A Solution to Task Allocation Problem Statically by Considering the Cost Factor in a Distributed Computing System

Pankaj Saxena[1], Dr.Kapil govil[2]

[1, 2] *Teerthanker Mahaveer University, Moradabad (U.P)*

Email: pankbly@gmail.com, drkapilgovil@gmail.com

Abstract: The problem of the task allocation in Distributed Computing System (DCS) is to need to allocate a number of tasks to different processors for execution. The purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users. We consider the problem of finding an optimal task allocation in distributed computing system with the goal of minimizing the system cost. In the present paper we are taking n to denote the number of processors and m for denoting the number of different tasks where m>n. A distributed network consists of multiple autonomous computers that communicate through a communication media. In a static task allocation, the information regarding the tasks and processor attributes is assumed to be known in advance, before the execution of the tasks.so it gives better execution environment. Execution cost can be termed as the amount of value of resource used. In this paper an algorithm is presented to solve the problem of static task allocation in DCS, i.e. given a set of communicating tasks to be processed on a set of processors, to which processor should each task be allocated to get the more reliable results in lesser cost.

Keywords- Processing Cost, Distributed Computing System (DCS), Task, Processor.

1. INTRODUCTION

In distributed networking, a problem is divided into many tasks, each of which is solved by one computer. A distributed network consists of multiple autonomous computers that communicate with each other in order to achieve a common goal. Distributed networking [1, 2] also refers to the use of distributed systems to solve computational problems. In distributed networking, each processor has its own private memory (distributed memory) Information is exchanged by passing messages between the processors. In the present paper we are taking n to denote the number of processors and m for denoting the number of different tasks where m>n. The tasks are allocated in FIFO (first in first out) order. The first tasks is assigning to the processor which is free and any other task will wait for allocating on that processor until the processor will not be free or will not complete task. In this paper the task is allocated statically to the processor. To improve the performance of distributed network [4, 7, 9] it is required to minimize the overall cost [10, 12] and it can be possible by applying a better task allocation [8, 5] schedule in a situation where tasks are more than the number of processors. Some other related methods which exist in literature are integer programming, load balancing, distributed computing, distributed networking etc.In present research paper we are designing an algorithm that, there any particular processor should not be overburdened means the tasks should be approximately balanced.

2. OBJECTIVE

The objective of the present research paper is to enhance the performance of distributed network in terms of processing cost [8, 11]. In distributed network, there is a network of many processors where each processor accomplishes the task to get the optimal result more quickly as well as more efficiently. In the present research paper, the type of task allocation [3, 6] to the processors is in static in nature. Here we have taken an example of distributed network where the number of processors is lesser in comparison to the number of tasks. In this research paper there is a set of tasks which are to be get processed by the processors in optimized processing cost.

Notations

n : Number of processors
m : Number of tasks
p : Processor
t : Task
PCM : Processor Cost Matrix
MPCM : Modified Processor Cost Matrix
APCM : Arranged Processor Cost Matrix

Technique

In order to evaluate the overall optimal processing cost of a distributed network we have chosen the problem where a set P= $\{p_1, p_2, p_3 \ldots p_n\}$ of 'n' processors and a set T= $\{t_1, t_2, t_3 \ldots t_m\}$ of 'm' tasks, where m>n. The processing cost of each and every task to each and every processor is known and it is mentioned in the processing cost matrix namely PCM (,) of order n*m. On making the addition of each row of PCM (,) we find MPCM (,) and after arranging MPCM (,) it in ascending order of their sum of row we find Arranged Processing Cost Matrix namely APCM (,).

3. ALGORITHM

Step 1: Read the number of tasks in m
Step 2: Read the number of processor in n
Step 3: Read the Processing Cost Matrix PCM (,) of order n*m
Step 4: Compute sum of each row of PCM (,) and store it into Modified Processor Cost Matrix MPCM (,)
Step 5: Arrange the MPCM (,) in ascending order of their row sum and store it into Arranged Processor Cost Matrix APCM (,)
Step 6: While (all tasks! = allocated)
Step 7: {
Select and allocate particular task from APCM (,) processor which has the minimal processing cost.
}
Step 8: processor Compute the wise overall processing cost.
Step 9: Display the result

4. IMPLEMENTATION

In the present research paper we have chosen a distributed networking consisting a set P of 4 processors $\{p_1,p_2,p_3,p_4\}$ and a set T of 10 tasks $\{t_1,t_2,t_3,\ldots..t_{10}\}$. It is shown by Fig. 1.

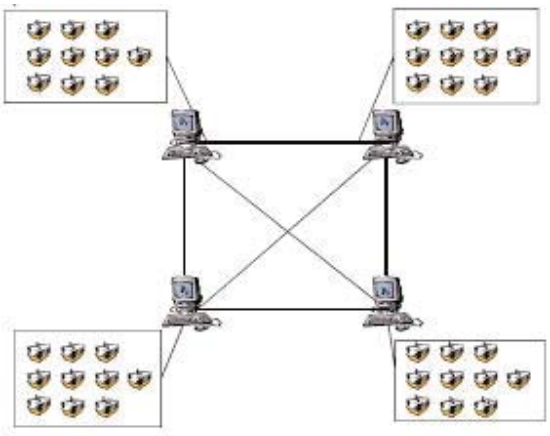

Fig. 1 Task allocation problem in distributed network

The processing cost (c) of each task on various processors are known and mentioned in the processing cost matrix namely PCM (,).

		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
	p_1	2000	2300	2800	2100	1400	2500	1800	1400	2900	2100
PCM(,) =	p_2	1500	2500	3000	1600	1800	2300	1600	1200	3000	1100
	p_3	1700	2400	2900	1200	2800	2600	1700	1800	2300	1400
	p_4	2200	2600	2400	1500	2700	1900	1800	2400	2400	1200

On making the addition of each row of PCM (,) we find Modified Processing Cost Matrix namely MPCM (,).

		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	Sum
	p_1	2000	2300	2800	2100	1400	2500	1800	1400	2900	2100	21300
MPCM(,) =	p_2	1500	2500	3000	1600	1800	2300	1600	1200	3000	1100	19600
	p_3	1700	2400	2900	1200	2800	2600	1700	1800	2300	1400	20800
	p_4	2200	2600	2400	1500	2700	1900	1800	2400	2400	1200	21100

On arranging MPCM (,) in ascending order of their row sum, we find Arranged Processor Cost Matrix APCM (,) of order n*m.

		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	Sum
	p_2	1500	2500	3000	1600	1800	2300	1600	1200	3000	1100	19600
APCM(,) =	p_3	1700	2400	2900	1200	2800	2600	1700	1800	2300	1400	20800
	p_4	2200	2600	2400	1500	2700	1900	1800	2400	2400	1200	21100
	p_1	2000	2300	2800	2100	1400	2500	1800	1400	2900	2100	21300

Now we select first 4 tasks (t_1, t_2, t_3, t_4) from APCM (,), as we have 4 processors (p_1, p_2, p_3, p_4). On selecting first 4 tasks row wise which have the minimal processing cost at any specified processor, we find the allocations which are mentioned in Table 1.

TABLE I
TASK ALLOCATION

Processors	Allocated Task	Processing Cost
p_1	t_2	2500
p_2	t_1	1500
p_3	t_4	1200
p_4	t_3	2400

Again, we select next four tasks (t_5, t_6, t_7, t_8) from APCM (,).Again on selecting 4 tasks which have the minimal processing cost at any specified processor, we find the allocations which are mentioned in Table 2.

TABLE II
TASK ALLOCATION

Processors	Allocated tasks	Processing Cost
p_1	t_5	1400
p_2	t_8	1200
p_3	t_7	1700
p_4	t_6	1900

Here we have only two tasks (t_9, t_{10}) more. On repeating the similar concept we select tasks which have the minimal processing cost at specified processor, we find the allocation which are mentioned in Table 3.

Processor	Allocated Task	Processing Cost
p_1	...	...
p_2	t_{10}	1100
p_3	t_9	2300
p_4	...	...

5. CONCLUSION

In the present paper the problem is discussed to allocate the different tasks to different processors. The number of tasks is more than the number of processors. The tasks are allocated to the processors with the objective to gain the result by optimum utilization of processors in terms of processing cost for each task. The concept in the present paper is presented in algorithmic form and it is tested for many inputs to check the accuracy of result. The implementation presented in this paper has 4 processors and 10 tasks. The problem is to allocate the task in order to evaluate optimal processing cost in distributed network.

We can show the overall task allocation cost wise by the fig 1

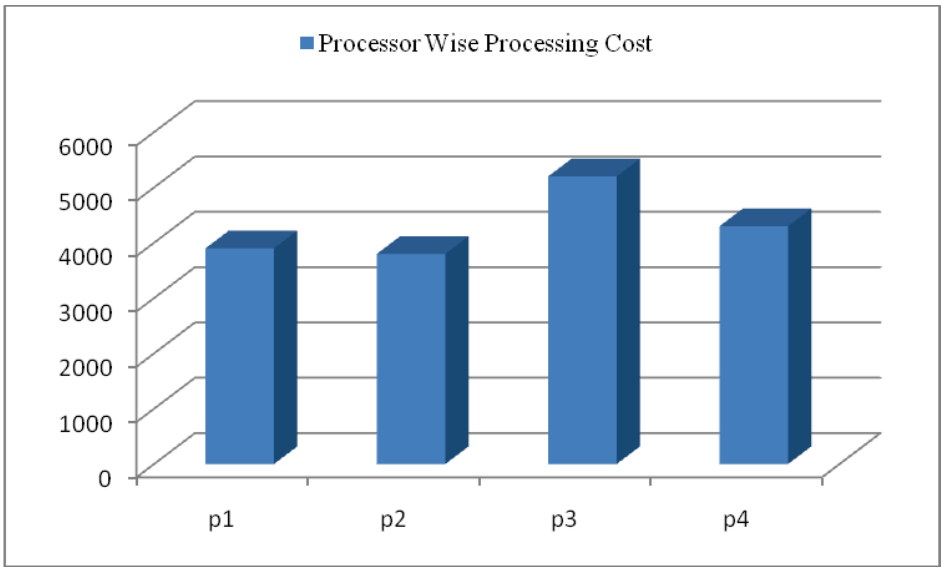

Fig 2: Processor Wise Cost

References

[1]. Hsieh, Chung-Chi, Hsieh and Yi-Che, Reliability and cost optimization in distributed computing systems, journal of Computers & Operations Research, vol. 30, pp. 1103-1119, 2003.

[2]. Ioannis A Moschakis and Helen D Karatza, Evaluation of gang scheduling performance and cost in a cloud computing system, Journal of Supercomputing, vol.59, pp. 975-992, 2012.

[3]. Keren A and 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, pp.: 39-50, 2003.

[4]. Kołodziej Joanna and 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, pp. 243–257, 2007.

[5]. Marwa Shouman,Gamal Attiya and 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, pp.18-24, 2011.

[6]. H.Kumar et al, A task allocation model for distributed data network, Journal of Mathematical Sciences, vol.1, pp.379-392, 2006.

[7]. Manoj B.S, Sekhar Archana and 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, vol. 69, pp. 12-19, 2009.

[8]. Manisha Sharma, Harendra Kumar and Deepak Garg, An Optimal Task Allocation Model through Clustering with Inter-Processor Distances in Heterogeneous Distributed Computing Systems, International Journal of Soft Computing and Engineering, vol. 2, pp.50-55, 2012.

[9]. Mostapha zbakh and said el hajji, Task allocation problem as a non cooperative game, Journal of Theoretical and Applied Information Technology, vol. 16, pp. 110-115, 2010.

[10]. Najjar Faïza, Slimani and Yahya, Extension of the one-shot semijoin strategy to minimize data transmission cost in distributed query processing, journal of Information Sciences, vol. 114, pp. 1-21, 1999.

[11]. Malith Jayasinghea, Zahir Taria, Panlop Zeephongsekulb, Albert Y. Zomayac , "On Choosing a Task Assignment Policy for a Distributed Server System" , Journal of Parallel and Distributed Computing, Volume 59, Issue 2, Pages 204-228 ,1999.

[12]. Deo Prakash Vidyarthia, Anil Kumar Tripathib, "Maximizing reliability of distributed computing system with task allocation using simple genetic algorithm" , Journal of Systems Architecture, Volume 47, Issue 6, Pages 549-554 ,2001.