INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

An Approach for Allocating Tasks in Optimized Time in A Distributed Processing Environment

Pankaj Saxena
&
Dr.Kapil Govil
Teerthanker Mahaveer University, Moradabad, (U.P).

Neha Agrawal
Saurabh Kumar
&
Deep Narayan Mishra
Future Institute of Engineering and Technology, Bareilly (U.P)

Abstract: *In distributed network in which a task is divided into many fractions and each of which is to be get processed, the task allocation problem can be explained in terms of number of tasks and number of processors available. A distributed network consists of multiple autonomous computers that communicate through a communication media. According to the task allocation problem if the number of task are more then the number of processors and every processor process the task in a particular time period for processing any particular task then we have to allocate each task to the single processor in such a way that the task should be completed in a optimal possible time and also there should not be overloading of task to any one processor. The number of processors and number of tasks are static in nature.*

If the number of processors is denoted by P and the number of tasks is denoted by T then as in general all real world problem T>P. It is required to perform all the tasks by allocating the task in such a way that the total processing time should be minimized.

Keywords- *Distributed Network; Task; Processor; Task allocation.*

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 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 it is required to minimize the overall time and it can be possible by applying a better task allocation schedule in a situation where tasks are more then the number 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.

Objective

The objective of the present research paper is to enhance the performance of distributed network in terms of processing time. 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 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 time.

Notations

n : Number of processors

m : Number of tasks

p : Processor

t : Task

PTM : Processor Time Matrix

MPTM : Modified Processor Time Matrix

APTM : Arranged Processor Time Matrix

Technique

In order to evaluate the overall optimal processing time of a distributed network we have chosen the problem where a set P= {p1, p2, p3…pn} of 'n' processors and a set T= {t1, t2, t3…tm} of 'm' tasks, where m>n. The processing time of each and every task to each and every processor is known and it is mentioned in the processing time matrix namely PTM (,) of order n*m. On making the addition of each row of PTM (,) we find MPTM (,) and after arranging MPTM (,) it in ascending order of their sum of row we find Arranged Processing Time Matrix namely APTM (,)

Algorithm

Step 1: Read the number of tasks in m

Step 2: Read the number of processor in n

Step 3: Read the Processing Time Matrix PTM (,) of order n*m

Step 4: Compute sum of each row of PTM (,) and store it into Modified Processor Time Matrix MPTM (,)

Step 5: Arrange the MPTM (,) in ascending order of their row sum and store it into Arranged Processor Time Matrix APTM (,)

Step 6: While (all tasks! = allocated)

Step 7: {Select and allocate a particular task from APTM (,) processor which has the minimal processing time}

Step 8: Compute the processor wise overall processing time

Step 9: Display the result

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.

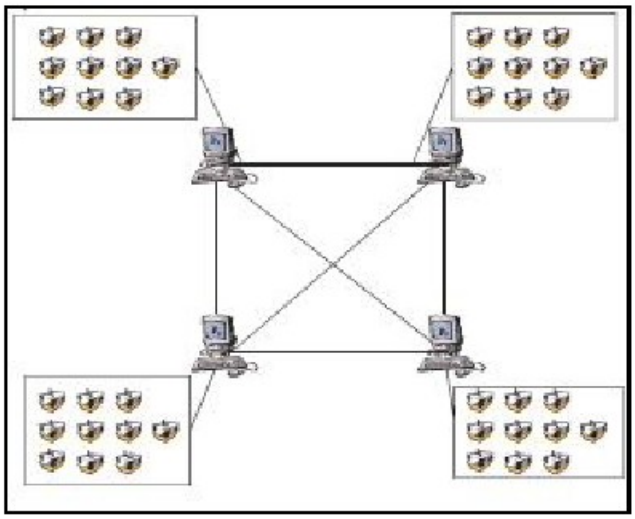

Figure: 1 Task allocation problem in distributed network

The processing time (t) of each task on various processors are known and mentioned in the processing time matrix namely PTM (,).

		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
	p_1	4	3	4	6	17	15	16	14	41	42
PTM(,) =	p_2	10	2	7	6	24	23	25	19	17	32
	p_3	15	20	12	30	40	26	35	34	28	25
	p_4	21	8	2	9	7	29	38	36	23	11

On making the addition of each row of PTM (,) we find Modified Processing Time Matrix namely MPTM (,).

		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	Sum
	p_1	7	2	3	4	15	16	17	19	45	44	172
MPTM(,) =	p_2	11	1	8	5	21	23	25	20	18	32	164
	p_3	12	22	9	30	40	26	37	34	28	27	265
	p_4	21	6	2	9	7	39	38	36	23	9	190

On arranging MPTM (,) in ascending order of their row sum, we find Arranged Processor Time Matrix APTM (,) of order n*m.

APTM(,) =		t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	
	p_2	11	1	8	5	21	23	25	20	18	32	164
	p_1	7	2	3	4	15	16	17	19	45	44	172
	p_4	21	6	2	9	7	39	38	36	23	9	190
	p_3	12	22	9	30	40	26	37	34	28	27	265

Now we select first 4 tasks (t_1, t_2, t_3, t_4) from APTM (,), 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 time at any specified processor, we find the allocations which are mentioned in Table 1.

Processor	Allocated task	Processing time(t)
p_2	t_2	1
p_1	t_3	3
p_4	t_4	9
p_3	t_1	12

Table 1: Task Allocation

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

Processor	Allocated task	Processing time(t)
p_2	t_8	20
p_1	t_5	15
p_4	t_7	38
p_3	t_6	26

Table 2: Task Allocation

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

Processor	Allocated task	Processing time(t)
p_2	t_9	18
p_1	t_{10}	44
p_4	--	--
p_3	--	--

Table 3: Task Allocation

Now the overall allocations on the various processors are mentioned in Table 4.

Processor	Task allocation	Processing time
p_2	t2*t8*t9	360
p_1	t3*t5*t10	1980
p_4	t4*t7	342
p_3	t1*t6	312
Overall processing time :		2994

Table 4: Task Allocation

Conclusion

In the present paper the problem is discussed to allocate the different tasks to different processors. The number of tasks is more then 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 time 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 time in distributed network.

Reference

1. Kim Jong-Kook, Shivle Sameer, Siegel Howard Jay, Maciejewski Anthony A, BraunTracy D, Schneider Myron, "Dynamically mapping tasks with priorities and multiple deadlines in a heterogeneous environment", Journal of parallel and distributed Computing, vol. 67, issue 2, pp. 154-169, 2007.
2. Shu Wanneng, Wang Jiangqing, Min-Min Chromosome, "Genetic Algorithm for Load Balancing in Grid Computing", International journal of Distributed sensor network, vol. 5, issue1, pp. 62-63,2009.
3. Wang Fan, Chen Ben, Miao Zhaowei, "A Survey on Reviewer Assignment Problem", Book on new frontiers in applied artificialintelligence, vol. 5027/2008, pp. 718-727, 2008.
4. Zhengao, Dandamudi Sivarama P, "An Adaptive Space-Sharing Policy for Heterogeneous Parallel Systems", Book on highperformance computing and networking, vol. 2110/2009, pp. 353-362, 2009.
5. Hua Xia Gui, Hui Li, "Study on logistic service supply chain task allocation based on MAS", IEEE International Conference, vol. 2,pp. 1621 – 1625, 2008.
6. Hsu Tsan-sheng, Lopez Dian Rae, "Bounds and algorithms for a practical task allocation model (extended abstract)", Book on Algorithms and Computation, vol. 1178/1996, pp. 397-406, 2006.
7. Jr Paulo R. Ferreira, Boffo Felipe S. Bazzan Ana L., "Using Swarm- GAP for Distributed Task Allocation in Complex Scenarios", Book on Massively Multi-Agent Technology, vol. 5043/2008, pp. 107-121, 2008.
8. Altine I. Kuban, Aras necati, Guney evren, Ersoy cem, "Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks", The InternationalJournal of Computer and Telecommunications Networking, vol.52, issue 12, pp. 2419-2431, 2008.
9. Herlihy Maurice, Luchangco victor, "Distributed computing and the multicore revolution", Journal of ACM SIGACT News, vol.39, issue 1, pp. 62-72, 2008.