Cost Optimization of Distributed Computing System with Dynamic Re-Allocation

Faizul Navi Khan
Teerthanker Mahaveer University
Moradabad, UP, INDIA

Kapil Govil
ITM University
Gwaliar, MP, INDIA

Prof. R.K. Dwivedi
Principal CCSIT, Teerthanker Mahaveer University,
Moradabad, UP, INDIA

ABSTRACT

A Distributed Computing System (DCS) is a combination of application and system programs that exchanges data across a number of independent terminals connected by a communication network. Cost optimization in DCS can be achieve by optimize the performance of DCS. In task allocation two types of approaches are available and these are dynamic and static. Dynamic approach of task allocation is much better as compare to static, since it makes the best use of available computational resources in DCS. Task allocation problem can be describe as ‘m’ number tasks are required to execute on ‘n’ number of processors where number tasks (m) is always greater than number of processors (n) (m>n). This research offers a cost optimization algorithm with dynamic re-allocation of tasks to allocate the ‘m’ number of tasks on ‘n’ number of processors in DCS and their execution completes in k number of phases. Proposed algorithm is tested in MATLAB environment and it is noticed that obtained results are better as compared to past algorithms. Cost optimization dynamic model present in this research is helpful in performance optimization of DCS and also reduce the cost of task allocation in DCS.

Keywords

Distributed Network, Dynamic Allocation, Performance, Residing cost, Reallocation cost

1. INTRODUCTION

Distributed Computing System (DCS) utilizes a network of many terminals in which each accomplish a partial operation of a task. Terminals are connected through communication channel either wireless or wired. DCS provides the facility of utilizing remote computing resources or data not reside in local terminals and also useful to increase the performance by providing facilities for distributed processing. DCS provide a higher performance, better reliability and better results over centralized network systems. In a DCS, the tasks of an application program must be allocated to processors to utilize the computational power and resources of the system in optimize manner. However, every time it is not possible that task willexecute to processors for which the total cost of execution is minimized.

Dynamic task allocation has several merits over to static allocation in DCS. Dynamic allocation is a best way to use maximum utilization of computing resources in DCS becausere-allocation of task modules dynamically are recommends during program execution, so as to take advantage of changes in the local reference patterns of the program. Hence Dynamic approach of task allocation are more common because they are more efficient and flexible.

This research consider a problem of task allocation for allocating ‘m’ tasks of an application to ‘n’ processors (m>n) in such a way that total cost of execution can be minimized in distributed computing system. A dynamic task allocation method is used here to solve task allocation problem and it allocates the tasks to optimize overall task allocation cost. Some of the other related task allocation methods have been reported in literature such asTask Assignment [1, 10, 19], task scheduling [2, 3], Distributed Processing Environment [4], Task Allocation [5, 7, 9, 11, 13], data communication in Mobile Computing [6], Dynamic Task Allocation [8], Dynamic Tasks Scheduling [12, 16, 20, 22], Dynamic Optimization Algorithm [14], dynamic programming [15], Dynamic Reassignment [17], Dynamic Load Balancing [18] and Dynamic Task Assignment [21].This research solve here a task allocation problem dynamically in DCS by allocating available processors in such a way that overall load should be balanced so system overloading can be avoided by using the proper utilization of processors.

2. OBJECTIVE

The objective of this research is to find out the optimal cost for a task allocation problem in Distributed Computing System (DCS) or develop a task allocation model that can minimize the overall system cost with the dynamic re-allocation approach. This research offeres a mathematical model that allocates the tasks dynamically as tasks executes in various phases. During the particular task execution rest of other task are residing in the particular phase. Execution cost for each phase [EC], inter task communication cost [ITCC], residence cost [RC] of each task on different processors and relocation cost [REC] for each task are considered to design a dynamic tasks allocation model. To achieve cost optimization in DCS allocation method finds an allocationwith minimum allocation cost.

3. TECHNIQUE

This research considers a distributed computing system consisting of a set $T = \{t_1, t_2, t_3, t_4, \ldots t_m\}$ of m tasks to be allocated on a set $P = \{p_1, p_2, p_3, \ldots p_n\}$ of n processors divided into k phases with criteria tasks m are more than the number of processors n (m>n). Execution cost for phase wise of each processor is given in the form of Execution Cost Matrix ECM(,,) of order k x m x n. The Residing Cost for residing the unexecuted tasks on the processor is mentioned in Residing Cost Matrix RCM(,,) of order k x m x n. The Inter Task Communication Costbetween executing and non-executing tasks are also considered and is mentioned in the Inter Task Communication Cost Matrix ITCCM(,) of order m x k and during the processing a task is re-allocate from one processor to another processor then it also obtained some cost i.e. reallocation cost and it is given in the Reallocation Cost

Matrix RECM(,) of order m x k. To calculateERCM(,) for each phase sum up the values of ECM(,,) and RCM(,,). Compute the average of each row of ERCM(,) and arrange the values in increasing order in AVG_ROW_{asc}(). Now it selectsfirst n number tasks from AVG_ROW_{asc}() and store them in ERCM_I(,) and remaining n number of tasks in ERCM_II by partitioning ERCM(,) into two sub problems. Follow the same process for next n or less than n and solve them using assignment method. Evaluate the Execution Cost, Communication Cost and Reallocation Cost. Follow the same process for all phasesand at the end calculate the value of Execution Cost, Communication Cost and Reallocation Cost to obtain the phase wise total execution cost. Calculate the sum of optimal cost of each phase to evaluate the overall optimal cost of distributed computing system.

4. ALGORITHM

Step 1: Start Algo
Step 2: Read the number of tasks in m
Step 3: Read the number of processors in n
Step 4: Read the number of phases in k
Step 5: Read the Execution Cost Matrix ECM(,,) of order k xm x n
Step 6: Read the Residing Cost Matrix RCM(,,) of order k x mx n
Step 7: Read the Inter Task Communication Cost Matrix ITCCM(,) of order m x k
Step 8: Read the Reallocation Cost Matrix RECM(,,) of orderm x k
Step 9: For I = 1 to m
i. K = I:
ii. Add the values for ECM(,,) and RCM(,,) and store theresults in ERCM(,,)
iii. Store the average of each row of ERCM(,,) and store it in AVG_ROW()
iv. sort AVG_ROW() in ascending and store the results inAVG_Row_{asc}()
v. While (All tasks of AVG_Row_{asc}() !=SELECTED)
{
a. Make partition of ERCM(,,) for n tasks, store it in ERCM_I(,) and ERCM_II(,)
b. Apply assignment method on ERCM_I(,) and ERCM_II(,)
}
v. Compute Execution Cost (EC),Inter Task Communication Cost (ITCC) and ReallocationCost (RC)
vi. Total Cost = EC + ITCC + RC
I = I + 1
vii. Optimal Cost = (Total Cost)
Step 10: End for
Step 11: State Results
Step 11: End Algo

5. IMPLEMENTATION

This research considers a distributed computing system which is made up of four tasks {t_1, t_2, t_3, t_4} to be allocated on two processors {p_1, p_2} in five phases. The phase wise execution cost of individual processor is known in the form of Execution Cost Matrix ECM(,,) of order k x m x n where k is the number of phases, m is the number of tasks and n is the number of processors. Residing costs for the remaining tasks, except for the executing task, on each processor is also known and mentioned in Residing Cost Matrix (,,) or order k x m x n.Inter Task Communication Cost between the executing task and all other task if they are on different processors also taken into the consideration and mentioned in Inter Task Communication Cost Matrix ITCCM(,) or order m x k. During the execution an allocated task is shifted from one processor to another processor during the next phase then some cost is incurred in reassignment process at the end of each phase andit is known as reallocation cost. Reallocation cost for the given example is also known and it is mentioned in Reallocation Cost Matrix RECM(,) of order m x k.

Table 1: Phase wise Execution Cost Matrix (ECM)

	Phase	Task	Execution Cost	
			p_1	p_2
ECM (,,) =	1	t_1	6	5
		t_2	-	-
		t_3	-	-
		t_4	-	-
	2	t_1	-	-
		t_2	8	7
		t_3	-	-
		t_4	-	-
	3	t_1	-	-
		t_2	-	-
		t_3	4	6
		t_4	-	-
	4	t_1	-	-
		t_2	-	-
		t_3	-	-
		t_4	5	3
	5	t_1	9	7
		t_2	-	-
		t_3	-	-
		t_4	-	-

Table 2: Phase wise Residing Cost Matrix (RCM)

	Phase	Task	Residing Cost	
			p_1	p_2
RCM (,,) =	1	t_1	-	-
		t_2	4	3
		t_3	3	4
		t_4	4	5
	2	t_1	2	3
		t_2	-	-
		t_3	4	5
		t_4	2	3
	3	t_1	4	2

		t_2	3	4
		t_3	-	-
		t_4	3	5
	4	t_1	5	3
		t_2	3	2
		t_3	2	3
		t_4	-	-
	5	t_1	-	-
		t_2	5	3
		t_3	2	3
		t_4	3	4

Table 3: Inter Task Communication Cost Matrix (ITCCM)

	Tasks ↓	Phases				
		1	2	3	4	5
ITCC(,)=	t_1	-	3	4	3	-
	t_2	2	-	3	4	2
	t_3	5	3	-	4	3
	t_4	2	3	5	-	0

Table 4: Reallocation Cost Matrix (RECM)

	Tasks ↓	Phases				
		1	2	3	4	5
RECM(,)=	t_1	2	4	3	3	-
	t_2	3	3	2	4	-
	t_3	2	3	5	3	-
	t_4	2	3	4	2	-

Task t_1 will execute in phase 1 as mentioned in ECM (Table 1) while remaining tasks i.e. t_2, t_3 and t_4 will be treat as a residing tasks. Allocation algorithm will compute the sum of ECM(,,) and RCM(,,) for phase 1 and will form another matrix named as ERCM:

		p_1	p_2
ERCM (,)=	t_1	6	5
	t_2	4	3
	t_3	3	4
	t_4	4	5

Average of each row of ERCM(,) is calculated and store in an linear array named as AVG_ROW(). On arranging the values of AVG_ROW in ascending order a new linear array AVG_ROW$_{asc}$() is formed.

$$\text{AVG\_ROW}() = \left\{ \begin{matrix} t_1 & t_2 & t_3 & t_4 \\ 5.5 & 3.5 & 3.5 & 4.5 \end{matrix} \right\}$$

$$\text{AVG\_ROW}_{asc}() = \left\{ \begin{matrix} t_2 & t_3 & t_4 & t_1 \\ 3.5 & 3.5 & 4.5 & 5.5 \end{matrix} \right\}$$

In order to find optimal allocation, allocation technique will split ERCM(,) by selecting first two tasks from AVG_ROW$_{asc}$() and store the values in ERCM_I(,) and last two tasks into ERCM_II respectively:

		p_1	p_2
ERCM_I(,)=	t_2	4	3
	t_3	3	4

		p_1	p_2
ERCM_II(,)=	t_4	4	5
	t_1	6	5

On applying assignment method on ERCM_I and ERCM_II to allocate the tasks, allocation and their costs is obtained and present in Table 5.

Table 5: Initial Dynamic Allocation Table

Phase	Executing Task	Processor	Assigned Task	Execution Cost (EC)	Communication Cost (CC)	Reallocation Cost (RC)	Phase-wise Total Cost (EC + CC + RC)
1	t_1	p_1	$t_3 * t_4$	7	2	0	17
		p_2	$t_2 * t_1$	8			

By following the same algorithm on the remaining phases final allocation is obtained as present in Table 6 for given example.

Table 6: Phase wise Dynamic Allocation Table

Phase	Executing Task	Processor	Assigned Task	Execution Cost (EC)	Communication Cost (CC)	Reallocation Cost (RC)	Phase-wise Total Cost (EC + CC + RC)
1	t_1	p_1	$t_3 * t_4$	7	2	0	17
		p_2	$t_2 * t_1$	8			
2	t_2	p_1	$t_3 * t_1$	6	3	7	26

		p_2	$t_2 * t_4$	10			
3	t_3	p_1	$t_4 * t_2$	6	5	5	23
		p_2	$t_3 * t_1$	8			
4	t_4	p_1	$t_4 * t_3$	7	4	7	23
		p_2	$t_1 * t_2$	5			
5	t_1	p_1	$t_2 * t_3$	7	2	0	20
		p_2	$t_1 * t_4$	11			
Total execution cost							109

6. CONCLUSION

This research designed a task allocation model with dynamic reallocation technique for execution of tasks in Distributed Computing System (DCS) and provides the optimal solution in order to get optimized costs for task allocation. This allocation model considered the several factors of dynamic environment i.e. execution cost, residing cost, reallocation cost, inter task communication cost and most important execution phases. In dynamic model a tasks completes its execution in various phase so presented dynamic allocation model provide optimal solution phase wise. The presented model is tested in MATLAB platform by creating distributed environment as mentioned in Fig. 1. Optimal cost is calculated for each phase and every task. Communication cost between executing and non-executing task, reallocation cost of task are also added to evaluate final optimal cost of each phase. Phase wise results are generated in MLATLAB for presented algorithm and algorithm [17], results are compared for both algorithms, on comparing phase wise execution cost and total execution cost, it is found presented model shows the better results as mentioned in Table 7.

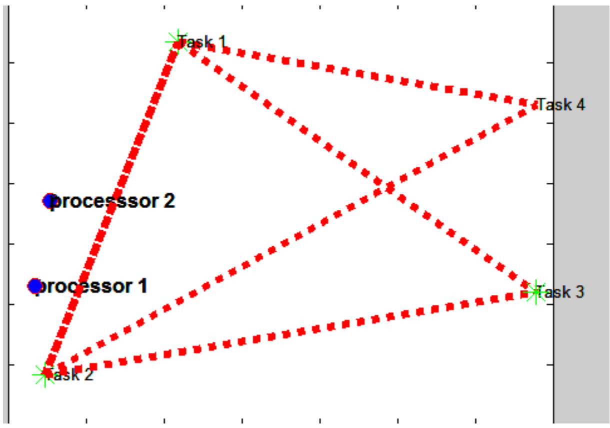

Fig.1: Distributed environment in MATLAB

Table 7: Algorithm results derived in MATLAB environment and compare with algorithm [17]

Phase	Executing Task	Processor	Assigned Task	Phase wise cost for presented algorithm	Phase wise cost for algorithm [17]
1	t_1	p_1	$t_3 * t_4$	18	18
		p_2	$t_2 * t_1$		
2	t_2	p_1	$t_3 * t_1$	24	24
		p_2	$t_2 * t_4$		
3	t_3	p_1	$t_4 * t_2$	33	25
		p_2	$t_3 * t_1$		
4	t_4	p_1	$t_4 * t_3$	28	30
		p_2	$t_1 * t_2$		
5	t_1	p_1	$t_2 * t_3$	15	23
		p_2	$t_1 * t_4$		
Overall Optimal Cost				118	120

Graphical representation of comparison as mentioned in Table 7 is also drawn and it is mentioned in Fig. 2.

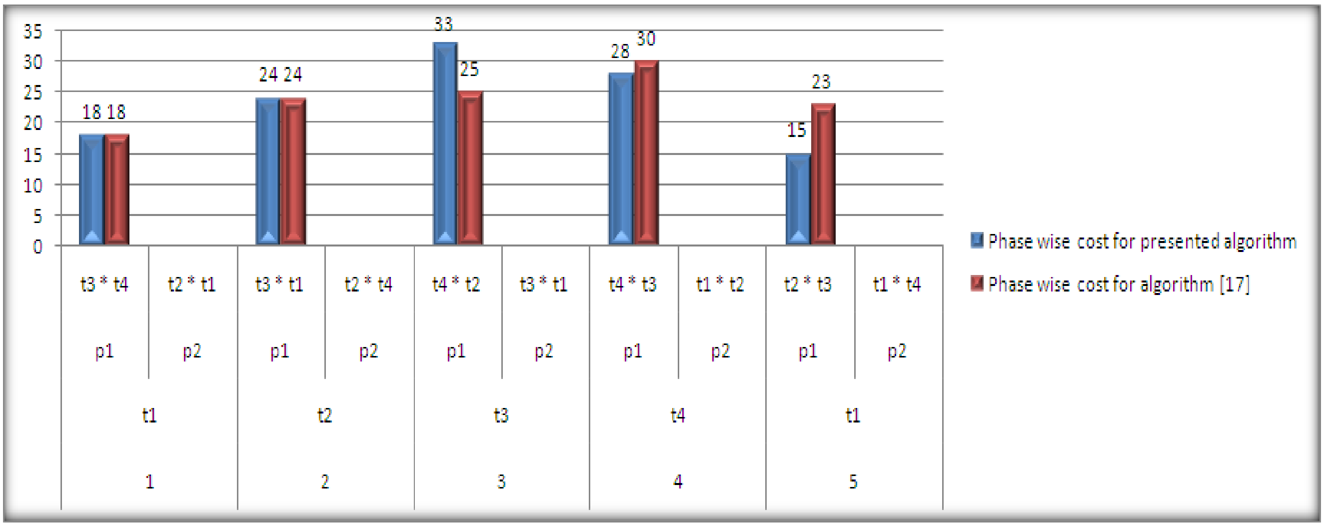

Fig. 2: Showing phase wise comparative results of presented algorithm with algorithm [17]

Overall results also evaluated and compared with algorithm [17] as mentioned in Fig. 3.

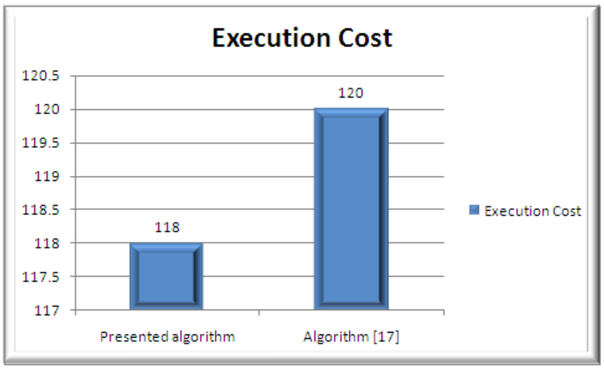

Fig.3: Execution cost for present algorithm and algorithm [17]

7. REFERENCES

[1] A.Farinelli, L. Iocchi, D. Nardi, V. A. Ziparo.2005. Task Assignment with dynamic perception and constrained tasks in a Multi-Robot System, Proc. of Intern. Conf. on Robotics and Automation (ICRA'05)

[2] Faizul Navi Khan, KapilGovil. 2014. A TRICKY TASK SCHEDULING TECHNIQUE TO OPTIMIZE TIME COST AND RELIABILITY IN MOBILE COMPUTING ENVIRONMENT, International Journal of Research in Engineering and Technology, Vol. 3 Issue 5, 823-829

[3] Faizul Navi Khan, KapilGovil. 2014. AN EFFICIENT TASK SCHEDULING ALGORITHM TO OPTIMIZE RELIABILITY IN MOBILE COMPUTING, International Journal of Advances in Engineering & Technology, Vol. 7 Issue 2, 635-641

[4] Faizul Navi Khan, KapilGovil. 2014. A Static approach to optimize time cost and reliability in Distributed Processing Environment. International Journal of Scientific & Engineering Research, Vol. 05, Issue 5, 1016-1021

[5] Faizul Navi Khan, KapilGovil. 2013. Cost Optimization Technique of Task Allocation in Heterogeneous Distributed Computing System, Int. J. Advanced Networking and Applications, Vol. 5 Issue 3, 1913-1916

[6] Faizul Navi Khan, Kapil Govil. 2014. Cluster based optimization routing strategy for data communication in Mobile Computing, International Journal of Computer Applications, Volume 99, Issue 2, 19-24

[7] Faizul Navi Khan, Kapil Govil. 2013. Distributed Task Allocation Scheme for Performance Improvement in Mobile Computing Network, International Journal of Trends in Computer Science, Vol. 2 Issue 3. 809-817

[8] Faizul Navi Khan, Kapil Govil, AlokAgarwal. 2014 Performance enhancement of distributed network system by Phase-wise dynamic task allocation, 2014, International Conference on Parallel, Distributed and Grid Computing (PGDC 2014), IEEE Proceedings, ISBN. 978-1-4799-7681-2

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

[10] Harendra Kumar, M. P. Singh, P. K. Yadav. 2013. Optimal Tasks Assignment for Multiple Heterogeneous Processors with Dynamic Re-assignment, International Journal of Computers & Technology, Vol. 4, No. 2, 528-535

[11] Kapil Govil. 2011. A Smart Algorithm for Dynamic Task Allocation for Distributed Processing Environment, International Journal of Computer Applications, Vol. 28, No. 2, 13-19

[12] M.P, Singh, P.K.Yadav, H. Kumar, B.Agarwal. 2012. Dynamic Tasks Scheduling Model for Performance Evaluation of a Distributed Computing System through Artificial Neural Network, Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) (Advances in Intelligent and Soft Computing: Published by Springer) Vol.130, 321-331

[13] Manisha Sharma, Harendra Kumar, 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, Vol. 2 No.1, 50-55

[14] Monika Choudhary, Sateesh Kumar Peddoju. 2012. A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment, International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 3, 2564-2568

[15] N. Beaumont. 2009. Using dynamic programming to determine an optimal strategy in a contract bridge tournament, Journal of the Operational Research Society, Vol 61, Issue 5, 732-739

[16] P Visalakshi, S N Sivanandam. 2009. Dynamic Task Scheduling with Load Balancing using Hybrid Particle Swarm Optimization, Int. J. Open Problems Compt. Math., Vol 2, No. 3, 475-488

[17] Pradeep Kumar Yadav, M. P. Singh and Harendra Kumar. 2008. Scheduling Algorithm: Tasks Scheduling Algorithm for Multiple Processors with Dynamic Reassignment, Journal of Computer Systems, Networks, and Communications, Vol 2008, doi:10.1155/2008/578180, 1-9

[18] SagarDhakal, Majeed M. Hayat, Jorge E. Pezoa, Cundong Yang, David A. Bader. 2007. Dynamic Load Balancing in Distributed Systems in the Presence of Delays:A Regeneration-Theory Approach, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, Vol. 18, No. 4, 485-497

[19] ShenChenglin, Zhang Xinxin. 2009. Dynamic Mechanisms of Task-assignment for Virtual Enterprises Based on Multi-agent Theory, Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA'09), 525-528

[20] SunitaBansal, Bhavik Kothari, ChittaranjanHota. 2011. Dynamic Task-Scheduling in Grid Computing using Prioritized Round Robin Algorithm, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2,472-477

[21] V. Pilloni, P. Navaratnam, S.Vural, L. Atzori, R. Tafazolli. 2014. TAN: A Distributed Algorithm for Dynamic Task Assignment in WSNs, Sensors Journal, IEEE, Vol. 14, Issue 4, 1266 - 1279

[22] XiangzhenKonga, Chuang Lina, YixinJianga, Wei Yana, Xiaowen Chub. 2011. Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction, Journal of Network and Computer Applications, Vol. 34, Issue 4, 1068–1077