IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 4, Ver. I (July – Aug. 2015), PP 94-99
www.iosrjournals.org

Task allocation model for Balance utilization of Available resource in Multiprocessor Environment

Ruchi Gupta[1]Dr. P.K Yadav[2]
Research Scholar, Uttarakhand Technical University, Dehradun
Central Building Research Institute, Roorkee, U.K, India.

Abstract: *Distributed computing systems are of current interest due to the advancement of microprocessor technology and computer network. The prime function of effective utilization of distributed system is accurately mapping of task and tasks their scheduling on different processors for reducing their total finish time. Total runtime is taken time for all module with the runtime of tasks and their communication cost among tasks. In Distributed processing system, partitioning of a task into modules and proper allocation of module among processors are more important factor for efficient utilization of resources. In this paper, we discuss allocation of a set of 'm' module of a task to a set of 'n' processors (where m > n) to get an optimal utilization of available processors. We partition the task in different size of module and design an algorithm for allocation of these modules in distributed processing environment. While designing the model Per Bit Processing Rate, Inter Task Communication Cost, Task Size has been taken into consideration. We consider the constraints of minimizing execution cost and their Inter task communication and maximize the overall throughput of the system to be taking as constraints in such a way that allocated load on all the processor is balances.*

Keyword: *Distributed computing, Task Allocation, Module distribution, Execution Cost, Communication Cost, Task Size.*

I. Introduction

A distributed computing system (DCS) consists of a set of multiple processors interconnected by communication links. A very common interesting problem in DCS is the task allocation. Partitioning of the application software into small module of task and the proper allocation of these module among processors are one of the important parameter which determine the efficient utilization of available resources (1) (2).the allocation policy could be either static or dynamic. A task allocation model that allocate application task among the processor in distributed system satisfying minimum inter-processor communication cost, balanced utilization of each processor and all application requirement. (3). A module incurs an execution cost that may be different for each processor assignment and optimal solution to the problem of allocating communication periodic tasks to heterogeneous processing nodes in a distributed system has been reported (4).Yadav et al have reported a reliability evaluations of distributed system based on failure data analysis (5).Kumar et al. Developed a model for allocating m task to n processors divided it into any number of phases (6). The main objective of this problem is to minimize the total program execution period by allocating the tasks in such a way that the allocated load on each processor should be balanced. The model utilized the mathematical programming technique for execution of the module considering that each module to be executed through all the processors. (7). we use the problem of task allocation in heterogeneous distributed system with the goal of maximizingthe system reliability (8). To solve the proposed model we utilizing a task allocation algorithm for optimum utilization of Processor's in heterogeneous distributed computing system (9) (12) to obtain the better result in reasonable amount of time.

II. Mathematical Modeling

The main objective of this problem is to minimize the total program execution period by allocating the tasks in such a way that the allocated load on each processor should be balanced. The model utilized the mathematical Programming technique for execution of the module considering that each module to be executed through all the processors. The Execution Cost and Inter Tasks Communication Cost are considered for developing the algorithm. The specific task allocation problem being addressed as follows:-

Considered an application program consisting of a set of "m" task $T = \{t_1, t_2, t_3, t_4 \ldots\}$ and heterogeneous distributed processing system consisting a set of 'n' processor $P = \{p_1, p_2, p_3, p_4 \ldots\}$. (10) Where we assume that m > n, and allocated each of m module to one of the processor in such a manner that total system time is minimizing and processing load on the processor is balanced . We partition the task in different size of module and design an algorithm for allocation of these modules in distributed processing environment.

Input Parameter

- Processor Service Rate (PSR)
- Task Set (TM)
- Execution Cost (EC)
- Inter Task Communication Cost(ITCC)

A. Processor Service Rate (PSR)

Processor service rate is the execution rate Er_j (Per Bit Processor) of each processor which implemented in the form of linear arrayPSR(j).Where (j=1, 2,3,4,5,6,7,8.................n).

$$\text{PSR(j)} = \begin{vmatrix} \text{Er1} \\ \text{Er2} \\ \text{Er3} \\ \text{Er4} \\ \text{Er5} \\ \vdots \\ \text{Ern} \end{vmatrix}$$

B. Task Size (TM)

It depends on the length of task (in bytes).we takes task size and break them in multiple module of task and put in the form of linear array TM(i) (where i=1,2,3,4,5,6,…..m).

$$\text{Tm(i)} = \begin{vmatrix} \text{tm1} \\ \text{tm2} \\ \text{tm3} \\ \text{tm4} \\ \text{tm5} \\ \vdots \\ \text{tmm} \end{vmatrix}$$

C. Inter Task Communication Cost(ITCC)

Inter Task Communication Cost between executing task and another tasks is taken in the form of matrix Inter Task Communication Cost Matrix (ITCC).The Inter Processor Communication cost CC_{ik} of the interacting tasks ts_i and ts_k is the minimum cost required for the exchange of data units between the processors during the process of execution. (11) Here we use ITCC as symmetric matrix which is order of m*m.

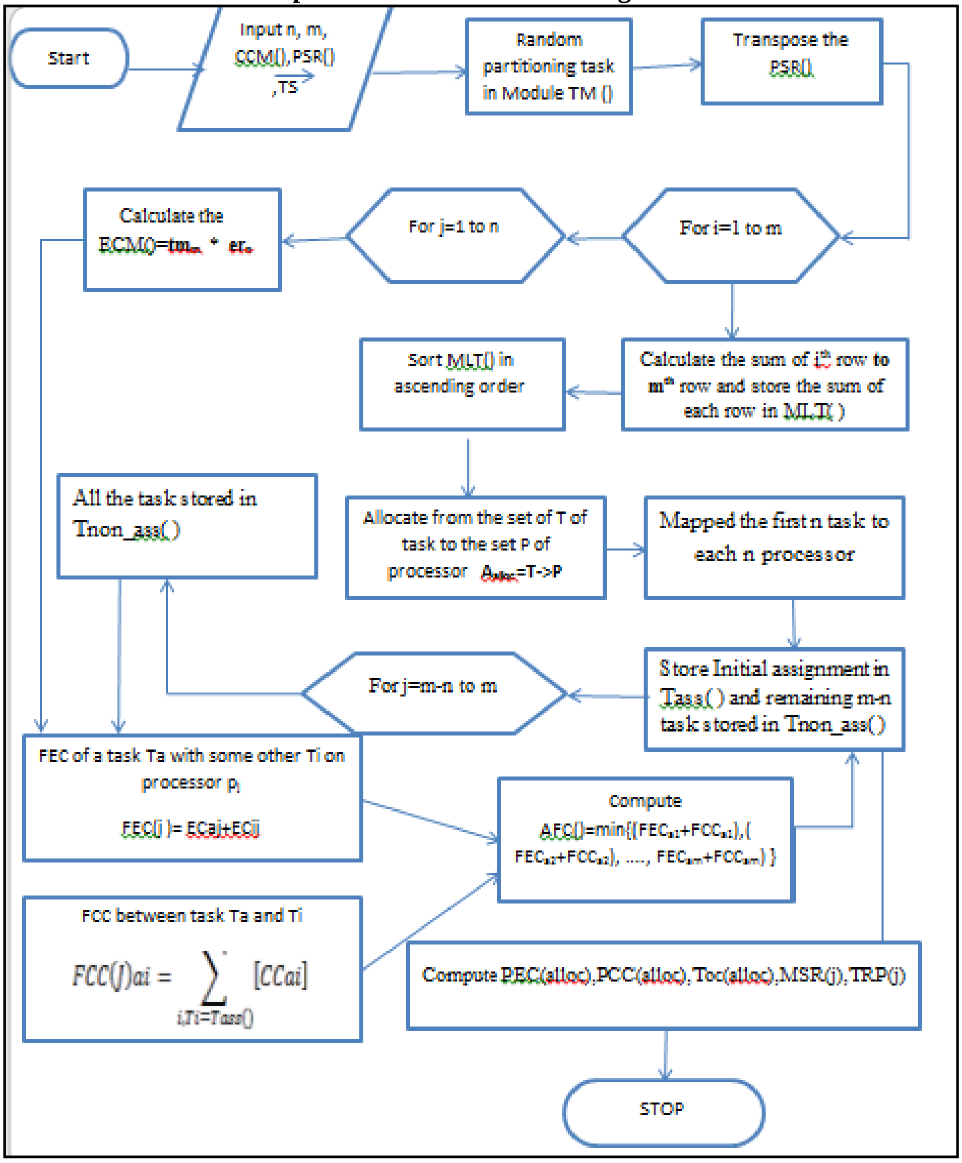
D. Execution Cost (EC)

The execution cost $\mathbf{ec_{ij}}$of each module $\mathbf{ts_i}$depends on the capability of processor $\mathbf{p_j}$ to which they assigned and the work to be performed by each module. To determine the **EC**, initially we have product of **PSR (j)** and **TM (i)** and store them in Execution Cost Matrix **(ECM (i,j))**of order **m*n**.

1. NOTATION

N= No of Processor
M= No of Module
PSR= Processor Service Rate
TM =Task Module
ITCM= Inter Task Communication Cost
ECM = Execution Cost Matrix
ITCC =Inter Task Communication Cost
TAM = Task Set
MLT= Minimally Linked Task
Tnon_Ass= Table for Non Assigned Task
TAss= Table for Assigned Task
FEC = Fused Execution Cost
FCC =Fused Communication Cost
AFC =Additive Fused Cost
PEC = Processor Execution Cost
PITCC= Processor Inter Task Communication Cost
TOC = Total Order Cost
MSR =Mean Services Rate
TRP = Throughput

Computational Flow Chart Of Algorithm



III. Implementation Of The Model

To justify the procedure discussed in section a system has been considered consisting a set of 3 heterogeneous processors i.e p_1, p_2, p_3 and a one task size we partition this task in 9 modules as: T (tm_1, tm_2, tm_3, tm_4, tm_5, tm_6, tm_7, tm_8, tm_9).

Inputs:-

$$PSR(\quad) = \begin{vmatrix} 0.585 \\ 0.756 \\ 0.679 \end{vmatrix}$$

Task Size =2268

After Partitions the task in module

$$TM(\quad) = \begin{matrix} tm1 \\ tm2 \\ tm3 \\ tm4 \\ tm5 \\ tm6 \\ tm7 \\ tm8 \\ tm9 \end{matrix} \begin{vmatrix} 165 \\ 235 \\ 293 \\ 277 \\ 252 \\ 293 \\ 242 \\ 224 \\ 287 \end{vmatrix}$$

Computer ECM (,)

$$\mathrm{ECM}(,) = \begin{matrix} \mathrm{tm1} \\ \mathrm{tm2} \\ \mathrm{tm3} \\ \mathrm{tm4} \\ \mathrm{tm5} \\ \mathrm{tm6} \\ \mathrm{tm7} \\ \mathrm{tm8} \\ \mathrm{tm9} \end{matrix} \begin{vmatrix} 165 \\ 235 \\ 293 \\ 277 \\ 252 \\ 293 \\ 242 \\ 224 \\ 287 \end{vmatrix} * \begin{vmatrix} p1 & p2 & p3 \\ 0.585 & 0.756 & 0.679 \end{vmatrix}$$

$$ECM(,) = \begin{matrix} \\ tm1 \\ tm2 \\ tm3 \\ tm4 \\ tm5 \\ tm6 \\ tm7 \\ tm8 \\ tm9 \end{matrix} \begin{vmatrix} p1 & p2 & p3 \\ 96.525 & 124.740 & 112.035 \\ 137.475 & 177.660 & 159.565 \\ 171.405 & 221.508 & 198.947 \\ 162.045 & 209.412 & 188.083 \\ 147.420 & 190.512 & 171.108 \\ 171.405 & 221.508 & 198.947 \\ 141.570 & 182.952 & 164.318 \\ 131.040 & 169.344 & 152.096 \\ 167.895 & 216.972 & 194.873 \end{vmatrix}$$

Input ITCC(i,k)-

$ITCC_{(i,k)}$=

	Tm_1	**Tm_2**	**Tm_3**	**Tm_4**	**Tm_5**	**Tm_6**	**Tm_7**	**Tm_8**	**Tm_9**
Tm_1	0	38	45	41	44	39	46	47	40
Tm_2	38	0	41	44	35	30	31	45	36
Tm_3	45	41	0	35	30	31	45	36	38
Tm_4	41	44	35	0	31	45	36	38	30
Tm_5	44	35	30	31	0	36	38	29	41
Tm_6	39	30	31	45	36	0	29	41	44
Tm_7	46	31	45	36	38	29	0	30	28
Tm_8	47	45	36	38	29	41	30	0	31
Tm_9	40	36	38	30	41	44	28	31	0

Compute the MLT( ) for all tasks by using equation 2

$$MLT(\quad) = \begin{matrix} tm1 \\ tm2 \\ tm3 \\ tm4 \\ tm5 \\ tm6 \\ tm7 \\ tm8 \\ tm9 \end{matrix} \begin{vmatrix} 340 \\ 300 \\ 301 \\ 300 \\ 284 \\ 295 \\ 283 \\ 297 \\ 288 \end{vmatrix}$$

Sort the MLT ( ) in ascending order

$$\mathrm{MLT}(\quad) = \begin{matrix} \mathrm{tm7} \\ \mathrm{tm5} \\ \mathrm{tm9} \\ \mathrm{tm6} \\ \mathrm{tm8} \\ \mathrm{tm2} \\ \mathrm{tm4} \\ \mathrm{tm3} \\ \mathrm{tm1} \end{matrix} \begin{vmatrix} 283 \\ 284 \\ 288 \\ 295 \\ 297 \\ 300 \\ 300 \\ 301 \\ 340 \end{vmatrix}$$

At the Time 1 there is no allocation to any processors.

$T_{non\_ass()}$={ tm_7, tm_5, tm_9, tm_6, tm_8, tm_2, tm_4, tm_3, tm_1},

$T_{ass(\,)}$={ø}.

Now we first allocate first three task from $T_{non\_ass(\,)}$ i.e. tm_7, tm_5, tm_9 and assign them respectively

P1- tm_7,

P2- tm_5

P3- tm_9

Now we eliminate them from $T_{non\_ass(\,)}$={ tm_6, tm_8, tm_2, tm_4, tm_3, tm_1},and add them in $T_{ass(\,)}$ ={ tm_7, tm_5, tm_9};

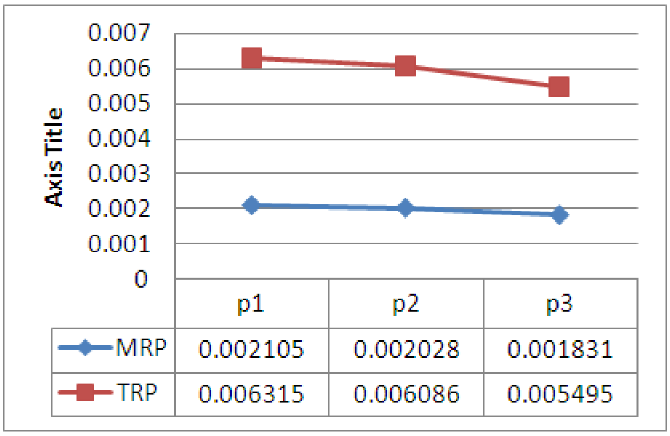
We take a first task from the $T_{non\_ass(\ )}$ to allocate, calculate fused execution cost and fused communication cost to tasks for corresponding processor .For task tm_6:-

Processor	**FEC (A)**	**FCC (B)**	**AFC (A+B)**
P1	312.975	146	458.975
P2	412.02	152	564.02
P3	393.82	134	527.82

In this calculation we find $AFC(1)_{6,7}$= min {AFC()ai}from equation ,so we allocate tm_6 to Processor P_1. Now we eliminate tm_6 them from $T_{non\_ass(\ )}$={ tm_8, tm_2, tm_4, tm_3, tm_1}, and add them in $T_{ass(\ )}$ ={ tm_7, tm_5, tm_9, tm_6};Similarly we repeat this process until we allocate all tasks to the respective processors. When we completely execute this process we get a task allocation table:-

Table-1

Processor	**List of allocated tasks**
P1	**tm_7, tm_6, tm_4**
P2	**tm_5, tm_2, tm_1**
P3	**tm_9, tm_8, tm_3**



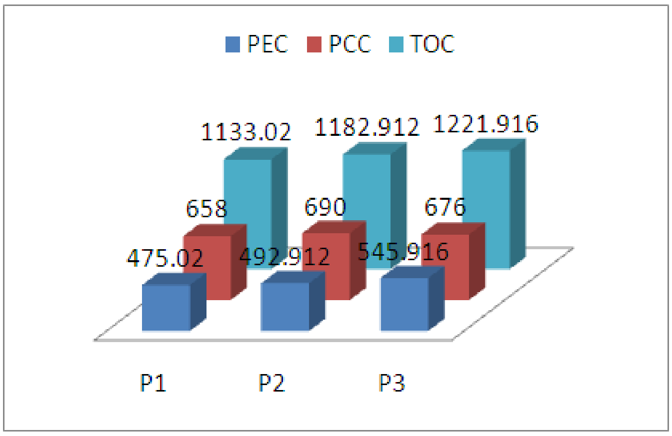


Fig.1.2-Total Busy time of processorFig.1.1-Mean Service Rate and Throughput of the Processor

IV. Conclusion

In this paper we deal with efficient Mathematical and computational algorithm to identify the optimal method use the processor capacity and upgrade the preformation of the distributed system. In table 1-2 tasks are executing on processors simultaneously and maximum busy time of the system are 1221.916 times units and average throughput of the system0.005965 time's units. A comparison study has also been conducted with the algorithm developed by Singh et al. (9).

Table-3

Processor	PEC	PCC	MRP	TRP	TOC
P1	**462.150**	**698**	**.00216**	**.00649**	**1160.150**
P2	**580.608**	**692**	**.00172**	**.00517**	**1272.608**
P3	**482.090**	**676**	**.00207**	**.00622**	**1158.090**

Comparison of the both algorithm is shown in table- 4

Table-4

Present Methods				**Yadav et al[9]**		
Processor	EC	ITCC	TOC	EC	ITCC	TOC
P1	475.02	658	**1133.02**	462.150	698	**1160.150**
P2	492.912	690	**1182.912**	580.608	692	**1272.608**
P3	545.916	676	**1221.916**	482.090	676	**1158.090**

It concludes that compared results have been shown in table 4 and it is shows that the present algorithm gives better result.

Bibliography

[1]. Introduction to distributed software enginnering . **SHATZ, S/M.AND WANG,J.P.** 2, s.l. : computer socity , 1987.

[2]. Allocation task to processor in a Distributed system. **Baca, D.F.** s.l. : IEEE Trans.On Software Engineering , 1989, Vol. 15.

[3]. A Task allocattion model for distributed computing system . **Ma.P, Y.R,Lee, E.Y.S, & Tsuchiya,M.** s.l. : Computers IEEE Trans., 1982, Vol. 1.

[4]. Assignment Scheduling Communicating periodic tasks in distributed real time system. **Peng.Dar-Tezen and K.G.Shin, Abdel,T.F.Zoher.** s.l. : IEEE Trans. on software Engineering, 1997, Vol. 13, pp. 745-757.

[5]. Reliability Evaluation of Distributed System Based on Failure Data Analysis,. **Yadav.P.K, Bhatia K,and Gulati Sagar.** s.l. : International Journal of Computer Engineering, 2010, Vol. 2.

[6]. An efficient algortihm for multiprocessor scheduling with dynamic reassignment . **V.Kumar, M.P Singh and P.K Yadav.** s.l. : national seminar on Theoretical Computer Science, 1996, Vol. 1.

[7]. Optimal Execution Cost of Distributed system through Clustring . **Khandelwal, Anju.** 3, s.l. : International Journal of Engineering Science and Technology, March 2011, Vol. 3 .

[8]. Task allocation for maximizing reliabilty of distributed systems: a simulated annealing approach . **Hamam, Gamal Attiya and Yskandar.** 10, s.l. : Journal of Parallel and Distributed Computing , 2006, Vol. 66.

[9]. A task allocation algorithm for optimum utilization of processor in heterogeneous distrubiuted system. **P.K Yadav, Preet Pal Singh,P.Pradhan.** 1, s.l. : International Journal of Research review in Engineering Sciecne and Technology, march 2013, Vol. 2.

[10]. Task allocation model for distributed system,. **G. Sagar, and A.K. Sarje.** 9, s.l. : Int. J. system science, 1991, Vol. 22.

[11]. “A fast algorithm for allocation task in distributed processing system,” . **Vinod Kumar, M.P. Singh, P.K. Yadav.** s.l. : Annual convention of computer society of India, Hyderabad, 1995.

[12]. Optimal utilization of processors in computer communication environment”. **Singh M.P, P.K.Yadav and Singh Jugmendra.** s.l. : National Conference on Modern Development in Engineering & Applied Seiences (NCMDES-09) at College of Engineering &Applied Research, Feb 27-28,2009.