IJMS, Vol. 11, No. 1-2, (January-June 2012), pp. 121-131

ISSN: 0972-754X

RELIABILITY EVALUATION OF DISTRIBUTED SYSTEM BASED ON FAILURE DENSITY DATA ANALYSIS

Manisha Sharma1, P. K. Yadav2, Surinder Kumar3

Abstract: Distributed system is the study of geographically separated processors that communicate with one another with message passing i.e. a processor works on data provided by some other processor. So, Reliability is the most important factor to be considered in such environment. The present piece of research suggests an approach for how the reliability of a distributed system gets affected with the introduction of probability of failure density in the execution and communication process. In the research, a study has been carried out on 'm' different tasks with different sizes, which have been executed on 'n' different processors (where m>n, as generally the scenario is) along with their different communication rates, with the assumption of a minute chance of failure. In this scenario, the reliability of the distributed system is calculated for different task combinations and thus the most reliable solution is determined from the obtained values of 'm' and 'n' and found workable in all possible cases.

Keywords and Phrases: Distributed System, Communication Reliability, Execution Reliability, Inter- processor communication time, Failure Density, Execution Time and Index.

1. INTRODUCTION

A distributed system consists of a collection of autonomous computers linked by a computer network and equipped with distributed system software [1]. Distributed systems are implemented on hardware platforms that vary in size from a few workstations interconnected by a single local area network to thousands of computers connected via multiple wide area networks. Distributed processing involves cooperation among several loosely coupled computers communicating over a network. Distributed Processing System provides cost-effective ways for improving computer system's resource sharing, performance, throughput, fault-tolerance and reliability [2], [3], [4], [5], [6], [7], [8]. A very common research problem for distributed computing systems is the allocation problem, in which system reliability is to be maximized. These problems are studied by various researchers such as, [9], [10], [11], [12], [13], [14], [15], and [16]. Kumar [15] discussed a task allocation problem for optimizing the execution and communication reliability of a computer communication network. Yadav [17] discussed a reliability evaluation of distributed system based on failure data analysis. He considered the unreliability matrices for the execution and communication for the purpose of the allocation. Raghavendra et. al. [18] described that the reliability of the distributed computing system depends not only on reliability of a communication network but also on the reliability of the processing nodes and distribution of the resources in the network. While Shatz *et. al*. [19] explained when the system hardware configuration is fixed the system reliability mainly depends on the allocation of resources. In the present work, we have introduced a probability of failure density in the communication and execution process and simulated the model to see its effect on reliability of the system. Several sets of input data are used to test the effectiveness and efficiency of model. It is found that the model is suitable for arbitrary number of processor with the random program structure.

The rest paper is organized as follows. Definition and notations are defined first in section 2, in section 3, problem statement. In section 4, proposed method and algorithm has been discussed. The section 5 shows the experimental results in comparison to other scheduling methods and concludes the paper.

2. DEFINITION AND NOTATIONS

2.1 *Execution Time*: Each task t_i has an Execution Time when executed on k^{th} processor $e_{ki}(1 \leq i \leq m \ and \ 1 \leq k \leq n)$,

$$ET = \sum_{k=1}^{n}\{\sum_{i=1}^{m} e_{ki}x_{ik}\} \quad (1)$$

2.2 *Failure Density:* The ratio of the number of failures during a given unit interval of time to the total number of items or Total initial population at the very beginning of the test and denoted by f_d

$$f_d = \frac{no.\ of\ failure}{Total\ initial\ population} \quad (2)$$

2.3 *Inter-Processor Communication Time:* The IPC time $c_{ik}(1 \leq i \leq m \ and \ 1 \leq k \leq n)$of the interacting tasks t_i and t_k is incurred due to the data units exchanged between them during the process of execution.

$$CT = \sum_{k=1}^{n}\{\sum_{k=1}^{n} c_{ik}y_{ik}\} \quad (3)$$

2.4 *Total Time:* The total time is expressed as the sum of execution costs along with communication cost.

$$\text{Total time consumed} = CT + ET \quad (4)$$

2.5 *Execution Reliability:* The Execution Reliability [ER] of a task t_i on the processor p_k is the probability r_{ki} $(1 \leq i \leq m \ and \ 1 \leq k \leq n)$that task will be successfully executed on processor p_k, within specified conditions.

$$ER = \prod_{i=1}^{m}\{\sum_{k=1}^{n} r_{ki}\, x_{ik}\} \quad (5)$$

2.6 ***Communication Reliability:*** The Communication Reliability r^{*}_{ik} $(1 \leq i \leq m \; and \; 1 \leq k \leq n)$ is the probability of successfully data units exchanged between the tasks $t_i \; and \; t_k$ under the given conditions.

$$CR = \prod_{k=1}^{n} \{ \sum_{k=1}^{n} r^{*}_{ik} \, x_{ik} \} \tag{6}$$

2.7 ***Total Reliability:*** Total reliability is expressed as product of the products of the execution reliabilities along with communication reliabilities.

$$Trel = \prod_{i=1}^{m} \{ \sum_{k=1}^{n} r_{ki} \, x_{ik} \} \times \prod_{k=1}^{n} \{ \sum_{k=1}^{n} r^{*}_{ik} \, x_{ik} \} \tag{7}$$

2.8 ***Task Communication Rate:*** The task communication rate w_{ik} is per unit time that a task t_i takes when communicates with task t_k.

$$TC = \prod_{k=1}^{n} \{ \sum_{k=1}^{n} w_{ik} \, y_{ik} \} \tag{8}$$

2.9 ***Index:*** The index represents the ratio of overall reliability to the total time consumed.

$$\text{Index} = \text{Trel} / \text{Total time consumed.} \tag{9}$$

Where, $x_{ik} = \begin{Bmatrix} 1, if \; i^{th} \; task \; is \; assigned \; to \; k^{th} \; processor, \; and \\ 0, \; otherwise \end{Bmatrix}$

$$y_{ik} = \begin{Bmatrix} 1, if \; the \; task \; assigned \; to \; processor \; i \; communicates \; with \\ the \; task \; assigned \; to \; processor \; k. 0, otherwise \end{Bmatrix}$$

3. PROBLEM STATEMENT

Let the given system consists of a set of 'n' processors P = $\{ p_{1,} p_{2\,,} . \; . \; . \; , p_n \}$, (with specific processing rate) interconnected by communication links and a set of 'm' tasks T = $\{t_{1,} \, t_{2,} . \; . \; ., \; t_m\}$ of different size(s) to be executed on these processors, with the possibility of failure in communication and execution process. The reliability of execution and communication has to be evaluated in this scenario which further would be tested with different task clusters to obtain the most reliable solution.

4. PROPOSED METHOD

We begin the work with a processor rate matrix PRM (,), task size matrix TSM (,) and task communication rate matrix TCRM (,) along with randomly generated execution failure matrix EFM (,) and communication failure matrix CFM (,) .First of all, with the help of PRM (,) and TSM (,), the execution time matrix ETM (,) is calculated followed by the generation of execution survival density matrix ESDM (,) and communication survival density matrix CSDM (,). Then corresponding execution reliability density matrix ERDM (,) as well as communication reliability density matrix CRDM (,) is also evaluated, using the method suggested by [20]. Task combinations are generated with the help of formula suggested by Bhatia *et. al,* [21]. To get the allocation a modified version of row and column assignment method of Yadav *et. al*. [22] is employed which allocates a task to a processor where it has maximum reliability and correspondingly less execution time. Overall reliability of the distributed system is evaluated as the product of communication reliability and execution reliability. The same process is repeated for the available set of task clusters which ultimately provide the most reliable solution to the problem.

ALGORITHM

To given an algorithmic representation to the technique mentioned in the previous section, Let us consider a system in which a set of 'm' tasks $T = \{t_1, t_2, ..., t_m\}$ is to be executed on a set of 'n' available problems $P = \{p_1, p_2, ..., p_n\}$.

Step-1: Input $PRM = [p_{k1}]_{n\times 1}$, $TSM = [t_{1i}]_{1\times m}$, where $1 \le i \le m$, $1 \le k \le n$, $TCRM = [c_{ik}]_{m\times m}$, where $1 \le i \le k \le m$ and randomly generate

$EFDM = [f_{ki}]_{n\times m}$, where $1 \le i \le m$, $1 \le k \le n \leftarrow$ Execution failure density matrix

$CFDM = [g_{ik}]_{m\times m}$, where $1 \le i \le k \le m \leftarrow$ Communication failure density matrix

Step-2: Evaluate $ETM = [e_{ki}]_{n\times m}$, where $e_{ki} = p_{k1}t_{1i} \leftarrow$ the execution time matrix ETM (,) is obtained as the product of PRM (,) and TSM (,).

Step-3: Apply the failure density i.e. Failure density $(f_d) = \dfrac{N}{T}$, where N= no. of failure, T= total initial population.

Step-4: Evaluate the matrices: $ESDM = [s_{ki}]_{n\times m}$, where $s_{ki} = e_{ki} - f_{ki} \leftarrow$ Execution survival density matrix

$ERDM = [r_{ki}]_{n\times m}$, where $r_{ki} = \dfrac{s_{ki}}{e_{ki}} \leftarrow$ Execution reliability density matrix

$CSDM = [s^*_{ik}]_{m\times m}$, where $s^*_{ik} = c_{ik} - g_{ik} \leftarrow$ Communication survival density matrix

$CRDM = \left[r_{ik}^*\right]_{n\times m}$, where $r_{ik}^* = \frac{s_{ik}^*}{c_{ik}}$ ←Communication survival density matrix

Step-5: Compute the total combinations and store them in TCOMB (,).

$TCOMB(,) = \left(\frac{n\times^m c_{m-n}}{ceil\left(\frac{m}{n}\right)}\right)$, where $ceil\left(\frac{m}{n}\right)$ rounds the elements of $\left(\frac{m}{n}\right)$ to the nearest integers greater than or equal to $\left(\frac{m}{n}\right)$.

Step-6: Modify $ETM = \left[e_{ki}\right]_{n\times m}$ by adding j^{th} row to i^{th} row and deleting j^{th} row and store the all entries in new matrix $ETM = \left[e_{ki}^*\right]_{n\times m}$.

Step-7: Modify, $ERDM = \left[r_{ki}\right]_{n\times m}$, $CRDM = \left[r_{ik}^*\right]_{m\times m}$ and $TCRM = \left[c_{ik}\right]_{m\times m}$ by multiplying j^{th} row to i^{th} row and deleting j^{th} row and store the all entries in new matrices

$NERDM = \left[r_{ki}^c\right]_{n\times m}$, $NCRDM = \left[r_{ik}^{c*}\right]_{m\times m}$ and $NTCRM = \left[c_{ik}^*\right]_{m\times m}$ respectively.

Step-7.1: For j = 1 to m and for k =1 to n, find the minimum of j^{th} row (say mn_{jk}) falling in j^{th} column and $mn_{jk} = 0$.

Step-7.2: For j = 1 to m and for k =1 to n, find the minimum of k^{th} column (say mn_{jk}) of $NERDM = \left[r_{ki}^c\right]_{n\times m}$, which has in j^{th} row and $mn_{jk} = 0$.

Step-7.3: For k = 1 to m and for j = 1 to n, search for a row in $NERDM = \left[r_{ki}^c\right]_{n\times m}$ having only one zero.

Step-7.4: For k = 1 to m and for j = 1 to n, search for a column in $NERDM = \left[r_{ki}^c\right]_{n\times m}$ having only one zero.

Step-7.5: If the task t_i is assign to processor p_k then evaluate the execution time and otherwise find the minimum of the entries elements for the remaining rows and replace at zero.

Step-8: Evaluate IPCTime between m and n i.e. IPCTime $=\sum_{k=1}^{n}\{\sum_{k=1}^{n} c_{ik}y_{ik}\}$, where $1 \le i \le m$, $1 \le k \le n$.

Step-9: Compute

$$ET = \sum_{k=1}^{n}\{\sum_{i=1}^{m} e_{ki}x_{ik}\}, \text{ where } 1 \le i \le m,\ 1 \le k \le n.$$

$$\text{TCRate} = \prod_{k=1}^{n}\{\sum_{k=1}^{n} w_{ik}\, y_{ik}\}, \text{ where } 1 \le i \le m,\ 1 \le k \le n.$$

Total Data Transferred = $\sum_{k=1}^{n}\{\sum_{k=1}^{n} c_{ik}y_{ik}\}\times\prod_{k=1}^{n}\{\sum_{k=1}^{n} w_{ik}\, y_{ik}\}$

Total Time Consumed = $\sum_{k=1}^{n}\{\sum_{i=1}^{m} e_{ki}x_{ik}\}+\sum_{k=1}^{n}\{\sum_{k=1}^{n} c_{ik}y_{ik}\}$

ER $=\prod_{i=1}^{m}\{\sum_{k=1}^{n} r_{ki}\, x_{ik}\}$, where $1 \le i \le m$, $1 \le k \le n$.

CR $=\prod_{k=1}^{n}\{\sum_{k=1}^{n} r_{ik}^{*}\, x_{ik}\}$, where $1 \le i \le m$, $1 \le k \le n$.

Total Reliability = $\prod_{i=1}^{m}\{\sum_{k=1}^{n} r_{ki}\, x_{ik}\}\times\prod_{k=1}^{n}\{\sum_{k=1}^{n} r_{ik}^{*}\, x_{ik}\}$

Index = Total Reliability/ Total time consumed

Step-10: Repeat Step-6 to Step-9.

Step-11: Find the all values of ET, IPCTime, ER, CRel., Total data transferred, Total time consumed, Total reliability, Index and some derived values of total combinations are store in reliability evaluation table.

Step-12: The highest value of reliability is chosen to find the most reliability solution.

Step-13: End.

6. EXPERIMENTAL RESULTS AND CONCLUSION

Reliability evaluation of distributed system based on failure data density analysis and most reliable solution was discussed in the previous section. To evaluate our proposed model we use a different size of problem in this section. In experimental problem we have consider a distributed program made up of seven tasks $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ to be executed on a set of three processors $P = \{p_1, p_2, p_3\}$.The work begins with consider a processor rate matrix PRM (,) and a task size matrix TSM (,).

$$\text{PRM} = \begin{bmatrix} 0.279 \\ 0.254 \\ 0.291 \end{bmatrix} \qquad \text{TSM} = [12 \quad 9 \quad 8 \quad 11 \quad 7 \quad 14 \quad 13]$$

The execution time matrix ETM (,) is obtained as the product of PRM (,) and TSM (,) as follows:

$$\text{ETM} = \begin{array}{c} \\ \end{array}\begin{bmatrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \\ p_1 & 3.348 & 2.511 & 2.232 & 3.069 & 1.953 & 3.906 & 3.627 \\ p_2 & 3.048 & 2.286 & 2.032 & 2.794 & 1.178 & 3.556 & 3.302 \\ p_3 & 3.492 & 2.619 & 2.328 & 3.201 & 2.037 & 4.074 & 3.783 \end{bmatrix}$$

Now consider a probability of failure that can take place in the execution process with the help of a randomly generated execution failure matrix EFM (,).

$$\text{EFM} = \begin{bmatrix} & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \\ p_1 & 0.0019 & 0.0013 & 0.003 & 0.0041 & 0.0011 & 0.009 & 0.0033 \\ p_2 & 0.0018 & 0.0033 & 0.004 & 0.007 & 0.007 & 0.0091 & 0.0036 \\ p_3 & 0.0072 & 0.0061 & 0.008 & 0.0029 & 0.0039 & 0.0018 & 0.0072 \end{bmatrix}$$

Using the failure density, with the help of ETM (,) and EFM (,), EFDM (,) is evaluated as follow:

FDM=

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
p_1	0.0019	0.0013	0.003	0.0041	0.0011	0.009	0.0033
p_2	0.0018	0.0033	0.004	0.007	0.007	0.0091	0.0036
p_3	0.0072	0.0061	0.008	0.0029	0.0039	0.0018	0.0072

With the help of ETM (,) and EFM (,), ESDM (,) and ERDM (,) is evaluated as follows [20]:

ESDM =

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
p_1	3.3464	2.5094	2.2307	3.0677	1.9514	3.9037	3.6261
p_2	3.0464	2.2846	2.0292	2.7915	1.1721	3.5525	3.309
p_3	3.4899	2.6167	2.3246	3.2001	2.0351	4.0736	3.7811

ERDM=

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
p_1	0.9995	0.9994	0.9994	0.9996	0.9992	0.9994	0.9998
p_2	0.9995	0.9994	0.9986	0.9991	0.9959	0.9990	1.0021
p_3	0.9994	0.9991	0.9985	0.9997	0.9991	0.9999	0.9995

The task communication rate matrix and randomly generated communication failure matrix are as follows:

TCRM =

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
t_1	1.00	0.97	0.96	0.94	0.93	1.00	0.94
t_2	0.97	1.00	1.00	0.97	0.96	0.93	1.00
t_3	0.96	1.00	1.00	0.97	0.98	0.96	0.99
t_4	0.94	0.97	0.97	1.00	0.96	0.94	0.92
t_5	0.93	0.96	0.98	0.96	1.00	0.92	0.93
t_6	1.00	0.93	0.96	0.94	0.92	1.00	0.91
t_7	0.94	1.00	0.99	0.92	0.93	0.91	1.00

CFM=

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
t_1	0.00	0.0023	0.0018	0.0055	0.0037	0.00	0.0041
t_2	0.0023	0.00	0.00	0.0048	0.0033	0.0021	0.00
t_3	0.0018	0.00	0.00	0.0018	0.0039	0.0027	0.0044
t_4	0.0055	0.0048	0.0018	0.00	0.0011	0.0018	0.0090
t_5	0.0037	0.0033	0.0039	0.0011	0.00	0.0018	0.0073
t_6	0.00	0.0021	0.0027	0.0087	0.0018	0.00	0.0071
t_7	0.0041	0.00	0.0044	0.009	0.0073	0.0071	0.00

With the help of TCRM (,) and CFM (,), Communication reliability density matrix is obtained as:

$$CRDM = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \end{array} \begin{array}{ccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \\ 1.000 & 0.9975 & 1.000 & 0.9937 & 0.9947 & 1.000 & 0.9953 \\ 0.9975 & 1.000 & 1.000 & 0.9949 & 0.9965 & 0.9975 & 1.000 \\ 0.9980 & 1.000 & 1.000 & 0.9980 & 0.995 & 0.9971 & 0.9956 \\ 0.9937 & 0.9949 & 0.9980 & 1.000 & 0.9989 & 0.9901 & 0.9893 \\ 0.9947 & 0.9965 & 0.995 & 0.9989 & 1.000 & 0.9968 & 0.9916 \\ 1.000 & 0.9975 & 0.9971 & 0.9901 & 0.9968 & 1.000 & 0.9914 \\ 0.9953 & 1.000 & 0.9956 & 0.9893 & 0.9916 & 0.9914 & 1.000 \end{array}$$

Compute the total combinations and store them in TCOMB (,).

$$TCOMB\,(,) = \left[\left(n * {}^{m}C_{m-n}\right) \Big/ Ceil\,(m/n) \right] = 35$$

(10) TCOMB (1) = (123, 456, 7)

By applying the method suggested by Yadav *et al.* [22] following result are obtained.

Tasks	*Processors*	*ET*	*IPCTime*	*TCRate*	*ER*	*CRel.*
$t_1t_2t_3$	p_1	8.091	26.308	0.643	0.9983	0.9590
$t_4t_5t_6$	p_2	7.528	25.336	0.538	0.9940	0.9413
t_7	p_3	3.783	3.783	0.725	0.9995	0.9637

ET (1) =19.402 ER (1) = 0.992 IPCTime (1) = 55.4207 TCRate (1) = 0.250
CRel (1) = 0.8699
Therefore, Total Reliability TRel = 0.863

Repeating the above process, suggested in the algorithm the corresponding values of ET, TCRate, ER, CRel, TRel and some derived values are obtained and shown in the following Table 1.

Table (1) Reliability Evaluation Table

S.No	*TCOMB*	*ET*	*IPCTime*	*TCRate*	*Total Data Transferred*	*Total time Consumed*	*ER*	*CRel*	*TRel*	*Index*
1	123,456, 7	19.402	55.4207	0.250	13.855	74.823	0.992	0.8699	0.863	0.0115
2	124, 356, 7	20.002	58.7855	0.256	15.049	78.788	0.996	0.8828	0.879	0.0112
3	125, 346, 7	19.977	59.7027	0.260	15.523	79.679	0.994	0.8848	0.879	0.0110
4	126, 345, 7	19.927	52.4254	0.221	11.586	72.352	0.996	0.8620	0.859	0.0119
5	127, 345, 6	19.964	59.0729	0.217	12.819	79.037	0.999	0.8660	0.865	0.0109
6	134, 256, 7	20.027	59.2672	0.290	17.187	79.294	0.995	0.8713	0.867	0.0109
7	135, 246, 7	19.952	59.6047	0.272	16.213	79.557	0.995	0.8896	0.885	0.0111
8	136, 245, 7	19.952	59.1484	0.221	13.072	79.100	0.995	0.8628	0.858	0.0108
9	137, 245, 6	19.989	60.1574	0.235	14.137	80.146	0.998	0.8736	0.871	0.0109
10	145, 236, 7	20.027	59.4536	0.267	15.874	79.481	0.995	0.8706	0.866	0.0109
11	146, 235, 7	19.847	57.2814	0.217	12.430	77.128	0.995	0.8823	0.878	0.0114

12	147, 235, 6	19.914	57.8424	0.256	14.808	77.756	0.999	0.8921	0.891	0.0115
13	156, 234, 7	19.377	54.9689	0.231	12.698	74.346	0.992	0.8663	0.859	0.0116
14	157, 234, 6	19.414	55.3812	0.256	14.178	74.795	0.996	0.8838	0.880	0.0118
15	167, 234, 5	19.755	58.3081	0.231	13.469	78.063	0.998	0.8748	0.873	0.0112
16	234, 156, 7	19.377	54.9689	0.231	12.698	74.346	0.992	0.8663	0.859	0.0116
17	235, 146, 7	19.847	57.2814	0.217	12.430	77.128	0.995	0.8823	0.878	0.0114
18	236, 145, 7	20.027	59.4536	0.267	15.874	79.481	0.995	0.8706	0.866	0.0109
19	237, 145, 6	19.464	54.6018	0.217	11.849	74.066	0.993	0.8690	0.863	0.0117
20	245, 136, 7	19.952	58.3813	0.221	12.902	78.333	0.995	0.8628	0.858	0.0110
21	246, 135, 7	19.952	59.6587	0.272	16.227	79.611	0.995	0.8896	0.885	0.0111
22	247, 135, 6	19.989	59.3595	0.246	14.603	79.349	0.997	0.8867	0.884	0.0111
23	256, 134, 7	20.027	56.7562	0.290	16.459	76.783	0.995	0.8713	0.867	0.0113
24	257, 134, 6	20.039	59.0431	0.245	14.466	79.082	0.996	0.8761	0.873	0.0110
25	267, 134, 5	19.830	59.8880	0.273	16.349	79.718	0.998	0.8747	0.873	0.0110
26	345, 126, 7	19.927	58.9651	0.222	13.090	78.892	0.996	0.8620	0.859	0.0109
27	346, 125, 7	19.977	59.7027	0.261	15.582	79.680	0.994	0.8848	0.879	0.0110
28	347, 125, 6	20.014	60.1443	0.256	15.397	80.158	0.998	0.8890	0.887	0.0111
29	356, 124, 7	20.002	58.3477	0.256	14.937	78.350	0.996	0.8828	0.879	0.0112
30	357, 124, 6	20.014	58.7693	0.235	13.811	78.783	0.996	0.8947	0.891	0.0113
31	367, 124, 5	19.855	59.5227	0.256	15.238	79.378	0.997	0.8914	0.889	0.0112
32	456, 123, 7	19.402	55.4206	0.250	13.855	74.823	0.992	0.8731	0.866	0.0116
33	457, 123, 6	19.439	55.4798	0.256	14.203	74.919	0.997	0.8807	0.878	0.0117
34	467, 123, 5	19.780	59.3623	0.279	16.562	79.142	0.998	0.8967	0.895	0.0113
35	567, 123, 4	19.328	55.9581	0.285	15.948	75.286	0.995	0.8806	0.876	0.0116

We concluded that the method presented with the help of algorithm is an iterative model for reliability evaluation. We found the values of execution time, execution survival, execution reliability and corresponding communication survival and communication reliability in matrices form using the failure density. A set of communication reliability, execution reliability, processing time, communication time, total time consumed, total data transferred and an index, a parameter to evaluate the performance, is obtained on completion of one iteration. The process is repeated for all the values of TCOMB (,), to obtain the reliability evaluation table. Finally the highest value of reliability is chosen to find the most reliable solution of the problem. To justify the efficiency of the algorithm, the runtime complexity is evaluated with the help of the method suggested by H. Ellis *et. al.* [23] that comes out to be $O(mn^2)$, which is compared with that of [15], i.e. $O(m^2n)$. The results obtained by the presented model are more optimal and reliable than P. K. Yadav, K. Bhatia and Sagar Gulati [17].

REFERENCES

[1] G. Coulouris, J. Dollimore and T. Kindberg ,"Distributed System- Concepts and Design," *Second Edition, Addison-Wesley Publications*.

[2] T. C. K. Chou and J. A. Abraham, "Load redistribution under failure in distributed systems," *IEEE Trans. Computer*, Vol. C-32, pp. 799-808, Sept. (1983).

[3] D. W. Davies, E. Holler, E. D. Jensen, S. R. Kimbleton, B. W. Lampson, G. Lelann, K. J. Thurber, and R. W. Watson, "Distributed systems architecture and

implementation," *in Lecture Notes in Computer Science* , Vol. 105, Berlin, Germany: Springer-Verlag, (1981).

[4] P. Enslow ,"What is a Distributed Data Processing System," *IEEE Computer Mag.,* Vol. 11, Jan. (1978).

[5] J. Garcia-Molina ,"Reliability Issues for Fully Replicated Distributed Database*," IEEE Computer Mag.,* Vol. 16, pp. 34-42, Sept. (1982).

[6] S. Hariri, C. S. Raghavendra, and V. K. Prasanna Kumar ,"Reliability Measures for Distributed Processing Systems*," in Proc. 6th Int. Conf: Distributed Computer System*, pp. 564-571, (1986).

[7] D. A. Rennels ,"Distributed Fault-tolerant Computer Systems," *IEEE Computer Mag.*, Vol. 13, pp. 45-55, Mar. (1980).

[8] J. A. Stankovic ,"A Perspective on Distributed Computer Systems*," IEEE Trans. Computer,* Vol. C-33, pp. 1102-1115, Dec. (1984).

[9] Sagar, G. and Sarje, A. K. ,"Task Allocation Model for Distributed System*," Int. J. System Science,* Vol. 22, pp. 1671-1678, (1991).

[10] Kumar, A., Pathak, R. M., Gupta, Y. P. ,"Genetic Algorithm based Reliability Optimization for Computer Network Expansion," *IEEE Transactions on Reliability*, Vol. 44, No. 1, pp. 63-72, (1995).

[11] Kubat, Peter, "Assessing Throughput and Reliability in Communication and Comuter Networks," *IEEE Transactions on Reliability*, Vol. 37, No. 3, pp. 308-311, (1988).

[12] P.Y. R. Ma,"A Task Allocation Model for Distributing Computer System*," IEEE Trans. Computers,* Vol. 31, No. 1, pp. 41-47, (1982).

[13] Kartik, S., and Murthy, C. Siva Ram ,"Improved Task Allocation Algorithms to Maximize Reliability of Redundant Distributed Computing Systems,"*IEEE Transaction on Reliability*, Vol. 44, No. 4, pp. 575-584, (1995).

[14] M. P. Singh, P. K. Yadav, Harendra Kumar ,"A Tasks Allocation Model for Efficient Utilization of Processor's Capacity in Heterogeneous Distributed Systems," *presented in. 9th Conference of International Academy of Physical Science, held Agra*, (2007).

[15] Kumar, Avinash., Yadav P. K. & A. R. Gupta ,"An Exhaustive Approach of Distributed System Based on Cost Assignment," *South Asian Journal of Mathematics and Mathematical Science,* Vol. 5, No. 1, pp. 29-44. (2006).

[16] Rotithor, H. G., "Taxonomy of Dynamic Task Scheduling in Distributed Computing Systems*," IEEE Proc. Computer Digit Tech*., Vol. 14, pp. 1-10, (1994).

[17] P. K. Yadav, K. Bhatia & Sagar Gulati ,"Reliability Evaluation of Distributed System Based on Failure Data Analysis," *International Journal of Computer Engineering*, Vol. 2, pp. 113-118, June (2010).

[18] Raghavendra, C. S., and Hariri, S., "Reliability Optimization in the Design of Distributed Systems*," IEEE Transactions on Software Engineering*, Vol. SE-11, pp. 1184-1193, (1985).

[19] Shatz, S. M., and Wang, Jai- Ping, "Models and Algorithms for Reliability-Oriented Task, Allocation in Redundant Distributed Computer Systems," *IEEE Transaction on Reliability*, Vol. 38, No. 1, pp. 16-27, (1989).

[20] L. S. Srinath ," Reliability Engineering," East-West Press, (2006).

[21] Bhatia K., Yadav P. K. & Sagar Gulati ,"Performance Evaluation of Distributed System Through Reliability Analysis," Proceedings of International Conference on Challenges and Applications of Mathematics in Science and Technology, NIT Rourkela, pp. 613-620, (2010).

[22] Yadav P. K., Kumar, Avinash, & M. P. Singh ,"An Algorithm for Solving the Unbalanced Assignment Problem," International Journal of Mathematical Science, Vol. 12. No. 2, pp. 447-461,(2004).

[23] H. Ellis, Sahni S., & S. Rajasekaram ,"Fundamentals of Computer Algorithm," Galgotia Publications Pvt Ltd., (2005).

Manisha Sharma
Department of Mathematics,
Panjab University,
Chandigarh-160014,
Punjab (INDIA)
Email:manishatewaripu@gmail.com

P. K. Yadav
2Central Building Research Institute,
Roorkee-247667, U.K (INDIA)
Email:prdyadav@redffimail.com

Surinder Kumar
Department of Mathematics,
Panjab University,
Chandigarh-160014,
Punjab (INDIA)
Email:sur_vinaik05@yahoo.co.in