

An Intelligent Algorithm for proper Allocation of Tasks in Distributed Network

Kapil Govil
Smart Online Academy
<http://smartonlineacademy.in>

Shilpi Sharma
Smart Online Academy
<http://smartonlineacademy.in>

ABSTRACT

A network is a set of devices, which are connected by media links. These devices may be called as nodes. A node can be computer, printer, or any other device, which may send and / or receive data generated by other node on the network. In distributed network, the task is divided among multiple modules and there are multiple processors so it looks like a virtual uniprocessor. The allocation of tasks in distributed network has an important role to the evaluation of the performance of these networks. This paper describes an efficient task allocation method for distributed network. It is based on the cost for the execution of the tasks to the various processors and also the communication amongst the tasks. In the almost all real world problem, the number of tasks is more than the number of processors of the distributed network.

Keywords

Distributed Network, Allocation, Task, Processor, Communication

1. INTRODUCTION

In the distributed network services are provided by the network reside at multiple sites. Instead of single large machine being responsible for all aspects of process, each separate processor handles subset. In the distributed network the program or tasks are also often developed with the subsets of independent units under various environments. It has drawn tremendous attention in developing cost-effective and reliable applications to meet the desired requirement.

The main research problem for such networks is the allocation problem, in which either processing time or cost is to be minimized or reliability is to be maximized of the networks. These problems may be categorized as static [4, 8, 11, 12] and dynamic [1, 14, 17, 19] in nature. Some of the other related methods have been reported in the literature, such as, Integer programming [15, 18, 20, 21], Reliability optimization [2, 7, 16] and Modeling [3, 5, 9, 10, 13]. The series parallel redundancy-allocation problem has been studied with different approaches, such as, dynamic programming [1, 14, 19] and Integer programming [6, 15].

2. OBJECTIVE

The objective of the present research paper is to enhance the performance of the distributed network by using the proper utilization of its processors. A set of tasks have to be processed by the processors of the network, while the number of tasks are more than the number of processors of the network. Performance is the measure in term of cost of a task that have to process on the processors of the network and these have to be optimally processed i. e. cost to be minimized.

3. TECHNIQUE

To evaluate the optimal cost for each task through optimal

allocation, initially concentration should be on those tasks that have the highest probability of their communication. By finding out the highest communication among tasks, on fusing the corresponding elements of tasks. Addition of all the elements of kth row and lth row reduces the matrix for each task in to a square matrix. Now the problem remains to determine the optimal cost through the allocation strategy by considering either task processing based on cost for all modules to individual processor(s) for each task. For allocation purpose a modified version of row and column assignment method proposed by Kumar et al [12] is employed which allocates all the modules of a task to a processor optimally. The function for obtaining the overall processing cost [Pcost] is as follows:

$$Pcost = \left[\sum_{j=1}^m \left\{ \sum_{i=1}^n PC_{ij} X_{ij} \right\} \right] \quad (1)$$

Where,

$$X_{ij} = \begin{cases} 1, & \text{if } i^{\text{th}} \text{ task is assigned to } j^{\text{th}} \text{ processor} \\ 0, & \text{otherwise} \end{cases}$$

4. ALGORITHM

Step 1: Start Algorithm

Step 2: Input the matrix for Processing Cost (PC) i.e, PC (). Select cost data corresponding to each task as per requirement

Step 3: Input the Communication Matrix CM ()

Step 4: Sort the upper diagonal elements of communication matrix in ascending order without changing their position in matrix

Step 5: Fuse the task in PC () matrix according to the sorting order of CM () to make the effectiveness matrix square

Step 6: Find the smallest element of each row and replace it with zero

Step 7: Find the smallest element of each column and replace it with zero

Step 8: Count the zero(s) in each row

Step 9: Mark the row(s), which have single zero

Step 10: Mark the column, which have single zero

Step 11: Go to the row(s), which have more than single zero. Now select any one zero and cross the leading zero(s), which are in same row and column

- Step 12: Mark the assignments
- Step 13: Count the total assignment
- Step 14: If total number of assignment < order of matrix
 - Go to Step 15
 - Else
 - Go to Step 22
- Step 15: Mark the rows for which assignment have not been made
- Step 16: Mark column that have zeros in marked rows
- Step 17: Mark rows that have assignment in marked column
- Step 18: Repeat Step 16 & Step 17 until chain of marking ends
- Step 19: Draw the minimum number of lines through

- unmarked rows and marked columns to cover all zeros
- Step 20: Select the smallest element of the uncovered elements and replace it by zero. Also add this element to positions at which lines intersect to each other only
- Step 21: Go to Step 9
- Step 22: State processing cost
- Step 23: End Algorithm

5. IMPLEMENTATION

Here, it is considered that the processing of the tasks based on the cost constraints. Further it is also noted that the number of tasks are more than the number of processors in the distributed network. On considering an example consisting of a set $T = \{t_1, t_2, t_3, t_4, t_5\}$ of 5 tasks and 3 processors are available in the distributed network to process the tasks that are represented by the set $P = \{p_1, p_2, p_3\}$. This problem is shown in Fig 1.

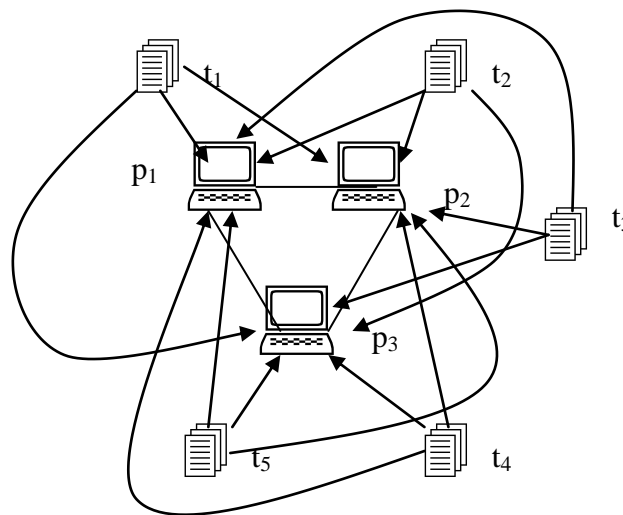


Fig 1: Unallocated task – processor combination

The processing cost (c) of each module of every task on various processors are known and mentioned in the Processing Cost (PC) matrix, namely, PC():

	p ₁	p ₂	p ₃
t ₁	130	140	150
t ₂	110	90	130
t ₃	90	80	140
t ₄	80	150	160
t ₅	170	120	150

The communication period amongst each task has also been considered and it is mentioned in the following Communication Matrix (CM), namely, CM():

The matrix CM () is as:

	t ₁	t ₂	t ₃	t ₄	t ₅
t ₁	0	2	3	9	7
t ₂		0	4	8	6
t ₃			0	1	5
t ₄				0	2
t ₅					0

There are five tasks, so that on the basis of highest communication, the tasks t₁ & t₄ and t₂ & t₅ are fused together to reduce the matrix into square matrix. The resulting matrix is as:

	p ₁	p ₂	p ₃
t ₁ *t ₄	210	290	310
t ₂ *t ₅	280	210	280
t ₃	90	80	140

The results of the allocations based on cost for the tasks are obtained after implementing the row & column assignment process as suggested by Kumar et al [12], as mentioned in the

Table 1.

Table 1: Time based Allocation for tasks

Processors	Tasks	Processing Cost	Overall Processing Cost
p ₁	t ₁ *t ₄	210	560
p ₂	t ₂ *t ₅	210	
p ₃	t ₃	140	

p ₁	t ₁ *t ₄	210	560
p ₂	t ₂ *t ₅	210	
p ₃	t ₃	140	

The overall task allocation is shown in the Fig 2.

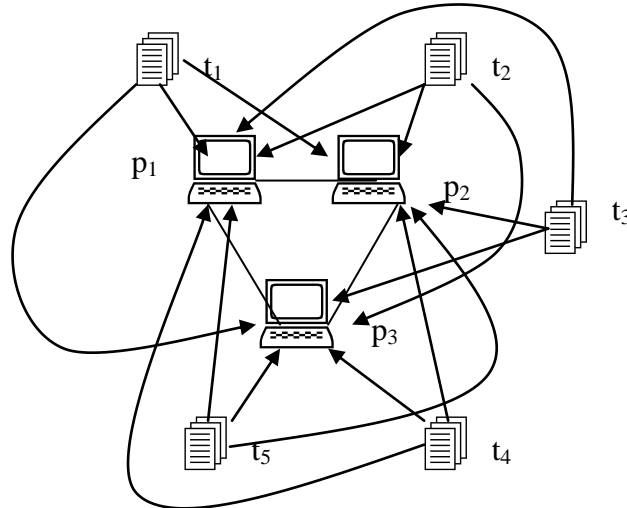


Fig 2: Allocated Tasks

The overall task allocation graph along with processing cost is shown in Fig 3.

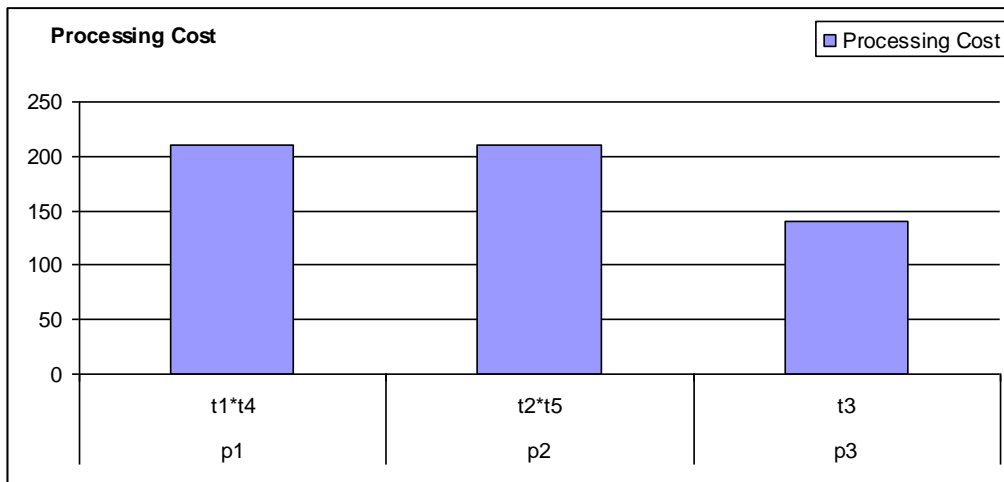


Fig 3: Allocated Tasks

6. CONCLUSION

Present paper chooses the problem in which the number of the tasks is more than the number of processors of the distributed network. The model addressed in this paper is based on the consideration of processing cost of the task to the various processors. The communication period amongst the tasks is also used. The method is presented in algorithmic form and implemented on the several sets of input data to test the performance and effectiveness of the algorithm. As it is the common requirement for any assignment that the tasks have to be processed with minimum cost, so this paper allocates the tasks with an efficient manner.

It is known that, the analysis of an algorithm is mainly focuses on time complexity. Time complexity is a function of input size 'n'. It is referred to as the amount of time required by an

algorithm to run to completion. The time complexity of the above mentioned algorithm is $O(m^2n^2)$. By taking several input examples, the above algorithm returns results which are mentioned in Table 2.

Table 2: Time Complexity Calculation Table

No. of processors (n)	No. of tasks (m)	Optimal Results
3	4	144
3	5	225
3	6	324
3	7	441
3	8	576
4	5	400
4	6	576
4	7	784
4	8	1024
4	9	1296
5	6	900
5	7	1225
5	8	1600
5	9	2025
5	10	2500

The graphical representation of the above results are shown by Fig 4, Fig 5 and Fig 6,

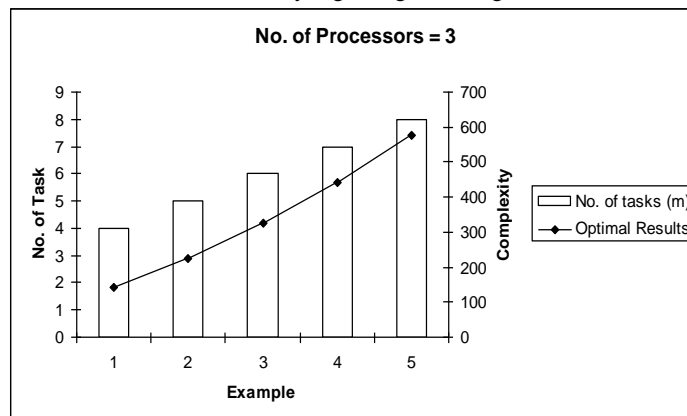


Fig 4: Complexity Chart for 3 processors

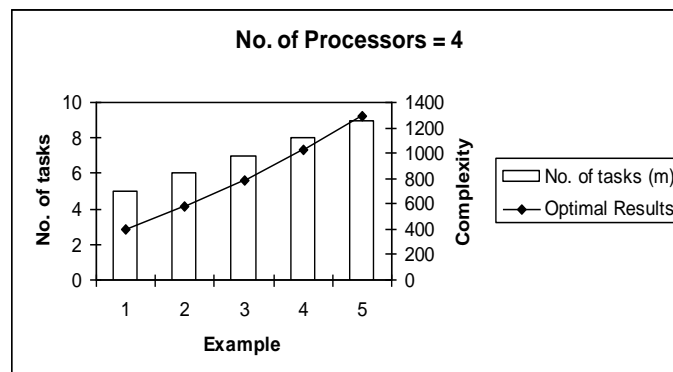


Fig 5: Complexity Chart for 4 processors

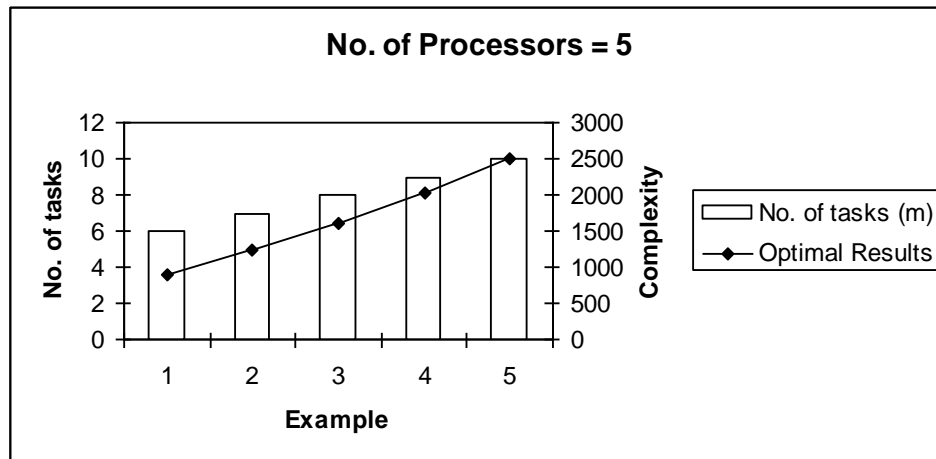


Fig 6: Complexity Chart for 5 processors

7. REFERENCES

- [1] A. Farinelli, L. Iocchi, D. Nardi and V. A. Ziparo. 2005. Task Assignment with dynamic perception and constrained tasks in a Multi-Robot System. Proc. of International Conference on Robotics and Automation (ICRA'05).
- [2] A. Y. Hamed. 2012. Task Allocation for Maximizing Reliability of Distributed Computing Systems Using Genetic Algorithms. International Journal of Computer Networks and Wireless Communications. 2 (5). 560 – 569.
- [3] Bassel Arafeh, Khalid Day and Abderezak Touzene. 2008. A multilevel partitioning approach for efficient tasks allocation in heterogeneous distributed systems. Journal of Systems Architecture: The EUROMICRO Journal. 54 (5). 530 – 548.
- [4] Bo Yang, Huajun Hu, Suchang Guo. 2009. Cost-oriented task allocation and hardware redundancy policies in heterogeneous distributed computing systems considering software reliability. Computers & Industrial Engineering, 56 (4), 1687 – 1696.
- [5] Dorta, C. Leon, and C. Rodriguez. 2010. Performance analysis of Branch-and-Bound skeletons. Mathematical and Computer Modeling, 51 (3 – 4), 300 – 308.
- [6] Ensink, Brian, Stanley, Joel and Adve, Vikram. 2003. Program Control Language: a programming language for adaptive distributed applications. Elsevier Inc., 63 (12), 1082 – 1104.
- [7] Faizul Navi Khan and Kapil Govil. 2014. A tricky task scheduling technique to optimize time cost and reliability in mobile computing environment. International Journal of Research in Engineering and Technology. 3 (5). 823 – 829.
- [8] Faizul Navi Khan and Kapil Govil. 2014. A Static approach to optimize time cost and reliability in Distributed Processing Environment. International Journal of Scientific & Engineering Research. 5 (5). 1016 – 1021.
- [9] Fitzgerald, Kent, Latifi, Shahram and Srimani, Pradip K. 2002. Reliability Modeling and Assessment of the Star-Graph Networks. IEEE Transactions on Reliability, 51, 49 – 57.
- [10] Jihyun Park, Byoungju Choi and Seungyeun Jang. 2020. Dynamic Analysis Method for Concurrency Bugs in Multi-process/Multi-thread Environments. International Journal of Parallel Programming. 48. 1032 – 1060.
- [11] Kumar Avanish, 2001. An Algorithm for Optimal Index to Tasks Allocation Based on Reliability and cost. Proceedings of International Conference on Mathematical Modeling, Roorkee. 150 – 155.
- [12] Kumar, V. Singh, M. P. and Yadav, P. K. 1995. A Fast Algorithm for Allocating Tasks in Distributed Processing System. Proceedings of the 30th Annual Convention of CSI, Hyderabad. 347 – 358.
- [13] Manisha Sharma, Harendra Kumar and Deepak Garg. 2012. An Optimal Task Allocation Model through Clustering with Inter – Processor Distances in Heterogeneous Distributed Computing Systems. International Journal of Soft Computing and Engineering. 2 (1). 50 – 55.
- [14] Monika Choudhary and Sateesh Kumar Peddoju. 2012. A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment. International Journal of Engineering Research and Applications (IJERA). 2 (3). 2564 – 2568.
- [15] Muhammad K. Dhodhi, Imtiaz Ahmad, Anwar Yatama and Ishfaq Ahmad. 2002. An Integrated Technique for Task Matching and Scheduling onto Distributed Heterogeneous Computing Systems. Elsevier Inc., 62(9), 1338 – 1361.
- [16] Ormon, Stephen W., Cassady, C. Richard and Greenwood, Allen G. 2002. Reliability Prediction model to Support Conceptual Design, IEEE Transactions on Reliability, 51, 151 – 157.
- [17] Ravindran, Binoy, Devarasetty, Ravi K. and Shirazi, Behrooz. 2002. Adaptive Resource Management Algorithms for Periodic Tasks in Dynamic Real-Time Distributed Systems. Elsevier Inc., 62 (10), 1527 – 1547.
- [18] Shu Wanneng, Wang Jiangqing, Min – Min Chromosome. 2009. Genetic Algorithm for Load Balancing in Grid Computing. International journal of Distributed sensor network, 5(1), 62 – 63.
- [19] V. Pilloni, P. Navaratnam, S. Vural, L. Atzori and R. Tafazolli. 2014. TAN: A Distributed Algorithm for Dynamic Task Assignment in WSNs. Sensors Journal, IEEE. 14 (4). 1266 – 1279.
- [20] Wang Fan, Chen Ben, Miao Zhaowei. 2008. A Survey on Reviewer Assignment Problem. Book on new frontiers in applied artificial intelligence. 5027, 718 – 727.
- [21] Zhengao, Dandamudi Sivarama P. 2009. An Adaptive Space-Sharing Policy for Heterogeneous Parallel Systems. Book on high performance computing and networking, 2110, 353 – 362.