ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Task Allocation Technique in Distributed Computing System for Utilization of Processors by Reducing Inter Task Communication Cost

Uma Sharma, Prem Sagar Sharma

M.Tech Scholar, Dept. of CSE., BSAITM Faridabad, Maharishi Dayanand University, India

Assistant Professor, Dept. of CSE, BSAITM Faridabad, Maharishi Dayanand University, India

ABSRACT: In Distributed Computing System (DCS), the application software divides into small tasks and proper mapping of these tasks among processors are one of the important parameter which determine the efficient utilization of available Processor's Capacity. In this paper the task allocation in DCSs in such a way that the load on each processing node is almost balanced. Further, the development of an effective algorithm for creating clusters of tasks to by using both inter task communication cost and capacity of each processor. Allocating n-clusters to n-processors (where, # of cluster equal to # of processors) with different capacities of a given distributed computing system. The proposed method is making the clusters of those tasks in which inter task communication is maximum.

KEY WORDS: Distributed Processing System, Processor's Capacity, Inter Task Communication, Execution Cost, Task Cluster Making (TCM) Algorithm and Task Allocation.

I.INTRODUCTION

Distributed Computing System (DCS) has attracted several researchers by posing several challenging problems. In a DCS, the execution of a program may be distributed among several processing elements to reduce the overall cost of execution by taking advantage of inhomogeneouscomputational capabilities and other resources within the system. The task allocation in a DCS finds extensive applications in the faculties where large amount of data is to be processed in relatively short period of time, or where real-time computations are required. These applications require not only very fast computation speeds but also different strategies involving distributed task allocation systems. In such applications the quality of the output is proportional to the amount of real-time computations. The total cost of execution of a distributed program consists of processors costs plus message transmission cost.

Assigning tasks to processors is called task allocation, which involves the allocation of tasks to processors in such a way that some effectiveness measures are optimized. Assigning "m" tasks to "n" processors, through exhaustive enumeration, results in nm possible ways.

The task partitioning and task allocation activities influence the distributed software properties such as inter task communication (ITC) cost [1, 2]. Many approaches have been reported for solving the task assignment problem in DCS.The problem of finding an optimal dynamic assignment of a modular program for a two-processor system is analyzed by [3].

One measure of usefulness of a general-purpose Distributed Computing System is the system's ability to provide a level of performance commensurate to the degree of multiplicity of resources present in the system. Taxonomy of approaches to the resource management problem is reported [4]. The taxonomy, while presented and discussed in terms of distributed scheduling, is also applicable to most types of resource management. A model for allocating information files have been reported by [5] the model considers storage cost, transmission cost, file lengths, and request rates, as well as updating rates of files.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

1.Homogenous Distributed Computing System

A bunch of processors is connected to a high speed network for accomplishing the required task, is defined as distributed computed system (DCS). HMDCS is a part of Distributed computing system. Itcomprises of group of identical processors connected to a high speed network. Identical processors means all the processors connected to the system are of same computationalspeed and processor complexity.To achieve maximal utilization of all the available resources tasks needs to be divided equally among all the processors present in HMDCS [7]. Some examples of HMDCS algorithms are ISH, HLFET, TSB, MTSB etc.[8]

2.Heterogeneous Distributed Computing System

HTDCS can be defined as the interconnection of several processors, having different computational speed, and are connected with different speed links used for solving group of problems which have different computational requirements.

All the processors in this type of computing system, usually incorporates specialized processing capabilities and thus tasks are assigns to them accordingly.HTDCSs [7] are generally applicable in the commercial and scientific research fields where, real-time security, robustness and high performance, is the main priority.

Some examples if HTDCS algorithms [9] are SNLDD, ACS, Tabu Search, NSGA-II, etc.

In both heterogeneous and homogeneous distributed computing system, proper task scheduling strategy is required in order to achieve high performance and to efficiently utilize available resources and processors. [8, 10]

II.STATIC SHEDULING ALGORITHM

Many researchers have studied in the field of static task scheduling problem for Distributed Computing System. Some of their static scheduling algorithms are compared as given in table-1.[8]

Table-1: Comparison between various Static Task Allocation Algorithms in DCS

S. No.	Algorithms	Objectives	Advantage	Limitations
1	TSB	To provide an effective tasks order of execution on the processors	Takes less scheduling length among BNP Class of scheduling	Work can be done to improve its scheduling length
2	MTSB	Communication time is not considered.	Provided shorter scheduling length	Can be applied in heterogeneous environment
3	HLFET	Provide good scheduling	Use static b-level level scheduling	Heterogeneous
4	CPOPA	To minimize the computation cost and time	Gives better results for graphs with higher communication to computation ratio (CCR)	Homogeneous system
5	ISH	Minimize makespan and completion time	Utilizes idle time slots in scheduling	Heterogeneous and dynamic
6	CEFT	To minimize the execution time	Minimum threshold values for schedule length	Use to multiple constrained critical path in subsequent of machine
7	PGSA	High performance, scalability, and fast running time	Provides optimal solutions	Dynamic system
8	MIACUA	Optimal assignment of distributed tasks.	Reduces average time and space complexities	Dynamic system

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

III. LITERATURE SURVEY

Table -2: Summary of Various research papers

Author/year	Technique	Advantages
Abhilasha Sharma, Surbhi Gupta IASIR- USA,2014	The model deals with the problem of optimal task allocation and load balancing in DCS.	The maximum value of TOC (j) will be the optimal cost of the system
Manisha Sharma, Harendra Kumar, Deepak Garg, 2012	The module allocation problem having multiple objectives such as: minimization of system cost, maximization of system reliability and load balancing for proper utilization of the processor‟s capacity	large amount of data is processed in a short period of time
Mohammad I. Daoud, Nawwaf Kharma, 2007	The LDCP algorithm is a list-based scheduling algorithm that uses a new attribute to efficiently select tasks for scheduling in HDCSs	The LDCP algorithm is more effectively able to deal with theincrease in communication cost compared to both the DLS and HEFT algorithms.
Poornima Bhardwaj, Vinod Kumar, IJCDS Vol. No.4, Issue II, 2014	the problem of task allocation in a DCS such that the load on each processing node is almost balanced. The model includes inter – task communication cost (ITCC) along with the execution cost (EC). A new concept of Load Balancing Factor (LBF) is introduced	It is noticeable that when LBF is used the total cost of the system is decreased.
Ranjit Rajak, IJCA (0975 – 8887) Volume 44– No11, April 2012	To provide an effective tasks order of execution on the processors	Takes less scheduling length among BNP Class of scheduling
Ranjit Rajak, C.P.Katti & Nidhi Rajak, IJNCAA- 2013	Communication time is not considered.	Provided shorter scheduling length

1. PROBLEM STATEMENTS

A critical look at the available literature indicates that the following issues need to be addressed while allocating tasks in distributed computing system

1. Efficient utilization of available Processor's Capacity
2. Minimize the data travelling in network i.e. Proper utilization of network bandwidth in distributed computing system.
3. Allocated load on all the processors is balanced.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

4. Allocation of a set of tasks of a program to a set of processors with the constraints of minimizing Inter Task communication (ITC) cost and maximizes the overall throughput of the system

2. DEFINITIONS

Tasks Size (TS): The task size depends on the length of tasks and generally counted in bytes.
Processor's Execution Rate (PER): The execution rate of each processor is the speed (bytes/ second) of the processor at which they execute the tasks.
Execution Time (ET): The execution time of each task depends on the capability of the processor to which it is assigned.
Inter Tasks Communication Time (ITCT): The inter task communication time c_{ij} of the interacting tasks t_i and t_j is incurred due to the data units exchanged between them during the process of execution.
Response Time (RT): The response time is a function of the amount of computation to be performed by each processor and the communication time. This function is defined by considering the processor with the heaviest aggregate computation and communication loads of the processors.

IV.ASSUMPTIONS

Several assumptions have been made to keep the algorithm reasonable in size while designing the algorithm. The program is assumed to be the collection of "m" tasks, which are to be executed on a set of "n" processors that have different processing capacity. It is assumed that the size (in bytes) of each tasks and execution rate (in bytes/ second) for each processor is known. The size of each task has been defined in column matrix TSM= [tsi] ,where tsi represent the size of i-th task. The execution rate of each processor is taken in the form of row matrix PERM=[erj] , where erj denotes the execution rate for j-th processor. The inter-processor communication cost is taken in the form of a symmetric matrix named as inter task communication cost matrix ITCCM= [cij] order m, where cij represents the communication cost between task ti and tj. Further cij=cji. Whenever a group of tasks is assigned to the same processor, the ITCC between them is zero.. A processor can simultaneously execute a task and communicate with another processor. The overhead incurred by this isnegligible, so for all practical purposes we will consider it as zero.

V.TASKS ALLOCATION PROCESS

The allocation of program tasks is to be carried out so that each task is assigned to a processor whose capacities are most appropriate for the tasks, and the inter-processor cost is minimized. The present model passes through the following steps:

1.Task Cluster Making

Tasks are clustered based on their communication requirement. Highly communicating tasks are clustered together to reduce communication delays. Usually number of tasks clusters should be equal to the number of processor so that one to one mapping may result.

ALGORITHM: TASK CLUSTER MAKING (TCM)

TCM (ITCM [][], m, n**)**
{ ***// m- number of tasks& n-number of processors***
sizeofCluster= floor(m/n) ; ***// Number of task in a cluster***
numberofCluster=n;
// Number of clusters is equal to number of processors

1. ***[Make all pair matrix(APM[][]) b/w t_i& t_j]***
   Initialize k=1;
   Repeat i=1 to m
      Repeat j=1 to m
         APM[k][1]=i; ***[r-th task]***

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

```
  APM[k][2]=j; [s-th task]
  APM[k][3]=ITCM[i][j];
      [cost between r-th & s-th task]
            k++;
      End loop
   End loop
```

2. *[Arrange APM (All Pair Matrix) in descending order on the basis of cost between r-th & s-th tasks]*

```
MAXCC [][]=descOreder(APM[][]);
```

3. *[Create Clusters using MAXCC]*

```
Initialize   r=1,s=1;
Repeat   r<=n
Repeat   s<=m
Cr =MAXCC[s][1];
// Cr is initial cluster i.e. 1st task of MAXCC is initialize to Cr
 p=1;
// p counter variable, initialize thecondition to open new cluster
Cs=Adjacent of Cr, & yet not assign to any clusters;
Repeat   p<=sizeofCluster;
Cr ← Cr ∪ Cs;
 P++;// increase the # task in Cluster
        End loop
s++;// Move to next task
    End loop
r++; //Open new cluster
 End loop

}
```

2.Calculation of execution cost: Using task size of cluster CS[] and execution rate of processor (PERM[ ]) , the execution time e_{ij} of each cluster is calculated and taken in the form of execution time matrix, ETM=[e_{ij}] of order n x n. Where

$$e_{ij}= CS_i * er_j \quad (1 \leq i \leq n , 1 \leq j \leq n)$$

Note: # of clusters is equal to # of processors.

3.Assignment of the Clusters: After making the 'n' number of tasks clusters we get a modified ETM (,) of order n. We assign these clusters to that processor using assignment problem technique for which execution time is minimum.

VI.RESULTS & DISCUSSIONS

To justify the present method, we considered an example of a Distributed Computing Systemwith consisting of a set of three (n=3) processors as P = {P1, P2, P3} connected by an arbitrary network and a set of nine (m=9) executable tasks T = {t1, t2, t3, t4, t5, t6, t7, t8, t9}. The size of each task and processor's execution rate have been taken in the form of matrices TSM ( ) and PERM( ) respectively. The Inter tasks communication cost between the tasks has been taken in the form of ITCTM (,) of order m x m.

Input of the system:

Number of processors in the DCS (n) = 3

Number of tasks to be executed (m) = 9

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

ITCCM (,)=

	T1	T2	T3	T4	T5	T6	T7	T8	T9
T1	0	38	45	41	44	39	46	47	40
T2	38	0	41	44	35	30	31	45	36
T3	45	41	0	35	30	31	45	36	38
T4	41	44	35	0	31	45	35	38	30
T5	44	35	30	31	0	36	38	29	41
T6	39	30	31	45	36	0	29	41	44
T7	45	31	45	36	38	29	0	30	28
T8	47	45	36	38	29	41	30	0	31
T9	40	26	38	30	41	44	28	31	0

TSM (,)=

t_1	258
t_2	320
t_3	185
t_4	300
t_5	210
t_6	270
t_7	200
t_8	230
t_9	295

1. Cluster Forming

Arranging the All Pair Matrix in descending order on the basis of cost, we get MAXCCM (,) as

Ti	Tj	Cost
8	1	47
7	1	46
3	1	45
6	4	45
7	3	45
8	2	45
4	2	44
5	1	44
9	6	44
4	1	41
3	2	41
8	6	41
9	5	41
9	1	40
6	1	39
2	1	38
7	5	38
8	4	38
9	3	38
6	5	36
7	4	36

Ti	Tj	Cost
9	2	36
5	2	35
4	3	35
6	3	31
5	4	31
7	2	31
9	8	31
6	2	30
8	7	30
9	4	30
5	3	30
7	6	29
8	5	29
9	7	28
4	4	0
7	7	0
2	2	0
3	3	0
5	5	0
6	6	0
8	8	0
1	1	0
9	9	0

Applying the **Task Cluster Making (TCM) Algorithm**, we get three tasks clusters and modified ETM (,) matrix as:

Cluster 1

List of Tasks [t8, t1, t7]

Cluster2

List of Tasks [t3, t2, t4]

Cluster3

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

List of Tasks [t6, t9, t5]

Cluster size matrix

CS[]=		
	Cluster1	**688**
	Cluster2	**805**
	Cluster3	**775**

ETM(,)=		P1	P2	P3
	Cluster1	402.48	520.128	467.152
	Cluster2	470.925	608.58	546.595
	Cluster3	453.375	585.9	526.225

Applying the assignment problem technique on ETM (,), we get the following assignments and results:
Cluster 1 [t8, t1, t7]➔ P2
Cluster2[t3, t2, t4]➔ P1
Cluster3 [t6, t9, t5]➔ P3

Table - 3: Compare Processor Execution Time (PET)of Proposed & Existing system [10]

	PET (Exiting)	PET (Proposed)
Processor-1	462	471
Processor-2	581	520
Processor-3	464	526
Total	1507	1517

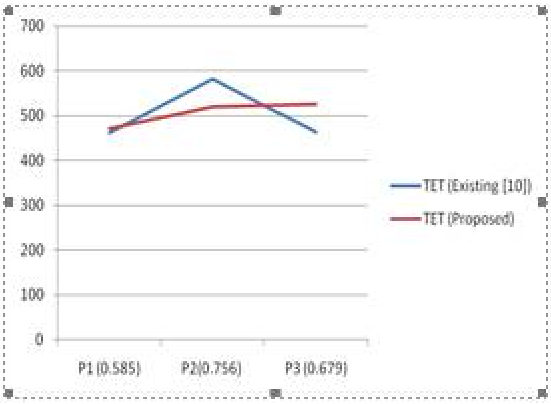

Figure- 2: depicts the Total Execution Time (TET) of each processor

Table - 4: Compare Inter task Communication cost of Proposed & Existing system [10]

	Inter task Communication cost (Exiting)	Inter task Communication cost (Proposed))
b/w C1 & C2	356	335
b/w C1 & C3	331	319
b/w C2 & C3	335	306
Total	**1022**	**960**

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

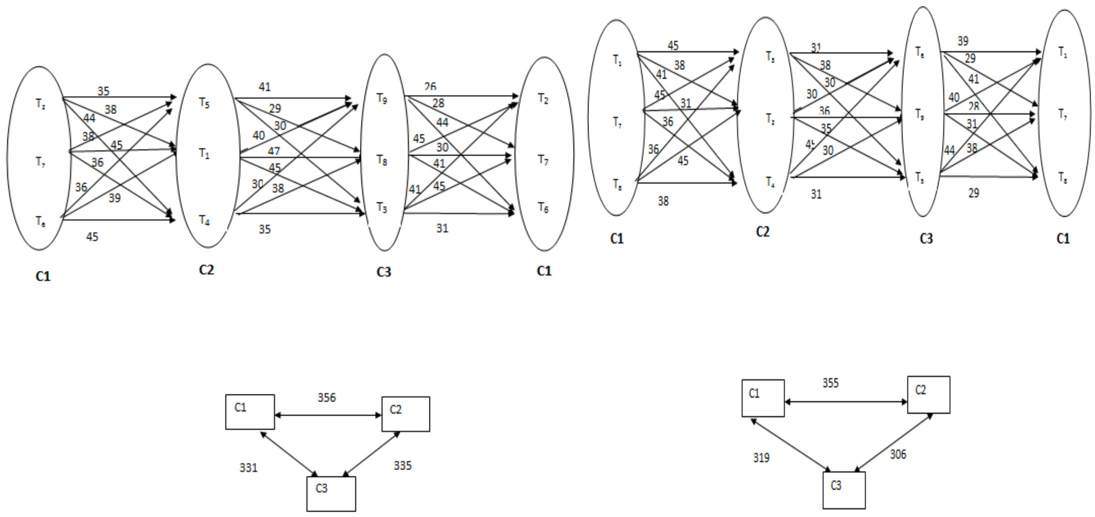

Figure- 3: Total communication cost between clusters for existing system [10] vs. proposed system

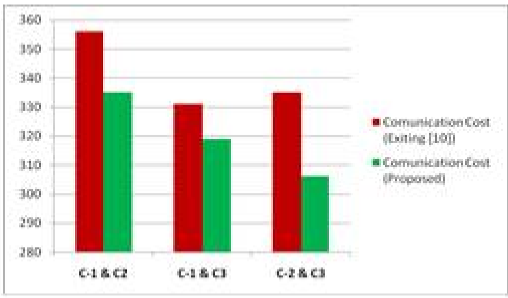

Figure- 4: depicts the Inter Task Communication cost between clusters

VII.CONCLUSION

In this paper, we have considered the task allocation problem in heterogeneous Distributed Computing System (DCS), a critical phase in DCS with the goal of minimizing cost of the system, and maximizing throughput of the system, various static scheduling algorithms for parallel computing system have been discussed in table-1. Fig-2 depicts the Total Execution Time (TET) of each processor and Fig-5depicts the Inter Task Communication cost between clusters, form the both figures it is concluded that the both are the ideally linked where the inter task communication cost between clusters is minimized & within cluster is maximized. Utilization of the processors is maximised and the service time of the processor is minimum.

REFERENCES

1. SHATZ, S/M., and WANG, J.P., “Introduction to distributed software engineering”, Computer, 1987,20 (10), pp. 23-31.
2. D.F. Baca, “Allocation Tasks to Processor in a Distributed System,” IEEE Trans. On Software Engineering, Vol.15 pp.1427-1436, 1989.
3. S.H. Bokhari, “Dual Processor Scheduling with Dynamic Re-Assignment”, IEEE Trans. On Software Engineering, Vol.SE-5 pp. 341-349, 1979.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

4. T.L. Casavent, and J.G. Kuhl, "A Taxonomy of Scheduling in General Purpose Distributed Computing System", IEEE Trans. On Software Engineering, Vol.14 pp.141-154, 1988.
5. W.W. Chu, "Optimal File Allocation in a Multiple Computing System", IEEE Trans. On Computer, Vol.C-18 pp.885-889, 1969.
6. O.I. Dessoukiu-EI and W.H. Huna, "*Distributed Enumeration on Network Computers,"* IEEE Trans. On Computer, Vol.C-29 pp.818-825, 1980.
7. Tarek H, Jan J, "*A Near Lower-Bound Complexity Algorithm for Compile-Time Task Scheduling in Heterogeneous Computing Systems*", Proceedings of the ISPDC/HeteroPar'04 IEEE, 2004.
8. Khushboo Singh ,Mahfooz Alam and Sushil Kumar Sharma, "*A Survey of Static Scheduling Algorithm for Distributed Computing System*'' International Journal of Computer Applications (0975 – 8887) Volume 129 – No.2, November2015
9. Kalpana A.M., Avinash W., "*Comparative Study of Static Task Scheduling Algorithms for Heterogeneous Systems*"International Journal on Computer Science and Engineering (IJCSE), Vol. 5, No. 03, 2013.
10. P.K. Yadav, Preet Pal Singh, P. Pradhan, ''A Tasks Allocation Algorithm for Optimum Utilization of Processor's in Heterogeneous Distributing Computing Systems'' IJRREST: International Journal of Research Review in Engineering Science Technology (ISSN 2278- 6643) | Volume-2 Issue-1, March 2013

By going through the literature analysis of some of the important web page ranking algorithms, it is concluded that each algorithm has some relative strengths and limitations. A tabular summary is given below in table 1, which summarizes the techniques, advantages and limitations of some of important web page ranking algorithm