

# Static Approach for Efficient Task Allocation in Distributed Environment

Faizul Navi Khan  
Teerthanker Mahaveer  
University  
Moradabad, UP, INDIA

Kapil Govil  
Teerthanker Mahaveer  
University  
Moradabad, UP, INDIA

## ABSTRACT

Task allocation in a distributed environment is a fundamental problem. In such environment an application runs on a single computer can be accessible on every system / terminal present in the same network. Therefore geographical distance does not matter. If number of tasks is less than or equal to available processors in the network, we can assign these task without any issue. But this allocation becomes complex when numbers of tasks are greater than the numbers of processors. The problem of task allocation for processing of 'm' tasks to 'n' processors ( $m > n$ ) in a distributed networks is addressed here through a new modified tasks allocation technique. The algorithm, presented in this paper allocates the tasks to the processor of different processing capacity to increase the performance of the distributed network. The Algorithm addressed in this paper is based on the consideration of processing time of the task to the processors. Keeping this in view we have suggested a new algorithm to assign all the tasks as per the required availability of processors and their processing capacity so that none of the tasks get remains unexecuted in the distributed environment.

## Keywords

Distributed Environment, Performance, Processing Time, Task allocation

## 1. INTRODUCTION

Distributed environment solves a large problem by making small parts of the problem to many computers to solve and then combining the solutions for the parts into a solution for the problem. Alternatively, each computer may have its own user with individual needs, and the purpose of the distributed environment is to coordinate the use of shared resources or provide communication services to the users. Instead of single large processor being responsible for all aspects of process, there are several separate processor handles these aspects. A distributed environment can be seen as virtual uniprocessor. In the distributed environment the program or tasks are also often developed with the subsets of independent units under distributed environments. In the present research paper, we have taken a domain of distributed environment where there are 'n' processors with the different processing speed and 'm' tasks (where  $m > n$ ). This condition 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 figure 1.

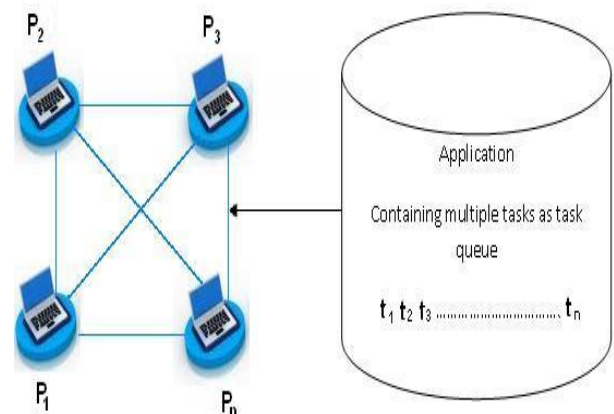


Figure 1: Task Allocation Problem

According to the scenario first number of tasks equal to numbers of processors will be assigned to available processors and rest of tasks have to wait until the present allocated task will execute. To overcome this problem of task allocation in a distributed environment more than single task will assign to a processor in order to get execute minimum response time. Some of the task allocation methods have been reported in the literature, such as Such as processing reliability [8,10], Task Allocation for Minimizing Cost [1,2,4,5,6,7,11,12,13,15, 16,17], Evaluation of gang scheduling performance and cost [9], Decentralized management of bi-modal network resource in a distributed stream processing platform[3] and Static workload distribution of parallel application[14]. In this research paper we have taken the problem of task allocation in distributed computing environment based on execution time. We have developed a new algorithm to get maximum optimization of task allocation in such a way that overall allocation should also be balanced that avoid the situation of overloading by using the proper utilization of processors of the environment.

## 2. NOTATIONS

p	Processor
t	Task
n	Number of Processors
m	Number of Tasks
TTM	Task Time Matrix
PTM	Processing Time Matrix
TPTM	Temporary Processing Time Matrix
STPTM	Sum of Temporary Processing Time Matrix

### 3. OBJECTIVE

The objective of this research paper is to minimize the overall processing time for a distributed environment through optimally assigning the tasks of various processor with different processing capacity of the environment, so that the performance of the distributed environment is to be upgraded. 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 all the modules of task as task modules are more than the numbers of processors of the environment. In this paper performance is measured in term of processing time of the task that have to be get processed on the processors of the environment and it have to be efficiently processed i.e., time to be minimized.

### 4. TECHNIQUE

In order to evaluate the overall optimal processing time of a distributed environment, we have chosen the problem where a set  $P = \{p_1, p_2, p_3, \dots, p_n\}$  of 'n' processors with different configuration and a set  $T = \{t_1, t_2, t_3, \dots, t_m\}$  of 'm' tasks and n-1 imaginary tasks will be added for any number of task to complete the frame of  $n \times n$ , to ensure that each task gets allocated, where  $m > n$ . Processing time are known for all task modules for every processor and will be arrange the processing time for each task for different processor in a Processing Task Time Matrix (PTM) of order  $(n \times m)$ . After arranging the task modules we will break PTM Matrix into another frame called Temporary Processing Time Matrix TPTM  $[n][n]$  and will also initialize the processing load for all processors by zero. Now will compute all possible combination of elements column wise by using TPTM (.). Hence the total number of possible combination will be  $n^3$  and will also calculate the sum of all possible combination and will store in another matrix Sum of Temporary Processing Time Matrix STPTM(.). After compute the sum of all possible combination in the matrix will find minimum value of sum and will compare processing load on within the available processor if task got matched with the correct processor where the processing load is also minimum then assign the task otherwise search for next minimum value of sum and the task will get assign, repeat these steps until all the task module will assign to the processors in distributed environment.

### 5. ALGORITHM

1. Start Algorithm
2. Read the number of task in m
3. Read the number of processor in n
4. Store task and Processing Time into Matrix PTM(.)  $n \times m$  of order
5. While (All task != Assigned)
  - {
  - i. Consider n tasks in sequence and store in TPTM(.) in order of  $n \times m$ .
  - ii. Create all possible combination and calculate the sum for each combination and store values in STPTM().
  - iii. Search the minimum value of sum and draw Combination Time Matrix (CTM)
  - iv. Check the processing load of matching processor for eligible task (with minimum processing time).
  - v. If it is minimum then the task will get assign otherwise will go to step iii and search next minimum value to task get assigned.
  - }
6. State the result
7. End of algorithm

### 6. IMPLEMENTATION

In the present research paper, we have taken distributed environment which consist a set P of 3 processors  $\{p_1, p_2, p_3\}$  with different configuration, and a set T of 6 tasks  $\{t_1, t_2, t_3, t_4, t_5, t_6\}$ . Here numbers of processors are three hence two imaginary tasks are also added to complete the frame  $3 \times 3$  column with 0 processing time ( $t_7$  and  $t_8$ ) to make sure all the original six tasks can be allocated. The processing time of each task varies processor wise and known and mentioned in the processing time matrix namely PTM of order  $3 \times 8$ .

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$
$p_1$	11	12	10	21	29	31	0	0
$p_2$	13	14	11	23	31	33	0	0
$p_3$	10	13	12	20	27	35	0	0

In order to task allocation in efficient manner we take first three tasks for the first frame and will create a temporary matrix TPTM of  $[3][3]$  as follows:

	$t_1$	$t_2$	$t_3$
$p_1$	11	12	10
$p_2$	13	14	11
$p_3$	10	13	12

Here we will make all the possible combinations of elements and find out the smallest sum of possible combinations. For the first frame the combination, we have as follows:

	$t_1$	$t_2$	$t_3$
$p_1$	-	-	10
$p_2$	-	-	11
$p_3$	10	-	-

Here  $t_1$  will get assign to  $p_3$  processor. Assignment table 1 will be as follows for the first frame:

**Table 1: Allocation Table**

Processor	Task	Processing Time
$p_3$	$t_1$	10

By the following the same procedure for the rest frames, we find following allocation table:

**Table 2: Final Allocation Table**

Processor	Task
$p_1$	$t_3 * t_5$
$p_2$	$t_2 * t_6$
$p_3$	$t_1 * t_4$

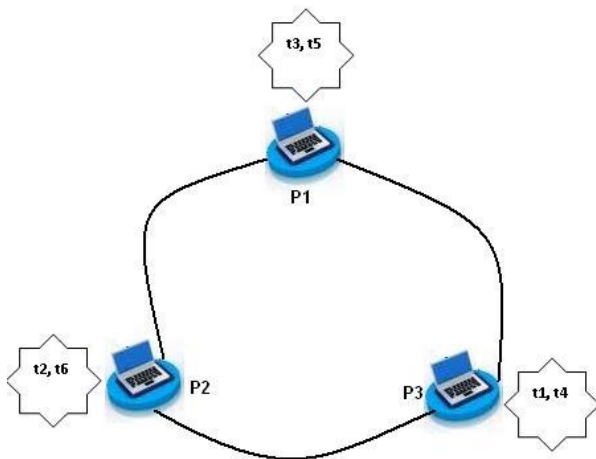
## 7. CONCLUSION

In this research paper we have consider m number of tasks need to assign n number of processor where m is always greater than n. We have tried here to allocate task in best efficient manner in terms of processing time in a distributed environment. The technique stated in pseudo code applied on several sets of input data and that verified the objective of get minimum response time for given tasks for their execution. Here we measured the performance in terms of processing time of the tasks that has been processed by the processor of the network. The outcome of the given example as mentioned below here:

Processor	Task	Processing Time
p <sub>1</sub>	t <sub>3</sub> * t <sub>5</sub>	39
p <sub>2</sub>	t <sub>2</sub> * t <sub>6</sub>	47
p <sub>3</sub>	t <sub>1</sub> * t <sub>4</sub>	30
Total Processing Time		116

**Table 3: overall processing time**

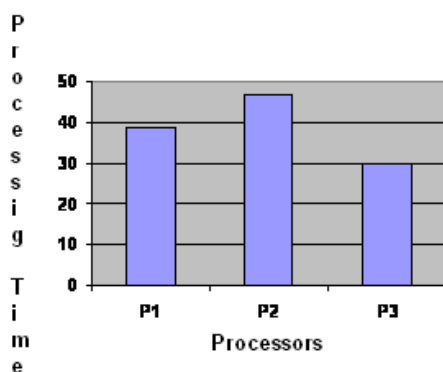
Figure 2 shows the final task allocation as mentioned in Table 3.



**Figure 2: Resultant task allocation**

We have also drawn graphical representation of stated outcome of the given input as shown below:

### Graphical representation of Processing Time



**Figure 3: Graphical representation of processor wise processing time**

## 8. REFERENCES

- [1] Anju Khandelwal, "Optimal Execution Cost of Distributed System through Clustering", International Journal of Engineering Science and Technology, Vol: 3, issue: 3, pp: 2320-2328, 2011.
- [2] Ahmed Younes, Hamed, "Task Allocation for Minimizing Cost of Distributed Computing Systems Using Genetic Algorithms", International Journal of Advanced Research in Computer Science and Software Engineering, vol: 2, issue: 9, pp: 1202-1209, 2012.
- [3] Asaduzzaman Shah, Maheswaran Muthucumaru, "Decentralized management of bi-modal network resources in a distributed stream processing platform", Journal of Parallel and Distributed Computing, vol: 71, issue 6, pp: 774-787, 2011.
- [4] Boeres, Cristina1, Rebello, Vinod E.F, "A versatile cost modeling approach for multicomputer task scheduling", journal of Parallel Computing, vol: 25, issue :1 , pp: 63-86, 1999.
- [5] Bo yang, huajun hu, suchang guo, "Cost-oriented task allocation and hardware redundancy policies in heterogeneous distributed computing systems considering software reliability", Journal Computers and Industrial Engineering, vol: 56, issue: 4, pp: 1687-1696, 2009.
- [6] G.sagar, anil K, sarj E, "Task allocation model for distributed systems", International Journal of Systems Science, vol: 22, issue: 9, pp: 1671-1678, 1991.
- [7] Grzegorz Chmaj, Krzysztof WalkoWiak, Michal Tarnawski, Michal Kucharzak, "Heuristic Algorithms for Optimization of Task Allocation And Result Distribution in Peer-To-Peer Computing Systems", International Journal of Appl. Math and Computer Science, Volume-22, No-3, Page 733-748, 2012
- [8] Hsieh, Chung-Chi, Hsieh, Yi-Che, "Reliability and cost optimization in distributed computing systems", journal of Computers & Operations Research, vol: 30, issue: 8, pp: 1103-1119, 2003.
- [9] Ioannis A Moschakis, Helen D Karatza, "Evaluation of gang scheduling performance and cost in a cloud computing system", Journal of Supercomputing, vol: 59, issue: 2, pp: 975-992, 2012.
- [10] Kapil Govil, "Processing Reliability based a Clever Task Allocation Algorithm to Enhance the Performance of Distributed Computing Environment", Int. J. Advanced Networking and Applications, Vol: 03, Issue: 01, pp: 1025-1030, 2011
- [11] Keren A, Barak A, "opportunity cost algorithms for reduction of I/O and inter process communication overhead in a computing cluster", IEEE transaction on parallel and distributed systems, vol: 14, issue: 1, pp: 39-50, 2003.
- [12] Kołodziej Joanna, Xhafa Fatos, "Modern approaches to modeling user requirements on resource and task allocation in hierarchical computational grids", International Journal of Applied Mathematics and Computer Science, vol: 21, issue: 2, pp: 243-257, 2007.
- [13] Manisha Sharma, Harendra Kumar, Deepak Garg "An Optimal Task Allocation Model through Clusering with Inter-Processor Distances in Heterogeneous Distributed

- Computing Systems". International Journal of Soft Computing and Engineering (IJSCE), Volume-2, Issue-1, March 2012
- [14] Marwa Shouman, Gamal Attiya, Ibrahim Z. Morsi, "Static Workload Distribution of Parallel Applications in Heterogeneous Distributed Computing Systems with Memory and Communication Capacity Constraints", International Journal of Computer Applications, vol:34, issue:6, pp:18-24, 2011.
- [15] Mostapha Zbakh, Said El Hajji, "Task allocation problem as a non cooperative game", Journal of Theoretical and Applied Information Technology, vol: 16, issue: 2, pp: 110-115, 2010.
- [16] Najjar Faiza, Slimani, Yahya, "Extension of the one-shot semijoin strategy to minimize data transmission cost in distributed query processing", Journal of Information Sciences, vol: 114, issue 1-4, pp: 1-21, 1999.
- [17] 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, vol: 2, issue:2, pp. 1-19, 2011.