocated to the processor having its parent nodes i.e. P0. Again t1, t2, t7 gets new processor P4 because during allocation of t7, its ancestor has to be assigned to processor P4.
c. Task t8 has multiple parents i.e. t3 and t4. Now computation cost from the root to task t8 from all possible paths are calculated. Here computational cost through t3 is less than through t4 and communication cost of t4 to t8 is again
less and hence t8 is added below the task t3 and task t4 is communicated to the task t8.
d. Again task t9 has multiple parents, again applying same condition as in previous step, t9 is added below t7 because task t9 has the highest communication cost from t7 and communication is given from task t6 and t8.
e. BFSTDB algorithm having time complexity O(ElogV) generates a schedule having schedule length 17 for DAG shown in Fig. 1.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

TABLE 1
COMPARISON OF RESULTS OF VARIOUS ALGORITHMS

Algorithm Name	Time Complexity	Schedule Length
PY	21	$O(V^2(V+VlogV))$
TDGA	18	$O(V^4)$
LWD	18	$O(V^2)$
BFSTD (Proposed Algorithm)	**17**	**O(ElogV)**

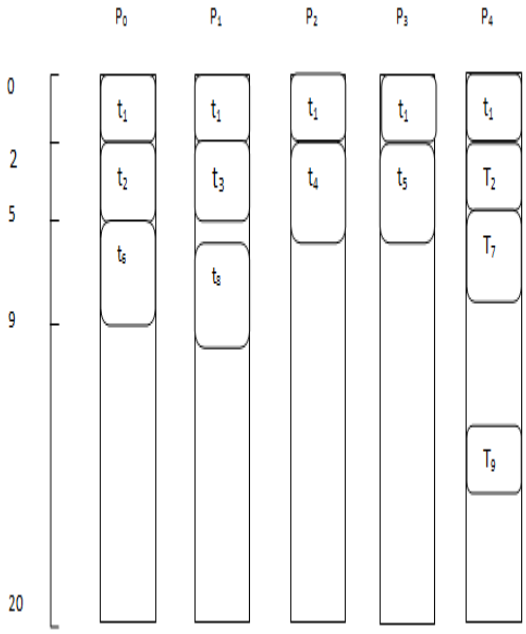

Fig. 2. Number of processors and task allotted to them

VI. RESULTS

Proposed Algorithm is implemented in java. Fig. 3 shows the output of the algorithm. The application has been developed using Jdk1.6. Fig. 3 shows 5 processors and 9 tasks. Processor 1 contains t1, t2, t6 and COM6-9-7 means communication cost from t6 to t9 and t9 is added below t7. Processor 2 contains t1, t3, t8 and COM8-9-7 means communication cost from t8 to t9 and t9 is added below t7.Processor 3 contains t1, t4 and COM4-8-3 means communication cost from t4 to t8 and t8 is added below t3. Processor 4 contains t1 and t5. Processor 5 contains t1, t2, t7and t9. Finally calculated schedule length is 17. Table 1 shows the comparison of proposed algorithm with previous ones. It is observed the proposed algorithm BFSTDB generates the optimal schedule length. The performance of the proposed algorithm is much better as compared to PY (Papadimitriou &Yannakakis), LWB (Lower Bound algorithm) and TDGA (Task Duplication Genetic Algorithm).

VII. CONCLUSION

In this paper, implementation of Breadth First Search Based Task Duplication (BFSTD) is used to solve the task scheduling problem. Proposed scheduling algorithm based on the task duplication pays attention on getting the shortest scheduling length. The performance of new algorithm i.e. BFSTDB is compared with the traditional PY algorithm and genetic algorithm TDGA. In performance analysis it was observed that that new algorithm outperform the existing task duplication algorithms PY (Papadimitriou &Yannakakis) and TDGA (Task Duplication Genetic Algorithm) etc. and gives less schedule length and time complexity of algorithm is also less.

VIII. FUTURE WORK

No doubt, proposed algorithm generates less schedule length but when genetics is applied to any algorithm then it can perform better. Therefore future work may include this algorithm combined with genetic algorithm.
Since proposed algorithm is better in terms of schedule length but its processor requirement is high. Therefore it can be modified such that number of processors requirement is reduced.

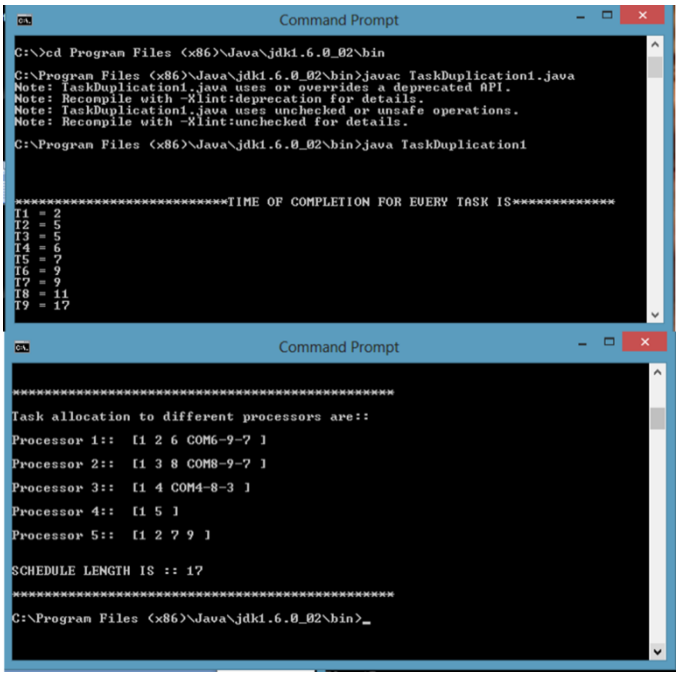

Fig. 3.Results of Algorithm 1 implementation on Fig. 1 Task Graph (DAG)

REFERENCES

[1] Jeffrey D. Ullmann, "Towards an Architecture ndependent Analysis of Parallel Algorithms", *SIAM Journal on Computing*, pp. 322-328, (Apr 1990).
[2] Fatma A. Omara, Mona M. Arafa, "Genetic algorithms for task scheduling problem", Journal on Parallel Distribution and Computation, (6 October 2009).
[3] Yu KwongKwak, Ishfaq Ahmed, "Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors", *Hong Kong Research Grants Council*, pp. 6-39, (July 1998).
[4] Ishfaq Ahmad, Yu-Kwong Kwok, "Analysis, Evaluation, and Comparison of Algorithms for Scheduling Task Graphs on Parallel Processors", 1994.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

[5] Ahmad, I. and Kwok, Yu-K, "Benchmarking and Comparison of the Task Graph Scheduling Algorithms", *IEEE Conference Publications*, pp. 1063-7133, 1998.

[6] .Y. Colin and P. Chretienne, "C.P.M. Scheduling with Small Computation Delays and Task Duplication, Operations Research", pp. 680-684, 1991.

[7] Y.C. Chung and S. Ranka, "Application and Performance Analysis of a Compile-Time Optimization Approach for List Scheduling Algorithms on Distributed-Memory Multiprocessors", *Proceedings of Supercomputing' 92*, pp. 512-521, (Nov. 1992).

[8] B. Kruatrachue and T.G. Lewis, "Grain Size Determination for Parallel Processing", *IEEE Software*, pp. 23-32, (Jan. 1988).

[9] [9] M.A. Palis, J.C. Liou, S. Rajasekaran, S. Shende, S.S.L Wei, "Online scheduling of dynamic trees. *Parallel Processing Lett. 5*", pp. 635-646, 1995.