IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-2 Issue-8 E-ISSN: 2347-2693

Reliability Based Task Allocation Scheme to Enhance the Performance of Distributed Environment

Faizul Navi Khan[1*] and Kapil Govil[2]

[1,2]Department of Computer Applications, Teerthanker Mahaveer University, Moradabad, UP, India
faizulnavi@yahoo.co.in, drkapilgovil@gmail.com

www.ijcseonline.org

Received: 23 Aug 2014 Revised: 04 Sep 14 Accepted: 29 Aug 2014 Published: 05 Sep 2014

Abstract**—** Distributed Environment (DE) aims at achieving higher execution speed than the one obtainable with uniprocessor system by exploiting the collaboration of multiple computing nodes interconnected in some fashion. The idea has been to partition a uniprocessor computing load into multiple units of execution and assigning them to the various processing nodes. The best possible speed up will obviously be obtained if the various partitions of the given computational task can run independently in parallel. The processing nodes of the system must be sharing the computational load of the system so as to be able to provide proper execution characteristics. All the processing nodes must be made busy as much as possible by receiving and executing multiple tasks. In such scenario the number of tasks are lesser than available number of processors in DE, the tasks will be assign to the processor without any concern, but incase the numbers of tasks are greater than the numbers of processors then the task allocation problem will introduce in to real life scenario. Task allocation problem for processing of 'm' tasks to 'n' processors (m>n) in a DE is presented here through a new modified tasks allocation scheme to allocate the task in DE. The allocation scheme, proposed in this paper allocates the tasks to the processor to increase the performance of the DE on the consideration of reliability of the task to the processors.

Keywords**—** Allocation Scheme, Distributed Environment, Performance, Processing Reliability, Task allocation

I. INTRODUCTION

A Distributed Environment is a collection of workstations, personal computers and/or other computing systems. Such systems may be heterogeneous in the sense that the computing nodes may have different speeds and memory capacities or may be having same configuration. A DE accepts tasks from users and executes different modules of these tasks on various nodes of the system. In DE, a problem is divided into one or multiple tasks, each of which solved by one computer. Each processor has its own memory and speed and the communication between any two processors of the system takes place by message passing over the DE. Such environment provide very powerful platform for executing high performance parallel applications. But the performance of the system is dependent on task allocation scheme in the DE.

In order to enhance the performance of DE, workloads are divided into small independent units called tasks. To allocate these tasks are the main concern while maintain the system reliability. In the present research paper, we have taken an example of DE where there are 'n' processors with the same processing speed and 'm' tasks (where m>n). This situation looks like a FIFO (First in First out) ordered queue. When first most tasks are assigned to the processor, remaining tasks have to wait until first task has been completed. As given in Fig. 1.

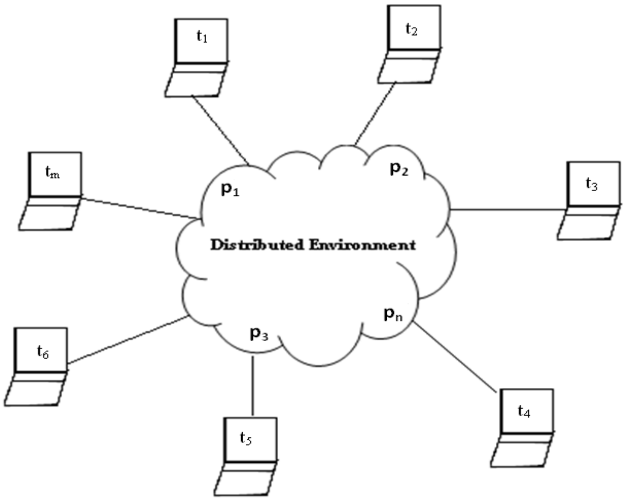

Fig. 1: Task Allocation Problem in Distributed Environment

As per the given scenario first numbers of tasks equal to numbers of processors will be allocate to available processors in the DE and remaining tasks will be on hold until the present allocated task will execute. So to handle the problem of task allocation in a DE required such arrangement where more than single task can assign to a processor so none of the task need to wait in queue and also keep processing reliability in regards that need to be maximized here. Some of the task allocation methods have been reported in the literature, such as Load Balancing [1, 4], Reliability Model [2, 5, 7, 11, 14,

Corresponding Author: Faizul Navi Khan

15], Task Allocation [3, 8, 9, 10, 18, 22], Branch-and-Bound [6], integer programming [12, 16], Mathematical programming Models[13], Distributed Computing System[17, 19, 21] and An approach for allocating tasks [20]. The problem chosen in this paper is related to task allocation in a DE, static in nature and enhance the performance of the distributed environment through optimizing the reliability constraints. In this research paper we have design an allocation scheme to assign all tasks in DE to the available processor in such a way that avoid the situation of overloading by using the proper utilization of processors of the network.

II. Notation

p	Processor
t	Task
n	Number of Processors
m	Number of Tasks
PRM	Processing Reliability Matrix
APRM	Arranged Processing Reliability Matrix

III. OBJECTIVE

The objective of this study is to maximize the processing reliability for a Distributed Environment through optimally assigning the tasks of various processor of the environment, so that the performance of the DE is to be enhanced. The processing of task is means that all of its sub tasks get processed. The type of assignment of tasks to the processor is static. Also insure the processing of the all tasks as tasks are more than the numbers of processors in DE. In this paper performance is measured in term of processing reliability of the task that have to be get processed on the processors of the DE and it have to be optimally processed i.e., reliability to be maximized.

IV. TECHNIQUE

We have demonstrated here the problem where a set P = {p_1, p_2, p_3,………p_n} of 'n' homogenous processors and a set T = {t_1, t_2, t_3,………t_m} of 'm' tasks, where m>n to evaluate the overall optimal processing reliability of a distributed environment. First tasks would be arranged in descending order of their processing reliability. After arranging task in descending order, would assign single task to each processors from the task queue. In next step for each allocation we search again the processor with maximum processing reliability and next task module will assign to that processor, we will repeat these steps until all the task module will assign to the processors in DE.

V. ALGORITHM

```
Step 1:   Start Algorithm
Step 2:        Read the number of task in m
Step 3:        Read the number of processors in n
Step 4:        Store the task and Processing Reliability
               into PRM (,) m x 2 of order
Step 5:   For i = 1 to m
Step 6:        Read the Processing Reliability of
               task $T_i$ in PRM (,)
Step 7:        i = i + 1
Step 8:   End For
Step 9:   Arrange the PRM (,) in descending order
          of their Processing reliability and store
          them in APRM (,) of order m x 2.
Step 10:  For i = 1 to n
Step 11:       Allocate task $T_i$ (from APRM(,))
               to $P_i$
Step 12:  i = i + 1
Step 13:  End For
Step 14:  While all task!=ALLOCATED
Step 15:  For task $t_j$ :
Step 16:       Search for the maximum
               processing reliability by
               multiplying the processing
               reliability $t_j$ and allocate the task
               to the relevant processor.
Step 17:  End While
Step 18:  Multiply all the processing reliability
          processor wise
Step 19: State the result.
Step 20: End Algorithm
```

VI. IMPLEMENTATION

In the present research paper, we have taken Distributed Environment which consist a set P of 4 processors {p_1, p_2, p_3, p_4}, and a set T of 7 tasks {t_1, t_2, t_3, t_4, t_5, t_6, t_7}. It is shown by the Table 1. The processing reliability of each task are known and mentioned in the processing reliability matrix namely PRM () of order 7 x 2.

Table 1: Processing Reliability Matrix

	Task	Reliability
PRM[,] =	t_1	0.999856
	t_2	0.999632
	t_3	0.999631
	t_4	0.999521
	t_5	0.999235
	t_6	0.999631
	t_7	0.999863

To achieve the objective of maximize the processing reliability first arrange task in descending order in regards processing reliability. After the processing reliability of each task in descending order in PRM (,) we get arranged processor reliability matrix namely APRM (,) of order 7 x 2 as mentioned in Table 2.

Table 2: Arranged Processing Reliability Matrix

Task	Reliability
t_7	0.999863
t_6	0.999631

t_5	0.999235
t_4	0.999521
t_3	0.999631
t_2	0.999632
t_1	0.999856

Since there are four processors in the environment so initially that will make us enable to allocate first four tasks (i.e. t_7, t_6, t_5, t_4) to the available processors. First allocation table will be as follows in Table 3:

Table 3: Allocation Table 1

Processor	Tasks	Reliability
p_1	t_7	0.999863
p_2	t_6	0.999631
p_3	t_5	0.999235
p_4	t_4	0.999521

After the first allocation step will consider the next one task (i. e. t_3) from APRM (,), and we will search the processor with maximum reliability by multiplying the reliability of task t_3 with the each processor in the DE. Here processor p_1 has found with the maximum reliability after multiplying the processing reliability of task t_3. So the task t_3 would be allocated to processor p_1. So the assignments are as mentioned in Table 4:

Table 4 : Allocation Table 2

Processor	Tasks	Reliability
p_1	$t_7 * t_3$	0.999494
p_2	t_6	0.999631
p_3	t_5	0.999235
p_4	t_4	0.999521

There will be repetition of step until all tasks get completed. So the final assignment along with the processing is given in the Table 5.

Table 5: Allocation Table 3

Processor	Tasks	Reliability
p_1	$t_7 * t_3$	0.999494
p_2	$t_6 * t_2$	0.999263
p_3	t_5	0.999235
p_4	$t_4 * t_1$	0.999377
Total Reliability		0.997371

VII. CONCLUSION

In this research paper we have taken the scenario, in which the number of the tasks is more than the number of processors in a Distributed Environment. The model mentioned in this paper is based on the consideration of processing reliability of the tasks to various processors. The method is presented in pseudo code and implemented on the several sets of input data to test the performance and effectiveness of the pseudo code. It is the common requirement for any assignment problem that the task have to be processed with maximum reliability. Here, performance is measured in terms of processing reliability of the tasks that has been processed by the processor of the environment and also it is found that all the tasks have been processed optimally. The time complexity of above mentioned algorithm is O(m+n). The optimal result of the example that is considered to test the allocation scheme and it is mentioned in the implementation section of the paper are mentioned in Table 6.

Table 6: Final Allocation Table

Processor	Tasks	Reliability
p_1	$t_7 * t_3$	0.997371
p_2	$t_6 * t_2$	
p_3	t_5	
p_4	$t_4 * t_1$	

Fig. 2 shows the final allocation in DE. That would process the allocated task in optimal way in order to get maximum processing reliability in DE.

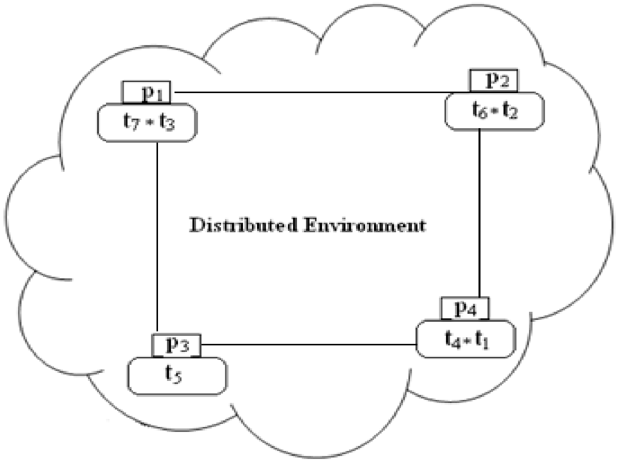

Fig. 2: Task Allocated in Distributed Environment

The graphical representation is also mentioned in Fig 3:

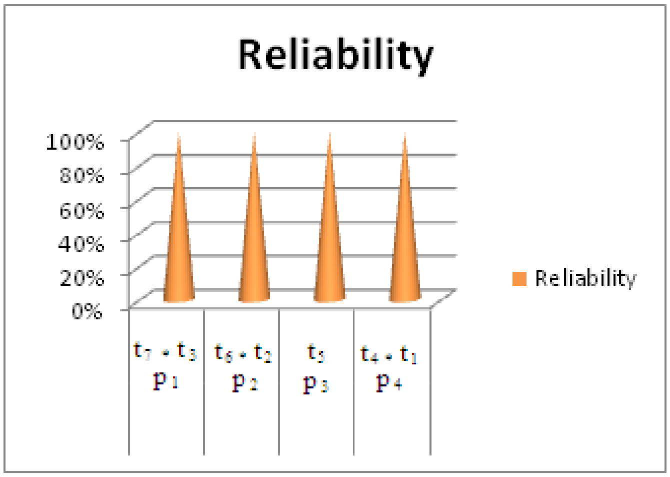

Fig. 2: Graphical Representation of Reliability in Distributed Environment

REFERENCES

[1] Andrey G. Bronevich, Wolfgang Meyer, "Load balancing algorithms based on gradient methods and their analysis

through algebraic graph theory". Journal of Parallel and Distributed Computing, Volume 68, Issue 2, February 2008, Pages 209-220

[2] Anurag Raii, Vikram Kapoor, "Reliable Clustering Model for Enhancing Processors Throughput in Distributed Computing System", International Journal of Computer Applications, Volume 38, Issue 8, Pages 47-50, 2012.

[3] Bo Yang, Huajun Hu, Suchang Guo, "Cost-oriented task allocation and hardware redundancy policies in heterogeneous distributed computing systems considering software reliability". Computers & Industrial Engineering, Volume 56, Issue 4, Pages 1687-1696

[4] Bruce Hendrickson, Karen Devine, "Dynamic load balancing in computational mechanics". Computer Methods in Applied Mechanics and Engineering, Volume 184, Issues 2-4, Pages 485-500, 2010

[5] D. Coit, A. Smith, "Reliability optimization of series-parallel systems using genetic algorithm", IEEE Tran on. Reliability, Volume 45, Issue 2, Pages. 254-266, 1996.

[6] Dorta, C. León, C. Rodríguez, "Performance analysis of Branch-and-Bound skeletons".Mathematical and Computer Modelling, Volume 51, Issues 3-4, Pages 300-308, 2010

[7] Faizul Navi Khan, Kapil Govil, "A Static approach to optimize time cost and reiliability in Distributed Processing Environment". International Journal of Scientific & Engineering Research, Volume 05, Issue 5, Pages 1016-1021, 2014

[8] Faizul Navi Khan, Kapil Govil, "Cost Optimization Technique of Task Allocation in Heterogeneous Distributed Computing System". Int. J. Advanced Networking and Applications, Volume 05, Issue 3, Pages 1687-1696, 2013

[9] Faizul Navi Khan, Kapil Govil, "Static Approach for Efficient Task Allocation in Distributed Environment". International Journal of Computer Applications, Volume 81, Issue 15, Pages 19-22, 2013

[10] Gamal Attiya, Yskandar Hamam, "Task allocation for maximizing reliability of distributed systems: A simulated annealing approach". Journal of Parallel and Distributed Computing, Volume 66, Issue 10, Pages 1259-1266, 2006

[11] Hsieh, Chung-Chi, Hsieh, Yi-Che, "Reliability and cost optimization in distributed computing systems", journal of Computers & Operations Research, Volume 30, Issue 8, Pages 1103-1119, 2003.

[12] I. Kuban Altinel, Necati Aras, Evren Güney, Cem Ersoy. "Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks". Computer Networks, Volume 52, Issue 12, Pages 2419-2431, 2008

[13] Jeffery L. Kennington, Eli V. Olinick, Gheorghe Spiride. "Basic mathematical programming models for capacity allocation in mesh-based survivable networks". Omega, Volume 35, Issue 6, Pages 629-644, 2007

[14] Kapil Govil, "Processing Reliability based a Clever Task Allocation Algorithm to Enhance the Performance of Distributed Computing Environment", Int. J. Advanced Networking and Applications, Volume 03, Issue 01, Pages 1025-1030, 2011

[15] Manoj B.S, Sekhar Archana, Siva Ram Murthy C, "A state-space search approach for optimizing reliability and cost of execution in distributed sensor networks" , Journal of Parallel and Distributed Computing, Volume 69, Issue 1, Pages 12-19, 2009

[16] Maria João Alves, João Clímaco, "A review of interactive methods for multiobjective integer and mixed-integer programming". European Journal of Operational Research, Volume 180, Issue 1, Pages 99-115, 2007

[17] Nirmeen A. Bahnasawy, Fatma Omara, Magdy A. Koutb, Mervat Mosa , "A new algorithm for static task scheduling for heterogeneous distributed computing system", International Journal of Information and Communication Technology Research , Volume 1, Issue 1, Pages 10-19, 2011.

[18] P. K. Yadav, M. P. Singh, Kuldeep Sharma, "An Optimal Task Allocation Model for System Cost Analysis in Heterogeneous Distributed Computing Systems: A Heuristic Approach", International Journal of Computer Applications, Volume 28, Issue 4, Pages 30-37, 2011

[19] Pankaj Saxena, Kapil Govil. "An Optimized Algorithm for Enhancement of Performance of Distributed Computing System", International Journal of Computer Applications, Volume. 64, No. 2, Pages 37-42, 2013

[20] Pankaj Saxena, Dr. Kapil Govil, Neha Agrawal, Saurabh Kumar, and Deep Narayan Mishra. "An approach for allocating tasks in optimized time in a distributed processing environment", International Journal of Innovative Research and Development, Volume 1, Issue 5, Pages. 431-437, 2012

[21] Pradeep Kumar Yadav, M.P. Singh, Kuldeep Sharma, "Task Allocation Model for Reliability and Cost optimization in Distributed Computing System", International Journal of modeling, simulation and scientific computations, Volume 2, Issue 2, Pages. 1-19, 2011.

[22] Zubair khan, ravinder singh, Jahangir alam, "task allocation using fuzzy inference in parallel and distributed system", Journal of Information and Operations Management, Volume 3, Issue 2, Pages-322-326, 2012

AUTHORS PROFILE

Faizul Navi Khan completed his Master in Computer Application from M.D. University Rohtak (Haryana) India in the year 2006, and currently pursuing Ph.D. in Computer Application from Teerthanker Mahaveer University Moradabad, UP, India. He has more than 7+ years of work experience in IT Industry. He is the author of more than 10 research papers published in various journals and conference proceedings.

Kapil Govil received his Ph.D. from Bundelkhand University, Jhansi, Uttar Pradesh, India; He has more than 6+ years of work experience in R&D. He has been contributed more than 50 technical research papers, published in various journals and conference proceedings.