

Optimal Tasks Assignment for Multiple Heterogeneous Processors with Dynamic Re-assignment

Harendra Kumar* & M. P. Singh

Department of Mathematics and Statistics
Gurkula Kangari University, Hardwar-249404, Uttarakhand (INDIA)
balyan.kumar@gmail.com*
mpsinghgkv@gmail.com

Pradeep Kumar Yadav

Central Building Research Institute
Roorkee-247667, Uttarakhand (INDIA)
prd_yadav@rediffmail.com

Abstract:

Distributed Computing System [DCS] has attracted several researchers by posing several challenging problems. In this paper we have developed a mathematical model for allocating "M" tasks of distributed program to "N" multiple processors ($M > N$) that minimizes the total cost of the program. Relocating the tasks from one processor to another at certain points during the course of execution of the program that contributes to the total cost of the running program has been taken into account. Most of the researchers have considered the cost for relocating the task from one processor to another processor at the end of the phase as a constant. But in real life situations the reallocating cost of the tasks may vary processor to processor this is due to the execution efficiency of the processors. Phase-wise execution cost [EC], inter task communication cost [ITCT], residence cost [RC] of each task on different processors and relocation cost [REC] for each task have been considered while preparing a dynamic tasks allocation model.

Keywords: Distributed Computing System, Phase-wise Execution, Inter-Tasks Communication Cost, Execution Cost, Relocation Cost, Residence Cost

* Corresponding Author

1. Introduction:

Although computer speed has been increased by several orders of magnitude in recent decades the demand for computing capacity increase at an even faster pace. DCS provide faster computation by facilitating parallel execution of tasks of a job. DCS is used to describe a system whenever there are several computers interconnected in some fashion so that a program or procedure runs on the system with multiple processors. In a heterogeneous computing system, the tasks of an application program must be assigned to the heterogeneous processors to utilize the computational capabilities and resources of the system efficiently. However, it is a difficult problem to compute an assignment of tasks to processors for which the total cost of running the program is minimized. Static and dynamic algorithms can be extremely efficient in homogeneous systems. If all processors are identical then the choice "to which node should we assign the next task?" falls on the processor with the fewest outstanding jobs [3]. However in heterogeneous distributed environments, where not all nodes have the same parameters, selecting the optimal target processor requires more elaborate algorithms.

Stone [2] has suggested an efficient optimal algorithm for the problem of assigning tasks to two processors by making use of the well-known network flow algorithm in the related two-terminal network graphs. If the structure of a distributed program is constrained in certain ways, it is possible to find the optimal assignment over a system made up of any number of processors in polynomial time. When the structure is constrained to be a tree, the shortest tree algorithm developed by Bokhari [4] yields the optimal assignment. Towsley [8] generalized the results of Bokhari to the case of series-parallel structures. A further generalized program structure is a partial K-tree in which trees are partial I-tree and series-parallel structures are partial 2-tree [6] assuming distributed system as fully interconnected, i.e., there exists a communication link between any two machines. However, the general n-processors problem ($N > 3$) in a fully interconnected system is known to be NP-complete [7, 10]. C.H. Lee [1] and S.Y. Cho [11] have suggested an optimal algorithm for the general structure problem in a linear array network with any number of processors.

In a general case each task may have multiple computational characteristics due to the dynamic control behavior of the task. Such tightly and medium-coupled portions of code task will be more difficult to break up and assign to different processors. When the characteristics of the computation change, a new assignment must be computed. By 'characteristics of the computation' we mean the ratios of the times that the program spends in different parts of the program. Dynamic algorithms are employed more frequently because they offer a better efficiency, having the flexibility of tasks migration. We use phase as the unit of the time slot for gathering more detailed information of program activity. During this period, the executing task may communicate with any number of the remaining task. A task may not be moved from one processor to another during a phase; it may be moved only between them. A task may complete its execution in more than two phases. Bokhari [5] analyzed the problem of dynamic assignment in a two-processor system, which permits relocation of tasks from one processor to the other at certain points during the execution of the program. This relocation

incurs a predefined relocation cost that contributes to the total cost of running the program and code-conversion overheads. Kumar et al. [12] developed a model for allocating m tasks to n processors divided it into any number of phases. In this model the effect of migration cost for reassigning each task from one processor to other at the end of phase have been not considered. Yadav et al. [13] have developed a mathematical model for arbitrary number of phases and processors with random program structure considering execution, residence, inter-task communication and relocation costs. Recently in 2012 [15] developed a dynamic task allocation model based on artificial neural network [ANN] scheduling approach to arrange the tasks to the processors. Relocating the tasks from one processor to another at certain points during the course of execution of the program that contributes to the total cost of the running program has been taken into account. In the above paper authors have considered that the cost for relocating the task from one processor to another processor at the end of the phase is constant which is not true in real life situation.

The rest of the paper is organized as follows. Notations and assumptions are defined in section 2. In section 3, tasks allocation problem has been discussed. The proposed method and algorithm have been explained in section 4, and section 5 concludes this paper.

2 Notations and Assumptions

The following notations and assumptions are used through the paper.

2.1 Notations

$P: \{P_1, P_2, \dots, P_n\}$ is the set of n -heterogeneous processors.

$T: \{t_1, t_2, \dots, t_m\}$ is the set of m -tasks of a program.

t_i : i -th task of the program.

P_j : j -th processor of the system.

t^*i : The restricted task to be executed first.

e_{kij} : The execution cost for the executing task t_i on the processor p_j during the k -th phase..

c_{kij} : Inter task communication cost between task t_i and task t_j during k -th phase.

r_{kij} : Residence cost for the task t_i to reside on P_j during k -th phase.

m_{ki} : The relocation cost for the task t_i at the end of the k -th phase.

TCOST (A): Cost of an assignment A.

RES_LIST $\{k\}$: Tasks priority list for the k -th phase.

2.2 Assumptions:

The program to be executed is assumed to be collection of tasks that are free, in general, to reside on either of the processors. A task may be a module of executable program. The given program runs in various phases with one of its task executing on a processor in each phase and each of the other tasks residing on any of the processors. In each phase, EC, RC of non-executing tasks, ITCC of executing task with non-executing tasks and REC for reassigning each task from one processor to the others at the end of phases are known. Some of the tasks have restrictions for being allocated on some particular processor because of the non-availability of desired facilities which may include access to particular peripheral device, access to a particular task, and high speed arithmetic capability of the processor. The execution costs of such tasks are taken to be infinity on those processors where these tasks cannot be executed.

3. Problem Statement

Consider a distributed program consisting of a set $T = \{t_1, t_2, t_3, \dots, t_M\}$ of M tasks to be allocated to a set $P = \{p_1, p_2, p_3, \dots, p_N\}$ of N processors divided into K phases. The problem for allocating a program dynamically that has been divided into K number of phases. The phase-wise processing efficiency of individual processor is given in the form of matrix $ECM(,) = [e_{kij}]$ of order $(K \times M \times N)$ and RC of each residing task is reflected through matrix $RCM(,) = [r_{kij}]$ of order $(K \times M \times N)$. ITCC between executing task and another tasks are taken in the form of matrix $ITCCM(, ,) = [c_{kij}]$ of order $(K \times M)$. The k -th column of this matrix represents the ITCC between the executing task and all other residing tasks for the k -th phase. Relocation costs for reassigning the tasks from one processor to another at the end of the phases is shown by matrix $RECM(,) = [relkixy]$ of order $(K \times M \times N \times N)$ where $relkixy$ is the relocation cost for reassigning the i -th task from p_x processor to p_y processor at the end of k -th phase. In this paper we have consider that the relocation cost $relkixy$ is not constant when i -th task relocates from one processor to another at the end of the k -th phase.

In general, the objective of task assignment is to minimize the completion cost of a distributed program by properly mapping the tasks to the processors. The cost of an assignment A, TCOST (A) is the sum of execution, inter-task communication, residence and relocation costs i.e.

$$TCOST (A) = EC (A) + RC (A) + ITCC (A) + REC (A) \quad (1)$$

In order to make the best use of the resources in a DCS we would like to find an assignment having minimum cost.

4. Proposed Method and Algorithm:

The model starts from the first phase for allocating the tasks and moves to the next phase. Let t_s is the s -th executing task and t_g ($s \neq g$) be the g -th residing tasks during k -th phase. The executing task t_s must be assigned to processor p_r during k -th phase for which sum of its execution and relocation costs is minimum. This minimum cost for the task t_s is represented by TEC_k . First for all the residing tasks we compute their priorities. Residing tasks are selected in order of their priorities and selected residing task is assigned on that processor for which the residing cost is minimum. For the k -th phase, the tasks priority list $RES_LIST\{k\}$ for the residing tasks is generated by shorting the residing tasks in decreasing order of $RELC(g)$. The $RELC(g)$ for the residing tasks are calculated by equation 2 as:

$$RELC(g) = \sum_{j=1}^n r_{kgj} \quad , \quad g = 1, 2, \dots, M, \quad g \neq s \quad (2)$$

Tie-breaking is done randomly. We select the first tasks t_g from list $RES_LIST\{k\}$. When residing task t_g resides on a processor it has three types costs i.e. residing, relocation and communication cost with the executing task t_s . Therefore, the residing task t_g must be reside to that processor p_h where it has minimum costs mentioned above. This minimum cost for the task t_g is represented by $TRC(g)_k$. List $RES_LIST\{k\}$ is modified by deleting the task t_g from $RES_LIST\{k\}$. This process of executing the residing tasks will be continue until $RES_LIST\{k\}$ becomes empty.

The total cost TRC_k due to the residing tasks in the k -th phase will be:

$$TRC_k = \sum_{\substack{g=1 \\ i \neq s}}^M TRC(g)_k \quad , \quad g = 1, 2, \dots, M, \quad g \neq s \quad (3)$$

The total cost $(TCOST)_k$ for the k -th phase is the sum of TEC_k and TRC_k i.e.:

$$(TCOST)_k = TEC_k + TRC_k \quad (4)$$

The total optimal cost, $TCOST$ for the program will be:

$$TCOST = \sum_{k=1}^K (TCOST)_k \quad (5)$$

Using the arrays representation for the execution, residing, inter task communication and relocation costs a mathematical model has been developed for allocating the distributed program dynamically which has been splitting into any number of tasks partition and completing their execution in any number of phases. The process of allocating the tasks to the processors is given below in the form of algorithm.

4.1 Algorithm:

To assign a distributed program consisting of a set $T = \{t_1, t_2, t_3, \dots, t_M\}$ of M tasks which is divided into K phases to a set of n processors $P = \{p_1, p_2, p_3, \dots, p_N\}$ in order to minimize the total cost.

Step-1: Input: $K, M, N, ECM(, ,), RCM(, ,), RECM(, ,), ITCCM(, ,)$

Step-2: Start with phase $k=1$.

Step-3: Calculate $NECM(, ,)$ by adding the $ECM(, ,)$ with $RCM(, ,)$ for k -th phase.

Step-4: Assign the executing task t_s to processor p_r for which cost is minimum say TEC_k .

Step-5: Add the effect of communication cost of executing tasks t_s with other residing tasks by adding k th column of $ITCCM(, ,)$ to all the column of $NECM(, ,)$ except the r th column at which this task is assigned.

Step-6 (a): For the k -th phase calculate:

$$RELC(g) = \sum_{j=1}^n r_{kgj} \quad , \quad g = 1, 2, \dots, M, \quad g \neq s$$

(b): Generate the task list $RES_LIST\{k\}$ by arrange the residing tasks values in decreasing order of their

$$RELC(g) = \sum_{j=1}^n r_{kgj} \quad , \quad \text{In case of tie do it randomly.}$$

(c): Initialize:

$$RES_LIST\{k\} \leftarrow RES_LIST\{k\}$$

Step-7(a): Select the first tasks t_g from list $RES_LIST\{k\}$.

(b): Assign the residing tasks t_g to processors p_h at which time is minimum say $TRC(g)_k$

(c): $RES_LIST\{k} \leftarrow RES_LIST\{k} \setminus \{t_g\}$

(d): If $RES_LIST\{k} \neq \emptyset$

then

repeat Steps 7(a) to 7(c)

else

go to Step 7(e)

(e): Calculate:

$$TRC_k = \sum_{\substack{g=1 \\ 1 \neq s}}^M TRC(g)_k$$

Step-8: Store executing and residing tasks and their position on processors in a linear array POSS (,)k. Calculate:

$$(TCOST)_k = TECK + TRCk$$

Step-9: $k=k+1$.

Step-10: Repeat step-3.

Step-11: Modify the cost of NECM (, ,) by adding rth row for the task t_s of the matrix REC(,,) for (k-1)th phase to the sth row.

Step-12: Modify the cost of NECM (, ,) by adding hth row for the task t_g of the matrix REC(,,) for (k-1)th phase to the gth row.

Step-13: Repeat Step-4 to Step-7.

Step-14: Store executing and residing tasks and their position on processors in linear array POSS (,) k+1.

Calculate:

$$(TCOST)_{k+1} = TECK+1 + TRCk+1$$

Step-15: If $k \neq K$

then

repeat step-9

else

go to step 16.

Step-16: Calculate:

$$TCOST = \sum_{k=1}^K (TCOST)_k$$

Step-17: End.

4.2 Implementation of the model:

Input of the Algorithm: Consider a program made up of four tasks $\{t_1, t_2, t_3, t_4\}$ to be allocated on three processors $\{p_1, p_2, p_3\}$ in six phases.

Table- 1 Phase-wise ECM (,,) and RCM (,,)

Tasks	Phases	Execution Costs			Residence Costs		
		ECM(,,)			RCM(,,)		
		p ₁	p ₂	p ₃	p ₁	p ₂	p ₃
1	t ₁	-	-	-	3	1	2
	t ₂	2	6	4	-	-	-
	t ₃	-	-	-	3	4	2
	t ₄	-	-	-	4	3	2
2	t ₁	6	4	8	-	-	-
	t ₂	-	-	-	2	1	3
	t ₃	-	-	-	3	2	2
	t ₄	-	-	-	3	4	2
3	t ₁	-	-	-	2	3	1
	t ₂	-	-	-	2	2	2
	t ₃	-	-	-	3	4	2
	t ₄	6	3	∞	-	-	-
4	t ₁	-	-	-	2	1	1
	t ₂	-	-	-	2	1	2
	t ₃	6	6	8	-	-	-
	t ₄	-	-	-	2	2	2
5	t ₁	-	-	-	1	2	1
	t ₂	∞	6	8	-	-	-
	t ₃	-	-	-	2	3	1
	t ₄	-	-	-	2	3	1
6	t ₁	-	-	-	2	1	1
	t ₂	-	-	-	1	1	2
	t ₃	8	2	6	-	-	-
	t ₄	-	-	-	2	3	3

Table-2 Phase wise ITCCM (,,)

	Phase-wise communication cost of executing task with other residing tasks					
	ITCCM(,,)					
Phases Tasks	1	2	3	4	5	6
t ₁	2	-	3	2	4	2
t ₂	-	0	2	3	-	0
t ₃	4	3	0	-	2	-
t ₄	2	2	-	5	1	3

Table-3 Phase-wise RECM (,,)

RECM(,,)													
Phases	Tasks	t ₁			t ₂			t ₃			t ₄		
	Processors	p ₁	p ₂	p ₃	p ₁	p ₂	p ₃	p ₁	p ₂	p ₃	p ₁	p ₂	p ₃
1	p ₁	0	2	3	0	1	1	0	4	5	0	3	1
	p ₂	2	0	1	1	0	2	4	0	1	3	0	1
	p ₃	3	1	0	1	2	0	5	1	0	1	1	0
2	p ₁	0	3	3	0	1	1	0	2	1	0	1	1
	p ₂	3	0	2	1	0	1	2	0	2	1	0	3
	p ₃	3	2	0	1	1	0	1	2	0	1	3	0
3	p ₁	0	4	2	0	4	2	0	4	2	0	1	4
	p ₂	4	0	2	4	0	2	4	0	5	1	0	2
	p ₃	2	2	0	2	2	0	2	5	0	4	2	0
4	p ₁	0	2	1	0	2	2	0	4	5	0	2	1
	p ₂	2	0	0	2	0	2	4	0	2	2	0	0
	p ₃	1	0	0	2	2	0	5	2	0	1	0	0
5	p ₁	0	2	3	0	1	1	0	2	1	0	4	2
	p ₂	2	0	1	1	0	3	2	0	2	4	0	2
	p ₃	3	1	0	1	3	0	1	2	0	2	2	0
6	p ₁	0	0	0	0	0	0	0	0	0	0	0	0
	p ₂	0	0	0	0	0	0	0	0	0	0	0	0
	p ₃	0	0	0	0	0	0	0	0	0	0	0	0

Output of the Algorithm: The present tasks allocation model gives the following optimal assignment as:

POSS (,) 1 = {(t₁, p₂), (t₂, p₁), (t₃, p₁), (t₄, p₃)}

(TCOST) 1= (TEC) 1+ (TRC) 1=12.

POSS (,) 2 = {(t₁, p₂), (t₂, p₁), (t₃, p₁), (t₄, p₃)}

(TCOST) 2= (TEC) 2+ (TRC) 2=16.

POSS (,) 3 = {(t₁, p₂), (t₂, p₂), (t₃, p₁), (t₄, p₂)}

(TCOST) 3= (TEC) 3+ (TRC) 3=15.

POSS (,) 4 = {(t₁, p₂), (t₂, p₂), (t₃, p₁), (t₄, p₁)}

(TCOST) 4= (TEC) 4+ (TRC) 4=16.

POSS (,) 5 = {(t₁, p₂), (t₂, p₂), (t₃, p₁), (t₄, p₃)}

(TCOST) 5= (TEC) 5+ (TRC) 5=15.

POSS (,) 6 = {(t₁, p₃), (t₂, p₂), (t₃, p₃), (t₄, p₃)}

(TCOST) 6= (TEC) 6+ (TRC) 6=13.

$$TCOST = \sum_{k=1}^6 (TCOST)_k = 87$$

The above result shows that the assignments of the tasks are changing phase wise i.e. result depend on the characteristic of the computation. The optimal cost for given tasks program is 185.

5. Conclusion:

In this paper we have presented an efficient solution technique for allocating the tasks of a program, divided into k-phases to N>2 processors. The present model is compared with Kumar et al.[9], Bokhari [5], Yadav et al.[13] and Singh et al [15]. Kumar et al. [9] have developed a mathematical model for allocating "M" tasks of a program to "N" processors (M>N) divided into any number of phases. In this paper authors have considered only execution, residence and inter-task communication costs. It is also observed the model [9] does not work for the system program having relocation cost for reassigning each task from one processor to the others at the end of phases. Bokhari [5] have developed a graph theoretical approach for finding out the optimal dynamic assignment of a modular program only for the two-processor system considering execution, residence, inter-task communication and relocation costs. The models presented by Yadav et al.[13] and Singh et al [15] have considered that the cost for relocating the task from one processor to another processor at the end of each phase is same for all the processors while in the present paper we have developed a mathematical model for arbitrary number of phases and processors with random program structure considering varying relocation costs processor to processor.

References:

1. C.H. Lee and D. Lee and M. Kim, "Optimal task assignment in linear array networks," IEEE Trans. Comput. C-41, 7, Jul. 1992, pp. 877-880,
2. H. S. Stone, "Multiprocessor scheduling with the aid of network flow algorithms," IEEE Trans. Software Eng. SE3, 1, Jan. 1977, pp. 85-93.
3. S.H.Bokhari "Assignment Problem in Parallel and Distributed Computing"Kulwer Academic Publishers, 1987.
4. S. H. Bokhari, "A shortest tree algorithm for optimal assignments across space and time in a distributed processor systems," IEEE Trans. Software Eng. SE-7, 6, Nov. 1981, 583-589.
5. S. H. Bokhari, "Dual processor scheduling with dynamic Reassignment," IEEE Trans. Software Eng.SE-5, 4,July 1979, 341-349.
6. D. Fernandez- Baca, "Allocating modules to processors in a distributed system," IEEE Tkans. Software Eng. SE-15, 11, Nov. 1989, pp. 1427-1436.
7. Ghafoor and J. Yang, "A Distributed Heterogeneous Supercomputing Management System,"IEEE Comput, 6, Jun. 1993, pp. 78-86.
8. D.F.Towsley, "Allocating programs containing branches and loops within a multiple processor system" IEEE Trans. Software Eng. SE-12,10, Oct.1986,pp.1018-1024.
9. V.Kumar, M.P.Singh, P.K.Yadav "An Efficient Algorithm for Multiprocessor Scheduling With Dynamic Re-assignment." Published in Proceeding of 6th National seminar on Theoretical Computer Science held at Banasthali Vidyapith form Aug. 5-8, 1996, pp.105-118.
10. M.P.Singh, P.K.Yadav ,Harendra Kumar "A Tasks Allocation Model for Efficient Utilization Of Processor's Capacity in Heterogeneous Distributed Systems" in. 9th Conference of International Academy of Physical Sciences, held at Agra University Agra, during Feb.-03-05, 2007.
11. S.Y. Cho, K.H.Park "Dynamic Task Assignment in Heterogeneous Linear Array Networks for Metacomputing" IEEE Proceeding, 1994, pp. 66-71.

12. V. Kumar, M. P. Singh, and P. K. Yadav, "An efficient algorithm for multiprocessor scheduling with dynamic reassignment", In proceeding of the 6th national Seminar on Theoretical computer Science (Aug. 1996) India, 105-118.
13. P. K. Yadav, M. P. Singh and H. Kumar, "Scheduling algorithms: tasks scheduling algorithm for multiple processors with dynamic reassignment", Journal of computer Systems, networks and communications, Vol.2008, article id 578180, 9 pages.
14. A. Elsadek and B. E. Wells, A heuristic model for task allocation in heterogeneous distributed computing systems, The Int. J. of computer and their applications, Vol.6, No.1 ,March 1999.
15. M.P, Singh, P.K.Yadav, H. Kumar, B.Agarwal, "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, , pp 321-331, 2012.

